

Universidad de las Ciencias Informáticas
Facultad 2



**Universidad de las Ciencias
Informáticas**

**Sistema de envío de postales MMS a través
de dispositivos móviles**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autores: Lex Karel Zayas Hernández

Yudier Villalta Villar

Tutor: Lic. Noel García Guimeras

Ciudad de la Habana, julio de 2007

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Lex Karel Zayas Hernández
Firma del Autor

Yudier Villalta Villar
Firma del Autor

Lic. Noel García Guimeras
Firma del Tutor

“Los conceptos y principios fundamentales de la ciencia son invenciones libres del espíritu humano.”

Albert Einstein

AGRADECIMIENTOS

A nuestro Comandante Fidel, a la máxima dirección de la Revolución y a la UCI, por darnos la oportunidad de formar parte de este proyecto futuro. A nuestro tutor Noel y todo el equipo de Procyon.

Judier y Lex Karel

A mis padres por su amor, preocupación, apoyo y sobre todo por guiarme siempre por la senda correcta.

A mis amigos y compañeros que tanto me ayudaron.

A el pibe Arian por toda su ayuda.

Judier

A mi novia (Tita) por su amor y entrega.

A mis amigos de siempre, Jordanis, Brocard, Miquel...

A mis compañeros de aula durante todos estos años.

A todos los que me han ayudado de una u otra forma, durante mi vida como estudiante.

En especial a mis padres, a mis abuelos y en general a toda la familia, gracias por su apoyo y comprensión.

Lex Karel

DEDICATORIA

A mis queridos padres, que para ellos esto es una gran satisfacción y que tanto hicieron para que pudiera recorrer todo el camino hasta aquí.

A mi abuelita Arminda que es la mejor del mundo.

A mi primo Yuni que para mi es un hermano.

A toda mi grande y linda familia que tanto cariño me brinda.

Yudier

A mi mamita querida, por estar siempre a mi lado en los buenos y malos momentos. A mi papá.

A mis hermanitas, Bety y Naismy.

A mi abuelitas, Mami y Lingo.

A mis compañeros de aula, a mis profesores, en general a todas las personas que han estado conmigo en las buenas y las malas.

A mi Tita por estar siempre a mi lado.

En especial a mi abuelito (Papi), a él le debo todo lo que soy.

Lex Karel

RESUMEN

Los servicios de mensajería inalámbrica poseen en la actualidad un gran impacto en el mercado. Haciendo uso de los mismos, se puede generar una amplia variedad de servicios de valor agregado que permitan a las diferentes empresas de este sector obtener ganancias. La entidad cubana Procyon se dedica al desarrollo de aplicaciones destinadas a la telefonía móvil, y en la actualidad desarrolla una plataforma de entrega de contenidos destinada a operadores móviles. Esta plataforma, en su versión actual carece de los servicios de valor agregado complementarios, sólo consta de los servicios básicos de una plataforma de este tipo. A raíz de este problema se ha desarrollado un sistema de envío de postales virtuales a través de mensajes MMS como primer servicio de valor agregado de la plataforma anteriormente mencionada. Luego de un estudio intensivo de los sistemas similares existentes, tomando de cada uno sus mejores características y añadiendo otras que eran de interés para la nueva solución, se decidió implementar el sistema planteado, el cual les permite a los usuarios personalizar fácilmente su postal mediante una interfaz WAP. Al software desarrollado se le realizaron las pruebas de aceptación requeridas, las cuales cumplió satisfactoriamente, y también se le efectuó un análisis de factibilidad utilizando un método de estimación, que dió como resultado que la realización del proyecto era viable.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 ESTADO DEL ARTE	5
1.2 INTRODUCCIÓN A LA TELEFONÍA MÓVIL	8
1.2.1 Servicios de valor agregado	10
1.3 METODOLOGÍA DE DESARROLLO UTILIZADA.....	12
1.3.1 ¿Por qué <i>Extreme Programming</i> ?	12
1.4 PLATAFORMA DE DESARROLLO UTILIZADA.....	14
1.5 TECNOLOGÍAS UTILIZADAS	15
1.5.1 Enterprise Java Beans.....	15
1.5.2 Java Architecture for XML Binding.....	16
1.5.3 Wireless Markup Language	17
1.5.4 WURFL-WALL.....	17
1.5.5 JSP y <i>Servlet</i>	19
1.5.6 SAMS-M	19
1.6 SERVIDOR DE APLICACIONES UTILIZADO	20
1.7 GESTOR DE BASE DE DATOS UTILIZADO	21
1.8 HERRAMIENTAS DE DESARROLLO UTILIZADAS.....	22
1.8.1 Entorno Integrado de Desarrollo.....	22
1.8.1.1 ¿Por qué Eclipse?.....	22
1.8.1.2 JBoss IDE	23

1.8.2 Control de versiones.....	23
1.8.2.1 SVN.....	24
1.8.2.2 Subclipse	24
1.9 CONCLUSIONES	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	25
2.1 OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN	25
2.2 OBJETO DE AUTOMATIZACIÓN.....	25
2.3 PROPUESTA DE SISTEMA.....	26
2.3.1 Flujo de procesos del sistema propuesto	29
2.4 ANÁLISIS COMPARATIVO DE OTRAS SOLUCIONES EXISTENTES CON LA PROPUESTA.....	29
2.5 CONCLUSIONES	32
CAPÍTULO 3: EXPLORACIÓN Y PLANIFICACIÓN	33
3.1 FASE DE EXPLORACIÓN	33
3.1.1 Historias de usuario	33
3.2 FASE DE PLANIFICACIÓN.....	36
3.2.1 Estimación de esfuerzo por historias de usuario	36
3.2.2 Plan de iteraciones	37
3.2.2.1 Primera interacción	37
3.2.2.2 Segunda iteración	37
3.2.2.3 Tercera iteración	38
3.2.3 Plan de Entrega.....	40
3.3 CONCLUSIONES	40
CAPÍTULO 4: DISEÑO, IMPLEMENTACIÓN Y PRUEBA.....	41
4.1 FASE DE DISEÑO.....	41
4.2 FASE DE IMPLEMENTACIÓN	42
4.2.1 Primera iteración.....	42

4.2.1.1 Tareas de las historias de usuario desarrolladas en la primera iteración	42
4.2.2 Segunda iteración	46
4.2.2.1 Tareas de las historias de usuario desarrolladas en la segunda iteración	46
4.2.3 Tercera iteración	48
4.2.3.1 Tareas de las historias de usuario desarrolladas en la tercera iteración	48
4.3 FASE DE PRUEBA	54
4.3.1 Pruebas de Aceptación	54
4.4 CONCLUSIONES	55
CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD	56
5.1 CARACTERÍSTICAS DEL PROYECTO	56
5.1.2 Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombre y costo	58
5.2 BENEFICIOS TANGIBLES E INTANGIBLES	60
5.3 ANÁLISIS DE COSTO	60
5.5 CONCLUSIONES	60
CONCLUSIONES	61
RECOMENDACIONES	62
BIBLIOGRAFÍA	63
ANEXOS	67
GLOSARIO	79

INTRODUCCIÓN

La tecnología móvil está creciendo rápidamente en alcance e importancia, proporcionando un acceso extenso e insólito a las redes de computadoras, incluyendo Internet. A diferencia de esta última, la telefonía móvil surge como una tecnología de voz operada desde un modelo de negocio de provisión de servicios integrado por operadoras, fabricantes, productores de contenido y mediadores. Dicho modelo implica una centralización jerárquica de la gestión y la inversión en infraestructuras que, a su vez, somete la innovación a la rentabilidad, garantizando unos estándares de calidad y seguridad.

De esta forma ha llegado a convertirse en una prestación básica para cualquier usuario por las grandes ventajas que ofrece en conectividad, movilidad, reducción de costos de infraestructura, rapidez y facilidad de despliegue.

El crecimiento que ha mostrado la telefonía móvil en los últimos años tiene su explicación en diversos motivos: la actualización tecnológica de las redes de los operadores, la proliferación de nuevos servicios y el abaratamiento de las comunicaciones debido a la fuerte competencia que acontece en la gran mayoría de los países, a pesar del proceso de concentración que vive el mercado. Estos factores confluyeron para que los servicios móviles expandieran su alcance hacia clases sociales que, tan sólo diez años atrás, no imaginaban contar con un dispositivo móvil. En este contexto de expansión, las comunicaciones vía telefonía celular han ido sustituyendo a las tradicionales de voz fija. A su vez, los fabricantes de terminales incorporan a los celulares nueva tecnología con funcionalidades para la transmisión de datos, lo cual posibilita a los operadores prestar nuevos servicios basados en datos móviles.

La telefonía móvil posee una amplia variedad de servicios de datos, entre ellos los servicios de mensajería, que reportan ganancias millonarias a los operadores. Estos servicios han abierto un mercado muy provechoso a nivel mundial para empresas proveedoras de Servicios de Valor Agregado (*VASP*). Para estas empresas las aplicaciones de mensajería representan una alta

fuentes de ingresos, debido a que es posible desarrollar múltiples servicios de mensajes tales como chat, horóscopo, postales, estado del tiempo, votaciones, noticias, correo electrónico, chequeo del estado de cuentas bancarias, entre otros y todos estos productos tienen una increíble aceptación entre los clientes.

Debido a esto muchas empresas han enfocado gran parte de sus esfuerzos, hacia el desarrollo de servicios encaminados a usuarios móviles, brindándoles la posibilidad de realizar cualquier actividad a través del teléfono.

En la actualidad la entidad Procyon lleva a cabo el desarrollo de una plataforma destinada a operadores móviles (*MobiStore*), la cual debe contar con los servicios básicos de una aplicación de este tipo, ya sea control de envío y recepción de mensajes, control de usuarios así como una serie de servicios avanzados. Debido a la importancia que tiene que dicha plataforma brinde la mayor cantidad de servicios, con el objetivo de entrar a competir en el mercado y aumentar su valor en el mismo, se ha decidido desarrollar un sistema que brinde el servicio de envío de postales como uno de los servicios avanzados que esta proporcionará. Esta situación problemática arrojó el siguiente problema de investigación.

Problema:

¿Cómo desarrollar un sistema para el servicio de envío de postales virtuales entre dispositivos móviles?

Para darle solución al mismo se plantea como **objetivo general** la implementación de un sistema que posibilite la personalización y envío de postales virtuales.

De aquí se derivan los siguientes **objetivos específicos**:

- Implementar una aplicación servidora que maneje todo lo referente a la solicitud del servicio de envío de postales y se encargue del envío de las postales MMS a los destinatarios.

- Desarrollar una interfaz Wap amigable para la personalización de las postales por parte del usuario.

El **objeto de estudio** de este trabajo es el sistema que le brinda el servicio de envío de postales a través de dispositivos móviles a la plataforma que se desarrolla actualmente en Procyon.

El **campo de acción** queda enmarcado en la plataforma *MobiStore* que se desarrolla en la entidad Procyon.

Las tareas desarrolladas para cumplir los objetivos son:

- Análisis de los sistemas existentes a nivel mundial que implementan este servicio de envío de postales a través de dispositivos móviles.
- Estudio y selección de las tecnologías existentes para el desarrollo de un sistema de este tipo.
- Definición de las historias de usuario.
- Conformación del plan de entrega.
- Conformación de las iteraciones del proyecto.
- Implementación de la aplicación servidora.
- Diseño e implementación de la interfaz *Wap*.

El presente trabajo está conformado por cinco capítulos, en los cuales se abordará todo el estudio realizado, así como la descripción de la solución propuesta.

En el Capítulo 1 se tratan los principales conceptos y términos abordados en la investigación, se realiza un análisis de los servicios de valor agregado para móviles y se abordan tecnologías, herramientas y metodología utilizadas en el desarrollo de la solución. En el Capítulo 2 se plantean los objetivos estratégicos de la entidad Procyon, se explican los procesos que serán objetos de automatización, así como la propuesta de sistema. En el Capítulo 3 se describen las fases de exploración y planificación propias de la metodología de desarrollo utilizada para la implementación del sistema que se propone. En el Capítulo 4 se describen las fases de diseño, implementación y prueba propias de la metodología de desarrollo utilizada. En el Capítulo 5 se realiza un estudio de factibilidad para la realización del sistema propuesto.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se tratarán los principales conceptos y términos abordados en la investigación, se realizará un análisis de los servicios de valor agregado (VAS) para móviles y se abordarán tecnologías, herramientas y metodología utilizadas en el desarrollo de la solución.

1.1 Estado del Arte

En el mundo existen viarias empresas que se dedican a prestar servicios de valor agregado para la telefonía móvil. Muchas de estas facilitan un servicio de envío de postales, en el cual las tarjetas postales escogidas por los usuarios se imprimen y la entrega a los destinatarios se hace a través del correo postal. A continuación se detallan algunos de estos servicios de envío de postales:

Movistar: El servicio de postales de *Movistar* funciona a partir de un mensaje multimedia (MMS) que se envía a un número designado. Este mensaje debe llevar la foto o imagen que se desea enviar, además se introduce el saludo personal, la palabra “dirección” y la dirección del destinatario con todos los detalles, nombre y apellidos, nombre de la calle, piso, código postal, ciudad y país. La empresa se encarga de imprimir la foto y enviarla por correo postal. La tarjeta postal es entregada en pocos días en la casa del destinatario indicado. En la figura 1 se muestra la experiencia de cliente con este servicio (Telefónica Movistar, 2007).

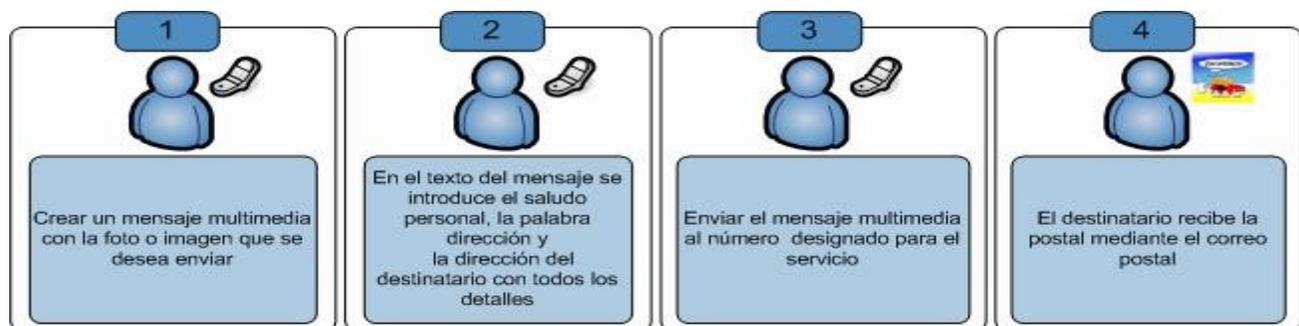


Figura 1. Experiencia del cliente (*Movistar*)

Pannon: El servicio de postales de *Pannon* funciona a partir de una foto tomada por el usuario en la mejor resolución del móvil, esta foto es adjuntada en un mensaje multimedia (*MMS*) y enviada al número designado para el servicio. Inmediatamente luego de ser enviado el mensaje, el usuario recibe un mensaje de servicio (*Wap Push*) con la dirección a donde debe conectarse para personalizar la postal. Ya dentro del sitio *Wap* el usuario, mediante un formulario, introduce el saludo personal y la dirección del destinatario con todos los detalles, nombre y apellidos, nombre de la calle, piso, código postal, ciudad y país. La empresa se encarga de imprimir la foto y enviarla por correo postal. La tarjeta postal es entregada en pocos días en la casa del destinatario indicado. En la figura 2 se muestra la experiencia de cliente con este servicio (Pannon, 2007).

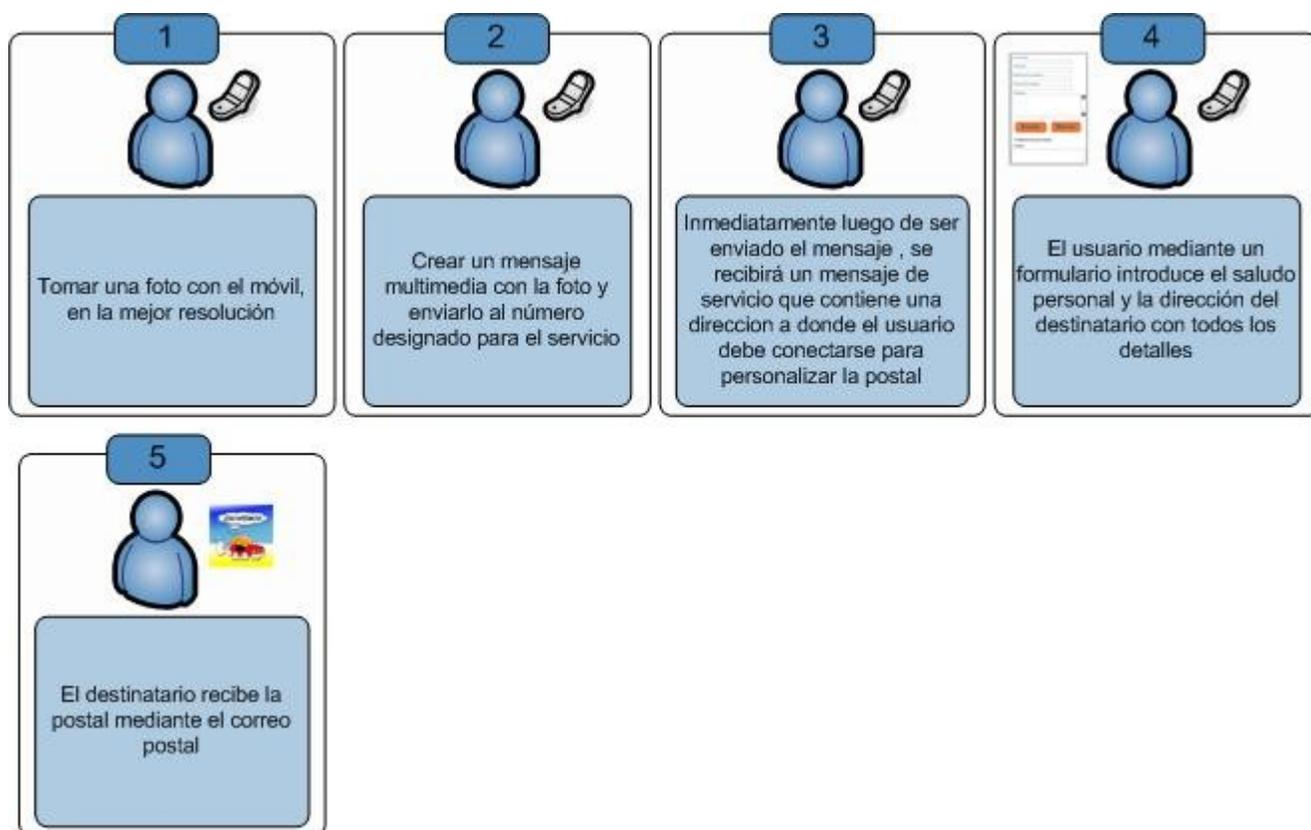


Figura 2. Experiencia del cliente (*Pannon*)

Swisscom: El servicio de postales de *Swisscom* funciona a partir de una foto tomada por el usuario en la mejor resolución del móvil, esta foto es adjuntada en un mensaje multimedia (*MMS*) y enviada al número designado para el servicio. Inmediatamente luego de ser enviado el mensaje, el usuario recibe un mensaje de servicio (*Wap Push*) con la dirección a donde debe conectarse para personalizar la postal. Ya dentro del sitio *Wap* el usuario, mediante un formulario, escoge el estilo de la postal, introduce el saludo personal y la dirección del destinatario con todos los detalles, nombre y apellidos, nombre de la calle, piso, código postal, ciudad y país. Al usuario, luego de personalizar la postal, se le muestra una vista previa de la postal. La empresa se encarga de imprimir la foto y enviarla por correo postal. La tarjeta postal es entregada en pocos días en la casa del destinatario indicado. En la figura 3 se muestra la experiencia de cliente con este servicio (*Swisscom Mobile, 2007*).

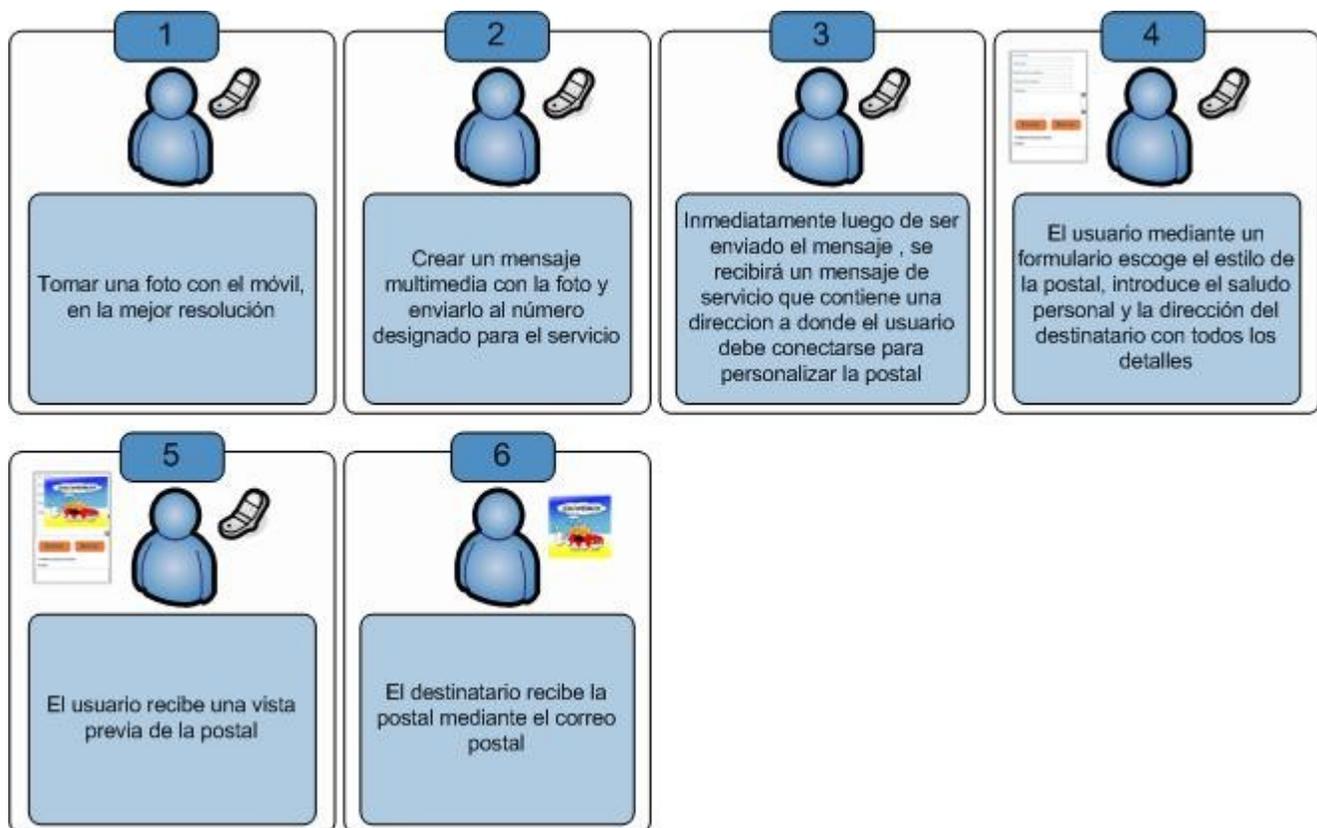


Figura 3. Experiencia del cliente (*Swisscom*)

1.2 Introducción a la telefonía móvil

A pesar de que la telefonía celular fue concebida estrictamente para la voz, hoy es capaz de ofrecer otro tipo de servicios tales como datos, audio y video, en un principio con algunas limitaciones; sin embargo, la telefonía inalámbrica en la actualidad hace posible el desarrollo de aplicaciones con más prestaciones debido al aumento del ancho de banda de las redes y la evolución de los dispositivos móviles.

Uno de los estándares más difundidos a nivel mundial es el Sistema Global para Comunicaciones Móviles (*Global System for Mobile Communications* o GSM). GSM soporta múltiples servicios, entre ellos se destaca el Servicio de Mensajería Corta (*Short Message System* o SMS) el cual es un servicio de telefonía móvil universalmente aceptado, que posibilita la transmisión de mensajes cortos (SM) alfanuméricos entre usuarios móviles y sistemas externos como correo electrónico, *paging* y sistemas de correos de voz (International Engineering Consortium, 2007).

Un Mensaje Corto (*Short Message* o SM) es un mensaje de texto breve enviado desde un dispositivo móvil MO (*Mobile Originated*), hacia un dispositivo móvil MT (*Mobile Terminated*), a través de un SMS. Un SM estándar consiste en 160 caracteres alfanuméricos máximo, aunque se pueden enviar mensajes con mayor cantidad de caracteres utilizando compresión de datos (TechTarget, 2005).

Los SMS no se envían punto a punto, por lo cual, se necesita una entidad en la red inalámbrica encargada del proceso. Esta entidad conocida, como Centro de Servicio de Mensajes Cortos (*Short Message Service Center* o SMSC), que es responsable de la transmisión, almacenamiento y reenvío de SM entre Entidades de Mensajería Corta (*Short Message Entity* o SME) (TechTarget, 2005).

El estándar GSM continuó su desarrollo y con la versión 97 de este estándar desarrollado por 3GPP (3rd *Generation Partnership Project*) surgió el Servicio General de Radio-Paquetes (GPRS), donde se adicionaron capacidades de empaquetamiento de datos. Con el surgimiento

de GPRS se adicionaron nuevos servicios, entre los más conocidos se puede mencionar el Servicio de Mensajes Multimedia, diseñado específicamente para GPRS (International Engineering Consortium, 2007).

Un mensaje multimedia se envía a través del Servicio de Mensajes Multimedia (*Multimedia Message Service* o MMS), que es una tecnología de comunicaciones que permite a los usuarios intercambiar mensajes multimedia entre teléfonos móviles y otros dispositivos. MMS es una extensión de SMS, que define una manera de enviar y recibir, casi instantáneamente, mensajes que incluyen imágenes, audio y videos clip además de texto (TechTarget, 2005).

El servicio MMS puede replicar el comportamiento de un correo electrónico con archivos adjuntos, y como valor añadido y diferenciador permite utilizar presentaciones animadas. Esto es posible mediante la inclusión en el mensaje de un elemento de presentación llamado SMIL (*Synchronized Multimedia Integration Language*) el cual es un simple pero poderoso lenguaje de etiquetas para especificar como y cuando mostrar clips (TechTarget, 2005).

Este lenguaje especifica de qué forma, en qué momento y en qué lugar las diversas partes del mensaje deben ser presentadas al usuario. Es un lenguaje basado en XML, que permite a los desarrolladores mezclar distintos medios para ser presentados y sincronizados unos con otros. El mensaje puede estar compuesto por varias páginas consecutivas, denominadas diapositivas (*slides*), con un tiempo de presentación de cada página que se determina al componer el mensaje. Cada una de ellas puede contener uno o varios de los siguientes campos: texto, imágenes (simples o animadas), sonido e incluso video en los terminales que lo soporten (TechTarget, 2005). (Ver figura 4)

Otro de los servicios relevantes que soporta el estándar GSM y por consiguiente GPRS son las Notificaciones WAP (*WAP Push*). Estas típicamente hacen referencia a un MMS, a un correo electrónico o a una descarga de contenidos como imágenes, juegos, entre otros. Una notificación Wap puede estar formada por un mensaje de texto y un URL, o solamente por un URL, donde el

teléfono móvil receptor del mensaje muestra el texto, en caso de que exista, y provee al usuario la opción de conectarse al URL con solo presionar una tecla (Ericsson, 2006).

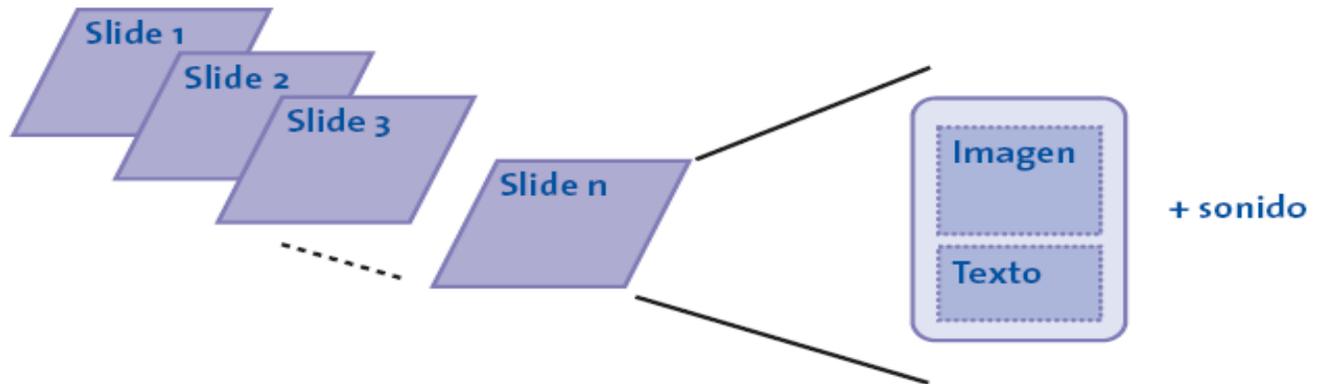


Figura 4. Esquema de composición de páginas en un mensaje MMS

1.2.1 Servicios de valor agregado

Los mensajes SMS, MMS y WAP Push, fueron utilizados de forma tradicional hasta el instante que diferentes empresas desarrollaron los primeros servicios de valor agregado (VAS) en este campo. A partir de este momento, un simple mensaje SMS que se enviaba desde un teléfono celular a otro se convirtió en toda una industria de aplicaciones con diversos servicios que incluían noticias, votaciones, horóscopo, chat, reporte del tiempo, entre muchos otros. Con el surgimiento de los MMS, los usuarios se beneficiaron de nuevas aplicaciones que combinan imágenes, sonidos, videos y texto, donde experimentaron una experiencia más agradable.

El mundo de los servicios de valor agregado es amplio y heterogéneo, abarcando distintas necesidades y diversas porciones del mercado de telefonía móvil. En tal sentido, la gama de servicios es grande y puede estar orientada al entretenimiento o a otros temas más relevantes para la actividad laboral y cotidiana.

Una aplicación o un servicio de valor agregado basado en MMS es la instrumentación que se hace del servicio básico MMS para llevar a cabo un modelo de negocio habilitado por las capacidades de esta tecnología. La variedad y utilidad de las aplicaciones y servicios disponibles redundarán en el incremento del ingreso medio por usuario de los operadores móviles (Telefónica I+D, 2005).

Los distintos tipos de servicios se pueden agrupar en cuatro bloques:

1. **Servicios basados en el contenido.** No poseen ningún tipo de procesamiento, encapsulado o tratamiento; en otras palabras, son los mensajes intercambiados entre los usuarios.
2. **Servicios básicos.** Pueden ser de dos tipos:
 - a) *Aplicaciones de entretenimiento.* Son la evolución natural de los servicios actuales de descarga de los elementos de entretenimiento para personalizar los terminales de los usuarios (logos, melodías, refrescadores de pantalla).
 - b) *Aplicaciones de almacenamiento de imágenes fijas.* En este caso existen dos vertientes: la publicación de fotografías u otros elementos en álbumes web, o el almacenamiento en un espacio privado del usuario para evitar el consumo de recursos en el terminal.
3. **Servicios complementarios.** Permiten la creación de contenidos por el usuario como el envío de tarjetas postales o felicitaciones, y los compositores de mensajes completos con repositorios de contenidos elementales como imágenes y sonidos.
4. **Servicios de información multimedia.** Son similares a los servicios de alertas SMS, pero explotan las capacidades ofrecidas por la tecnología MMS. Entre ellos se encuentran los servicios de suscripción, los servicios de noticias, los servicios financieros, de información

meteorológica, los anuncios musicales o videos promocionales, los servicios de envío de fotografías capturadas por *webcams* y los juegos interactivos.

1.3 Metodología de desarrollo utilizada

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto teniendo en cuenta recursos técnicos, humanos y tiempo de desarrollo.

Para la puesta en marcha y desarrollo del sistema se decidió utilizar la metodología de desarrollo *Extreme Programming* (XP). Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Beck, 1999).

1.3.1 ¿Por qué *Extreme Programming*?

Se tomó XP como metodología de desarrollo, debido a que se adapta en gran medida al proyecto a desarrollar, tanto al tipo de proyecto como a las condiciones de trabajo. A continuación se exponen algunas de las razones más importantes que evidencian esta afirmación:

- El proyecto es pequeño. XP está concebida para ser utilizada dentro de proyectos pequeños.

- Poca disponibilidad de personal. El sistema debe ser desarrollado por dos personas solamente, por lo cual no es factible que existan muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos pequeños (de 2 a 12 personas) con pocos roles, y que los miembros del equipo puedan intercambiar responsabilidades en un momento determinado.
- No existe un contrato previo para especificar tiempo, recursos y alcance. Para la implementación del sistema no se dispone de un contrato con un presupuesto ni un alcance previamente definidos.
- El cliente forma parte del equipo de desarrollo. El cliente en este caso es la entidad donde se radica. Mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades.
- Los requisitos del sistema cambian frecuentemente. Con el rápido desarrollo que presenta la telefonía móvil, la aparición de nuevas tendencias y tecnologías en este campo hacen que el sistema se adapte a las mismas. Uno de los principios básicos de XP es que el cambio frecuente de los requerimientos es algo normal en el proceso de desarrollo. Esta metodología se adapta perfectamente a los proyectos cuyos requerimientos cambian a menudo.
- El riesgo de desarrollo es alto debido al corto tiempo de entrega planeado y los continuos cambios de requerimientos. XP está diseñada para mitigar los riesgos en proyectos con estas características.
- XP enfatiza la comunicación de los programadores a través del código, utilizando líneas directivas bien establecidas para la codificación. Desde sus inicios la empresa cuenta con una línea directiva para la codificación.

1.4 Plataforma de desarrollo utilizada

Para el desarrollo de aplicaciones empresariales existen varias tendencias hoy en día, una de las de más aceptación es la plataforma Java EE (*Java Enterprise Edition*) de Sun Microsystems.

Para el desarrollo de la aplicación se escogió la plataforma Java EE debido al vínculo que existe entre ésta y las aplicaciones para la telefonía móvil. A nivel mundial la plataforma Java EE es la más usada para el desarrollo de este tipo de proyectos. Otra de las causas que apuntan al uso de esta plataforma por encima de otras como Microsoft .NET es la experiencia del equipo de desarrollo en su uso, además de que la utilización de Java EE es una política establecida de la entidad Procyon.

La plataforma Java EE es una tecnología para el diseño de componentes de transacción escalables y basados en servidor y comprende un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones distribuidas.

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para componentes. Estas incluyen Enterprise Java Beans (EJB), Servlets, Portlets (siguiendo la especificación de Portlets Java), Java Server Pages y varias tecnologías de servicios web. Esto permite al desarrollador crear una aplicación empresarial escalable, portable entre plataformas e integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, lo cual significa que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes, en lugar de tareas de mantenimiento de bajo nivel (Jendrock y otros, 2006).

Uno de los beneficios de Java EE como plataforma es que es posible empezar con poco o ningún costo. La implementación Java EE de Sun Microsystems puede ser descargada gratuitamente, así como también el servidor de aplicaciones, y hay muchas herramientas de código abierto disponible para extender la plataforma o para simplificar el desarrollo.

1.5 Tecnologías utilizadas

En este epígrafe se detallan las tecnologías utilizadas en el desarrollo del sistema. Estas tecnologías fueron seleccionadas de acuerdo a la plataforma de desarrollo escogida y las ventajas que estas presentan, además de que la utilización de estas es una política establecida por la entidad Procyon.

1.5.1 Enterprise Java Beans

De las especificaciones Java EE para componentes se utilizó EJB para el soporte de la capa de negocio y persistencia. EJB es una de las API que forman parte del estándar de construcción de aplicaciones empresariales Java EE. Los EJB proporcionan un modelo de componentes distribuido estándar para el lado del servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes nos permite que éstos sean flexibles y sobre todo reutilizables. Existen tres tipos de EJB:

- **EJB de Entidad** (*Entity Beans*). Su objetivo es encapsular los objetos del lado del servidor que almacena los datos. Los EJB de entidad presentan la característica fundamental de la persistencia (Patel y otros, 2006).

- **EJB de Sesión** (*Session Beans*) gestionan el flujo de la información en el servidor. Generalmente sirven a los clientes como una fachada de los servicios proporcionados por otros componentes disponibles en el servidor (Patel y otros, 2006).
- **EJB dirigidos por mensajes** (*Message-driven Beans*). Presentan funcionamiento asíncrono. Usando Java Messaging System (JMS), se suscriben a un tema (*topic*) o a una cola (*queue*) y se activan al recibir un mensaje dirigido a dicho tema o cola. No requieren de su instanciación por parte del cliente (Patel y otros, 2006).

1.5.2 Java Architecture for XML Binding

JAXB (*Java Architecture for XML Binding*) suministra una API y herramientas para automatizar el mapeo entre documentos XML y objetos Java. Brinda a los desarrolladores de Java una estándar y eficiente vía de mapeo entre documentos XML y código Java (Patel y otros, 2006).

JAXB proporciona a una manera rápida, conveniente de crear uniones de dos vías entre los documentos XML y los objetos Java. Dado un esquema, que especifica la estructura de los datos XML, el compilador JAXB genera un conjunto de clases de Java que contienen todo el código para analizar los documentos XML basados en el esquema. Una aplicación que utilice las clases generadas puede construir un árbol de objetos Java que representa un documento XML, manipular el contenido del árbol, y regenerar los documentos del árbol, todo ello en XML sin requerir que el desarrollador escriba código de análisis y de proceso complejo (Ort y Mehta, 2003).

Los desarrolladores de Java que usan JAXB son más productivos porque pueden escribir menos código ellos mismos y no tienen que ser expertos en XML. JAXB hizo más fácil para los desarrolladores agrandar sus aplicaciones con XML y tecnologías de servicios WEB (Palos, 2004).

Usar JAXB para una aplicación de proceso de datos tiene muchos beneficios porque usa tecnología Java y XML, garantiza la validez de los datos, es rápido, fácil de usar, puede restringir datos, es personalizable y extensible (Palos, 2004).

1.5.3 Wireless Markup Language

Para el desarrollo de interfaces estándares *Wap* se utiliza el lenguaje WML (*Wireless Markup Language*) cuyo origen es el XML (*eXtensible Markup Language*). Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDA) dotados de tecnología *Wap*. Es una versión reducida del lenguaje HTML que facilita la conexión a Internet de dichos dispositivos y que además permite la visualización de páginas web en dispositivos inalámbricos que incluyan la tecnología WAP. La visualización de la página dependerá del dispositivo que se use y de la forma en que este interprete el código, ya que varían entre sí.

WML es un metalenguaje, lo que implica que además de usar etiquetas predefinidas se pueden crear componentes propios y tiene ciertas similitudes con otro lenguaje de etiquetas bastante conocido, el HTML (*Hypertext Markup Language*), utilizado para la creación de páginas web convencionales. Un consorcio formado por Nokia, Phone.com, Motorola y Ericsson, conocido como el WAP Forum, define la sintaxis, variables y elementos utilizados en WML. Algunos fabricantes han desarrollado capacidades adicionales a este estándar. Al igual que el HTML se sirve de un lenguaje de script como Javascript para dotar de cierto dinamismo a sus documentos, WML dispone del WMLS que es un lenguaje bastante similar al Javascript, pero con algunas diferencias (Routt, 2003).

1.5.4 WURFL-WALL

Como consecuencia de la gran diversidad de dispositivos móviles que existen hoy en día debido a las distintas dimensiones de pantallas, formato de imágenes y audio que soportan, en el

mundo se genera una gran problemática para los desarrolladores en la creación de páginas o sitios *Wap*. Para dar solución a este problema surge la combinación de WURFL-WALL.

WURFL (*Wireless Universal Resource File*) y WALL (*Wireless Abstraction Library*) son un conjunto de ficheros y librerías orientadas a facilitar el desarrollo de aplicaciones *Wap* para clientes que consten de poco procesamiento, con una gran portabilidad entre distintos tipos de navegadores embebidos dentro de clientes móviles (navegadores *Wap* que soporten WML, cHTML, iMode, XHTML) (Passani, 2002).

WURFL en realidad es un fichero XML donde se expresan, de manera compacta, las capacidades de diversos dispositivos. WALL es la combinación de una API Java, y una serie de librerías de etiquetas (*taglibs* JSP) que permiten construir páginas JSP portables entre dispositivos.

Mediante su uso, se puede construir aplicaciones que pueden ser visualizadas en diferentes entornos, adaptadas a las capacidades disponibles en cada dispositivo. Así, el mismo contenido, ya sea imagen, animación o sonido, puede emplearse en distintos dispositivos sin la necesidad de tener varias versiones de la aplicación o codificar usando muchos bloques condicionales en función del elemento.

El uso típico de WALL es la utilización de la librería de etiquetas incluida para la creación de vistas JSP, asociadas al resto del modelo de la arquitectura. Su uso es compatible con cualquier otra librería de etiquetas, como las que proporciona JSTL (Guemes, 2006).

Otro uso de WALL es el de la API que nos informa de las características de los dispositivos directamente. Así, dentro del código Java, se puede analizar si un dispositivo dispone de una determinada característica (tamaño de pantalla, color, y demás) y en función de ello adaptar la funcionalidad de la aplicación a los dispositivos (Passani, 2002).

Las capacidades de WALL dependen del fichero de WURFL, que se actualiza constantemente. Conviene disponer de la última versión que cubra los dispositivos que se utilizarán en el proyecto.

1.5.5 JSP y Servlet

Un *Servlet* es un objeto que se ejecuta dentro del contexto de un servidor o contenedor Java EE. Fue especialmente diseñado para ofrecer contenido dinámico desde un servidor web, generalmente es HTML. Los *servlets* forman parte de la plataforma Java EE. Un *servlet* es un objeto Java que implementa la interfaz `javax.servlet.Servlet` o hereda alguna de las clases más convenientes para un protocolo específico (ejemplo: `javax.servlet.HttpServlet`). Al implementar esta interfaz el *servlet* es capaz de interpretar los objetos de tipo `HttpServletRequest` y `HttpServletResponse` quienes contienen la información de la página que invocó al *servlet* (Hall y Brown, 2003).

Entre el servidor de aplicaciones (o contenedor web) y el *servlet* existe un contrato que determina cómo han de interactuar. La especificación de éste se encuentra en los JSR (*Java Specification Requests*) del JCP (*Java Community Process*) (Perry, 1999).

JSP (*Java Server Pages*) es una tecnología Java que permite a los programadores generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Las páginas JSP permiten que tanto el código Java como algunas acciones predefinidas puedan ser incrustadas en el contenido estático del documento web (Bayern, 2001).

1.5.6 SAMS-M

La existencia de múltiples protocolos de comunicación y diferentes servicios de mensajería como SMS, MMS y *WAP Push* representa un problema para los desarrolladores. La industria necesitaba un estándar que regulara las interfaces para el desarrollo de aplicaciones con servicios de SMS, MMS y *WAP Push* con una arquitectura que permitiera agregar implementaciones para cada uno de los protocolos de comunicación con Centros de Mensajería.

Este estándar fue llamado API Servidora para Servicios Móviles de Mensajería (*Server API for Mobile Services-Messaging* o SAMS-M) y no fue desarrollado hasta el 2004, como una alternativa a la variedad de soluciones propietarias que existían en el mercado donde cada una de estas definía cuales protocolos implementar (Sun Microsystems, 2004).

SAMS-M representa el primer estándar que se define para homogenizar las interfaces de envío y recepción para las aplicaciones desarrolladas por los VASP. En el desarrollo de la aplicación se utilizó la versión de SAMS-M implementada en la entidad Procyon.

1.6 Servidor de aplicaciones utilizado

Como servidor de aplicaciones se utilizó JBoss AS. Este es un servidor de aplicaciones J2EE de código abierto implementado en Java. Al estar basado en Java, JBoss AS puede ser utilizado en cualquier sistema operativo que lo soporte.

JBoss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, que ofrece una plataforma de alto rendimiento para aplicaciones empresariales. Combinando una arquitectura orientada a servicios revolucionaria y con una licencia de código abierto (Monson-Haefel, 2006).

JBoss AS puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia (JBoss, 2007). Por este motivo es una plataforma popular entre los desarrolladores, vendedores independientes de software y, también, para grandes empresas.

En la entidad Procyon se ha utilizado este servidor de aplicaciones debido a sus características, llegándose a alcanzar alto nivel de experiencia en el desarrollo de aplicaciones utilizando JBoss AS. De esta forma el uso de este servidor de aplicaciones se ha convertido en una estrategia para la entidad.

1.7 Gestor de Base de Datos utilizado

El gestor de base de datos utilizado fue MySQL. Este posee una gran velocidad a la hora de realizar operaciones, lo que le hace uno de los gestores que ofrece mayor rendimiento. Su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema (MySQL, 2006). Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que se está ejecutando (Kofler, 2005).

Proporciona motores de almacenamiento transaccional y no transaccional. Utiliza tablas muy rápidas (MyISAM) con la compresión del índice. Es relativamente fácil agregar otros motores de almacenaje. Esto es útil si se desea proporcionar un interfaz de SQL para una base de datos interna. Es un sistema muy rápido de asignación de memoria basado en hilos (MySQL, 2006).

En cuanto a la seguridad es un sistema de privilegio y de contraseña muy flexible y segura, y que permite la verificación basada en *host*. Las contraseñas son seguras porque se cifra todo el tráfico de la contraseña cuando se conecta con un servidor (MySQL, 2006).

La escalabilidad y límites de este gestor de base datos es muy amplio: maneja grandes bases de datos, estas pueden contener 50 millones de records, 60.000 tablas y cerca de 5.000.000.000 de filas (MySQL, 2006).

Posee hasta 64 índices por tabla. Cada índice puede consistir de 1 a 16 columnas o partes de columnas. El tamaño máximo del índice es 1000 bytes (767 para InnoDB). Un índice puede utilizar un prefijo de una columna para los tipos de la columna CHAR, VARCHAR, BLOB, o TEXT (MySQL, 2006).

En la entidad Procyon se ha utilizado este Gestor de Base de Datos debido a sus características, llegándose a alcanzar alto nivel de experiencia en el desarrollo de aplicaciones utilizando MySQL. De esta forma el uso de este Gestor de Base de Datos se ha convertido en una estrategia para la entidad.

1.8 Herramientas de desarrollo utilizadas

En este epígrafe se fundamentan las herramientas de desarrollo utilizadas para la realización del sistema. Estas herramientas fueron escogidas debido al nivel de experiencia en su uso adquirido por el equipo de desarrollo, además de que el uso de estas herramientas es una política establecida de la entidad Procyon.

1.8.1 Entorno Integrado de Desarrollo

Se conoce como IDE a un entorno integrado de desarrollo, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. En este proyecto se utilizó Eclipse como IDE para la implementación del sistema (Wikipedia, 2007).

1.8.1.1 ¿Por qué Eclipse?

Eclipse es una poderosa plataforma de código abierto que permite integrar diferentes aplicaciones para construir un entorno integrado de desarrollo (IDE). Su principal aplicación es JDT, herramienta para crear aplicaciones en Java, además de una serie de *plug-ins* que se encuentran acompañando a la plataforma, tales como: Ant, Compare, Core, CVS, Debug, Help, JDT, Jface, Releng, Scripting, Search, SWT, Text, UI, Update, Team y WebDAV (IBM, 2007).

El usuario se comunica con Eclipse por intermedio del marco de trabajo que se inicia al ejecutar Eclipse (*workbench*). El *workbench* es la interfaz de usuario de la plataforma que está compuesto de un conjunto de vistas, editores y perspectivas. Las herramientas integradas a Eclipse operan

en archivos del espacio de trabajo (*workspace*) del usuario. El *workspace* consta de uno o más proyectos donde cada uno se mapea a un directorio especificado por el usuario en el sistema de ficheros (IBM, 2007).

Eclipse se considera como uno de los IDE más poderosos para el desarrollo de aplicaciones en Java, provee soporte mediante *plug-ins* para trabajar con entornos J2SE y J2EE utilizados en el desarrollo de este trabajo. Con la adición de estos *plug-ins* se puede facilitar la automatización de diferentes tareas para los desarrolladores, tales son los casos del *plug-in* JBoss IDE y de un *plug-in* que permite la administrar el control de versiones de un proyecto. Seguidamente abordaremos más sobre los mismos.

1.8.1.2 JBoss IDE

JBoss IDE es una serie integrada de *plug-ins* de Eclipse para construir aplicaciones empresariales, diseñado para ser usado con JBoss AS. Este entorno integrado de desarrollo se extiende de la plataforma Eclipse permitiéndoles a los programadores desarrollar, desplegar, probar y depurar sus aplicaciones basadas en JEMS (*Java Enterprise. Middleware System*) sin abandonar en ningún momento el IDE. De esta forma simplifica el ciclo de desarrollo, proporcionando un rápido desarrollo combinado con ningún gasto en licencias de software y reduciendo en gran medida el costo total en la creación de aplicaciones empresariales (Eclipse Magazine, 2007).

1.8.2 Control de versiones

El control de versiones es el proceso de gestión de las versiones o revisiones de todos los elementos de configuración que forman la línea base de un producto o una configuración del mismo. Para la automatización de este proceso se emplea básicamente SVN, herramienta libre que provee una serie funcionalidades necesarias para el control de versiones (Wikipedia, 2007).

1.8.2.1 SVN

Subversion (SVN) es un sistema de control de versiones que ofrece muchas funcionalidades como: integración con Eclipse, JDeveloper y NetBeans, presenta la mayoría de las facilidades de CVS (*Concurrent Version System*) y manejo eficiente de ficheros binarios (Tigris.org, 2007).

SVN se utiliza como servidor para el almacenamiento de los elementos de configuración. Para acceder a este servidor de manera fácil desde el Eclipse se utiliza un cliente denominado Subclipse.

1.8.2.2 Subclipse

Subclipse es un *plug-in* que integra el Eclipse con el SVN. Haciendo uso de este software se puede facilitar el control de versiones desde el propio IDE Eclipse de forma tal que se pueden realizar todas operaciones sobre los recursos que se encuentran bajo el control de versiones. Esta herramienta es libre y es ampliamente utilizada por los desarrolladores de software que utilizan el Eclipse como IDE a nivel mundial (Tigris.org, 2007).

1.9 Conclusiones

En este capítulo se trataron las principales conceptos abordados en la investigación, se realizó un análisis del estado de los servicios de valor agregado, se hizo un estudio sobre algunos de los servicios de envío de postales de mayor aceptación a nivel mundial y se detalló la metodología, plataforma, tecnologías y herramientas de desarrollo utilizadas en la construcción del sistema.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se plantean los objetivos estratégicos de la entidad Procyon, se explican los procesos que serán objetos de automatización, así como la propuesta de sistema para el servicio de envío de postales virtuales a través de dispositivos móviles. También se hace un análisis comparativo de las soluciones existentes con la nueva propuesta que se plantea en el trabajo.

2.1 Objetivos estratégicos de la organización

La entidad Procyon provee soluciones y servicios en el mundo de la telefonía móvil, prestando especial atención en el empleo de nuevas tecnologías de acceso a contenidos, nuevos modelos de dispositivos móviles así como nuevos formatos multimedia. Constantemente se actualiza la infraestructura tecnológica garantizando capacidad inmediata de respuesta a los distintos modelos de negocios que puedan presentar operadores, clientes y proveedores de contenido. El principal objetivo de la entidad es adentrarse en el mercado de las aplicaciones para móviles.

2.2 Objeto de automatización

Serán objeto de automatización los procesos que intervienen en el servicio de envío de postales MMS a través de dispositivos móviles. Estos se pueden desglosar de la siguiente forma: **Recepción de mensajes SMS**, proceso mediante el cual se recogen los mensajes de solicitud de servicio de los usuarios; **Procesamiento de solicitud**, proceso mediante el cual se crean las pre-orden que contienen todos los datos referentes a la solicitud de servicio; **Personalización de Postal**, proceso mediante el cual los usuarios pueden personalizar su postal de manera simple mediante una interfaz Wap; **Conformación de postales**, proceso mediante el cual se crean los

mensajes MMS que son enviados a los destinatarios escogidos por los usuarios cuando personalizan su postal; **Manejador de contenidos**, proceso mediante el cual se crean, modifican y eliminan los contenidos relacionados con las postales; **Manejador de reportes**, proceso mediante el cual se manejan los reportes que genera el sistema que son de interés a la hora de realizar el cobro del servicio.

2.3 Propuesta de Sistema

Se propone la puesta en marcha de un sistema que añada un nuevo servicio de valor agregado a la plataforma que se desarrolla actualmente en la entidad Procyon. Se decidió la implementación de una aplicación que permita a los usuarios de dicha plataforma el envío de postales virtuales a dispositivos móviles.

Para la organización fundamental del sistema se definió una arquitectura Java EE que tiene como objetivo principal aportar elementos que ayuden a la toma de decisiones. La arquitectura Java EE define un modelo de programación encaminado a la creación de n-capas (Jendrock y otros, 2006). La **Capa de cliente** representa el interfaz de usuario que maneja el cliente, la **Capa de presentación** representa el conjunto de componentes que generan la información que se representa en el interfaz de usuario del cliente, la **Capa de lógica de negocio** contiene los componentes de negocio reutilizables y componentes que permiten hacer más transparente el acceso a la **Capa de datos**. Esta última engloba nuestro sistema de información. Las ventajas de utilizar un modelo como este es que permite hacer modificaciones en cualquiera de las capas sin tener que interferir en las demás, además de obtener mejoras en cuanto a mantenimiento, extensibilidad y reutilización de componentes. La arquitectura del sistema se visualiza en la figura 5.

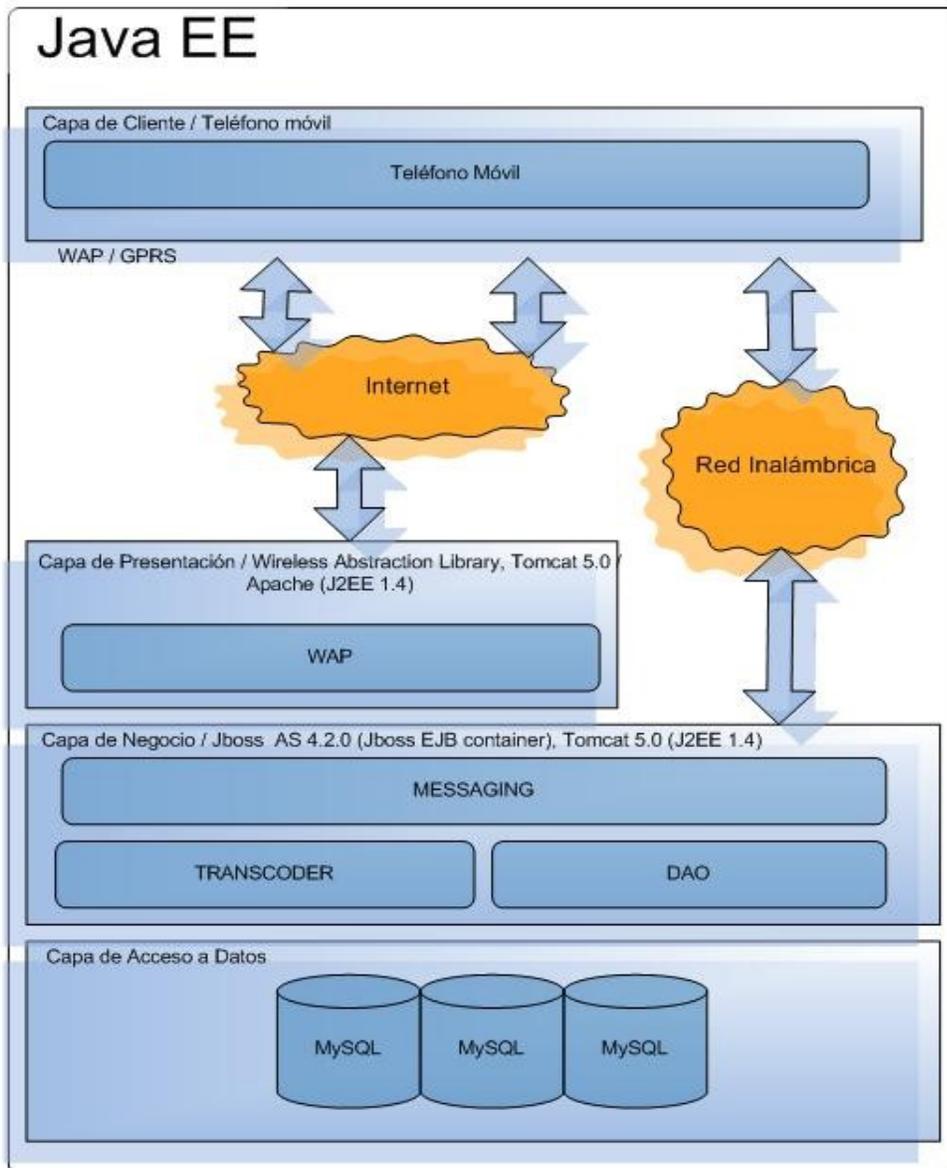


Figura 5. Arquitectura del Sistema

La aplicación está formada por dos módulos, los cuales se explican a continuación (ver figura 6):

- **mmspostcad-core**: Este módulo es el encargado del servicio de mensajería del sistema, además es donde se implementa toda la lógica del negocio desde que se recibe el

mensaje de solicitud de servicio hasta que se envían los mensajes multimedia a los destinatarios escogidos por los clientes del servicio de postales.

- **mmspostcard-wap**: Este módulo es el encargado de darle posibilidad al cliente del servicio de personalizar la postal a su gusto mediante una interfaz *Wap*. Aquí se escoge el formato de la postal y se insertan los datos que lleva la misma, además se muestra una vista previa de la postal al usuario.

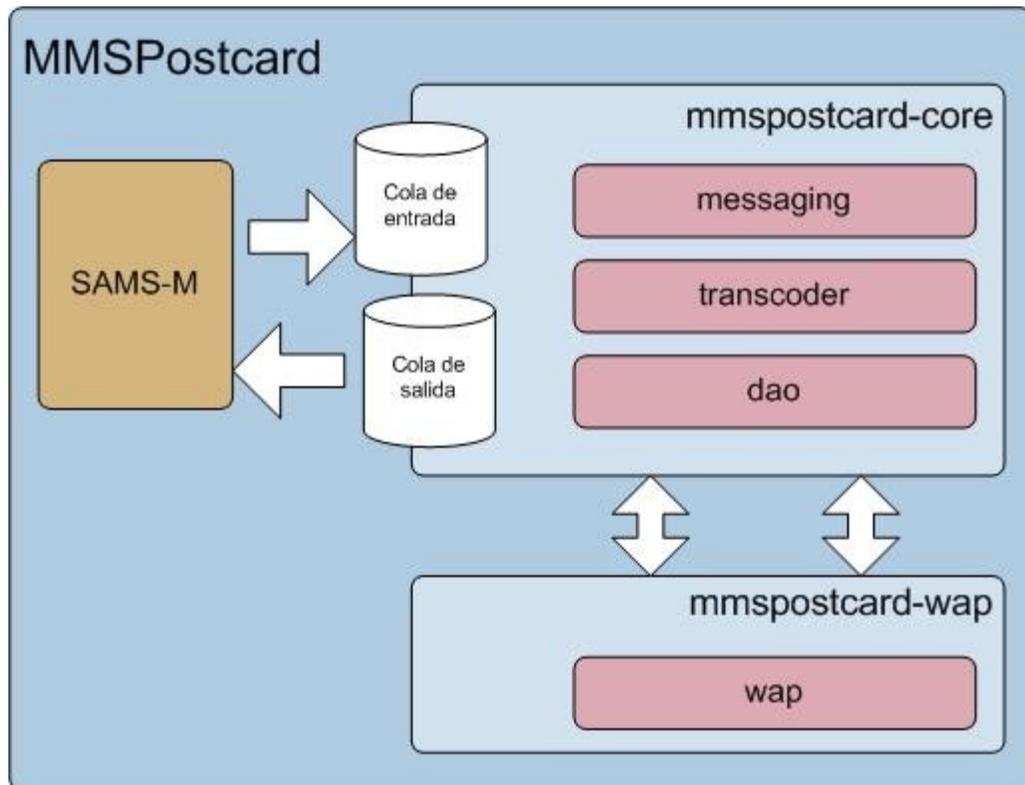


Figura 6. Módulos que componen el Sistema

2.3.1 Flujo de procesos del sistema propuesto

Cuando el usuario envía un SMS con una palabra reservada (por ejemplo: *postcard*) a un número previamente asignado, el módulo **mmspostcard-core** procesa este SMS y crea una pre-orden de servicio que es almacenada en la base de datos. Esta pre-orden contiene los siguientes campos: fecha de creación, fecha de expiración, guid (código que es generado a partir de la hora del sistema para la identidad de la orden), número telefónico del usuario y estado (el estado es una variable lógica que tiene valor “falso” cuando el usuario todavía no ha personalizado la postal y “verdadero” en caso contrario). Luego de realizar este proceso se crea un mensaje *Wap Push* y se envía al usuario la URL a donde debe conectarse para personalizar la postal. Esta URL lleva consigo el código de identidad de la orden para que, cuando el usuario acceda a personalizar la postal, el sistema pueda asociarle la orden correspondiente. El usuario accede a la interfaz *Wap* y personaliza su postal, escoge la categoría, selecciona la imagen que va a conformar la postal, inserta el mensaje de la misma y los destinatarios a los cuales va a ser enviada. Cuando estos datos son almacenados la pre-orden se convierte en una orden (cambia el estado inicial de la pre-orden) y es procesada nuevamente por el **mmspostcard-core**, se conforma el mensaje MMS que contiene la postal y es enviado a los destinatarios correspondientes. En la figura 7 se muestra el flujo de procesos de la aplicación.

2.4 Análisis comparativo de otras soluciones existentes con la propuesta

Durante la etapa inicial del desarrollo del sistema se analizaron algunos servicios de envío de postales que ofrecen otras plataformas de entrega de contenidos. Luego de realizar un profundo estudio sobre estos se eligieron las mejores características con fin de incorporarlas a la solución que se plantea en el trabajo. De los sistemas de envío de postales de *Swisscom* y *Pannon* se escogió la característica de personalizar la postal, ya que para el usuario es más fácil acceder a una página *Wap* y mediante un asistente insertar los datos necesarios para la personalización de

su contenido. Además se eligió la característica de que el servicio iba a ser inicializado por un mensaje enviado a un número telefónico previamente asignado. También se optó por la característica de enviar un mensaje de servicio (*Wap Push*) con la URL asociada para personalización de la postal.

Se definieron nuevas características que no fueron encontradas en las soluciones estudiadas y que eran de interés para la nueva propuesta. Estas características se exponen a continuación.

- El sistema dispondrá de una galería de imágenes ordenadas por categorías, para que los usuarios puedan escoger la que más le guste e incorpórala a su postal.
- Las postales serán enviadas a través de un mensaje MMS que contenga la imagen y el texto adicionado por los usuarios.
- Los usuarios pueden enviar la postal a un máximo de tres destinatarios.

La principal funcionalidad del nuevo servicio de postales es permitir a los usuarios la conformación de una tarjeta postal vía Wap de una forma simple, así como también el envío de estas de manera virtual.

Se considera que las características tomadas de otros sistemas así como las añadidas en la nueva propuesta, son las que más se ajustan al entorno en que nuestro sistema se va a desenvolver. De igual forma, consideramos que la nueva propuesta mejora la solución del sistema de envío de postales de *Movistar*, donde el usuario se ve obligado a insertar todos los datos que lleva la postal en un mensaje MMS. Además en todas las soluciones estudiadas se realiza el envío por correo postal, lo cual no es aplicable a nuestras condiciones, teniendo en cuenta de que uno de los objetivos de la nueva propuesta era el envío de postales usando MMS.

Los sistemas estudiados presentan una ventaja con respecto a la nueva propuesta. Esta ventaja consiste en que los usuarios pueden enviar una imagen sin tener que escogerla obligatoriamente

de la galería de imágenes. Esta facilidad no se pudo implementar en la nueva propuesta debido a que en la entidad Procyon no existen las tecnologías apropiadas para desarrollar este tipo de funcionalidad.

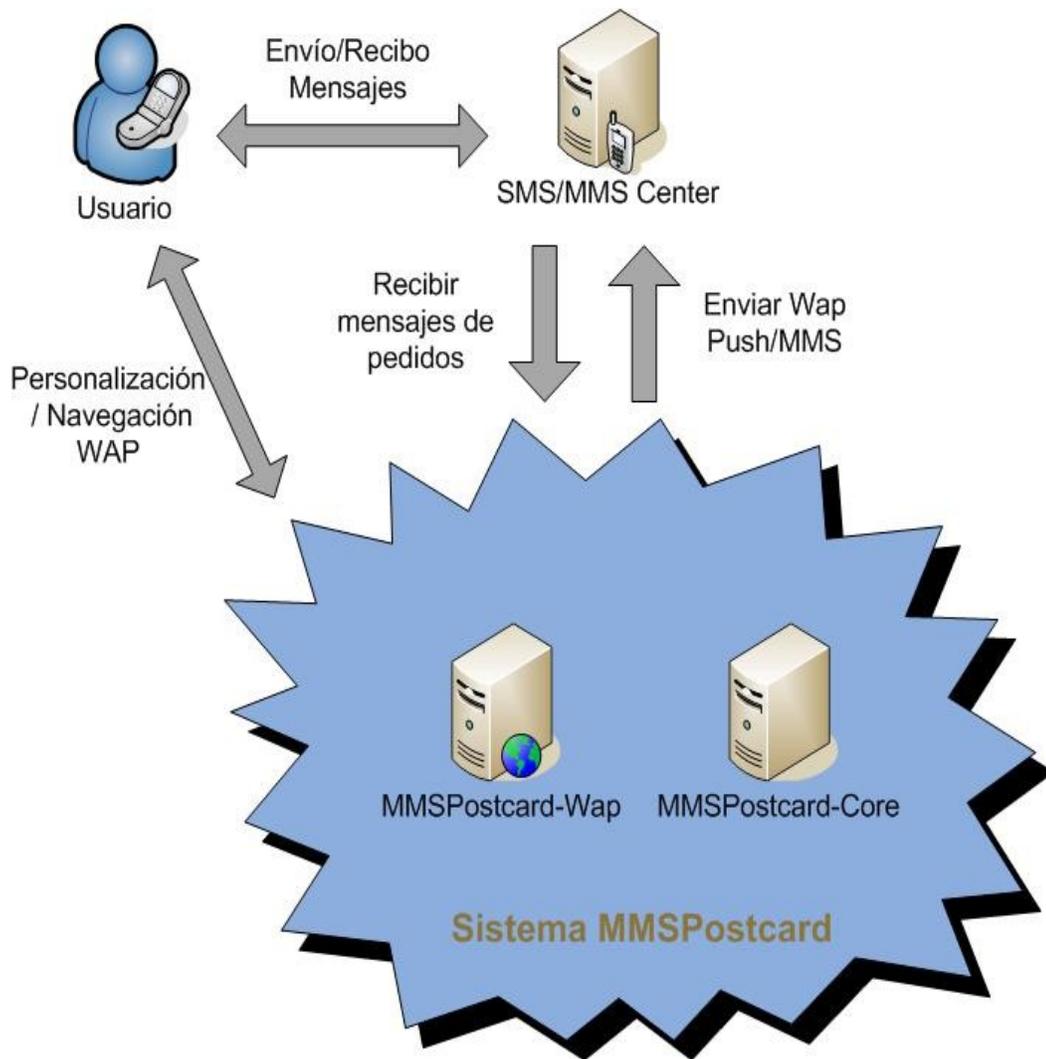


Figura 7. Flujo de procesos del Sistema

2.5 Conclusiones

En este capítulo se explicaron los objetivos estratégicos de la entidad Procyon, se expuso la propuesta de sistema para el servicio de envío de postales, se definió la arquitectura a seguir y el flujo de procesos del mismo. Además se realizó un análisis comparativo con las otras soluciones existentes definiendo de una manera crítica las diferencias y similitudes que existen entre estas.

CAPÍTULO 3: EXPLORACIÓN Y PLANIFICACIÓN

En el presente capítulo se describen las fases de exploración y planificación propias de la metodología de desarrollo utilizada para la implementación del sistema que se propone. Se hace énfasis en los artefactos generados durante el transcurso de las mismas.

3.1 Fase de Exploración

Durante esta fase se realiza el proceso de identificación de las historias de usuario, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

3.1.1 Historias de usuario

Las historias de usuario son la forma en que se especifican en XP los requisitos del sistema. Estas se escriben desde la perspectiva del cliente aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido de estas debe ser concreto y sencillo. El tiempo de desarrollo ideal para una historia de usuario es entre una y tres semanas (Beck, 1999). Las historias de usuario también conducen el proceso de los *test* de aceptación, que son empleados para verificar que las historias han sido implementadas correctamente. Existen diferencias entre las historias de usuario y la tradicional especificación de requisitos. La principal diferencia es el nivel de detalle. Las historias de usuario solamente proporcionan los detalles sobre la estimación de riesgo y cuanto tiempo conllevará la implementación de dicha historia (Fowler, 1999).

Durante esta fase se identificaron seis historias de usuario, las cuales se exponen a continuación.

Tabla 1. Historia de usuario Gestionar Solicitud de Servicio

Historia de Usuario	
Número: 1	Nombre: Gestionar Solicitud de Servicio
Usuario: Cliente	
Prioridad en Negocio: alta	Riesgo en Desarrollo: medio
Puntos Estimados: 2	Iteración Asignada: 1
Descripción El cliente envía un SMS a un número designado con la palabra reservada (ejemplo: <i>postcard</i>), se genera una pre-orden con los datos recibidos y se le envía un <i>Wap Push</i> con la dirección URL a donde debe conectarse para personalizar la postal.	
Observaciones: La pre-orden tiene un tiempo de vida de 24 horas.	

Tabla 2. Historia de usuario Gestionar Orden

Historia de Usuario	
Número: 2	Nombre: Gestionar Orden
Usuario: Reloj	
Prioridad en Negocio: alta	Riesgo en Desarrollo: medio
Puntos Estimados: 2	Iteración Asignada: 2
Descripción: Cada 5 minutos se chequean la ordenes, si están completas se crea un MMS con los datos de postal de la orden y se envía a los destinatarios correspondientes. Si la orden no esta completa se verifica su fecha de expiración.	
Observaciones: Cuando se envía el MMS se pasan los datos importantes de la orden a la tabla de servicios completados, y las ordenes que expiran se pasan sus datos para la tabla de servicios no completados. Estos datos son necesarios a la hora de hacer el cobro del servicio.	

Tabla 3. Historia de usuario Chequear Existencia de pre-orden

Historia de Usuario	
Número: 3	Nombre: Chequear Existencia de pre-orden
Usuario: Cliente	
Prioridad en Negocio: media	Riesgo en Desarrollo: bajo
Puntos Estimados: 1	Iteración Asignada: 2
Descripción: Cuando el cliente se intenta conectar al sitio <i>Wap</i> hay que verificar que el código que da el usuario pertenece a alguna de las pre-órdenes que están guardadas.	
Observaciones:	

Tabla 4. Historia de usuario Gestionar Servicio

Historia de Usuario	
Número: 4	Nombre: Gestionar Servicio
Usuario: Cliente	
Prioridad en Negocio: media	Riesgo en Desarrollo: medio
Puntos Estimados: 2	Iteración Asignada: 2
Descripción: Si el cliente decide seguir con el servicio se re direcciona para que siga con la personalización de la postal, si no se elimina la pre-orden.	
Observaciones: Los datos de la pre-orden eliminada se pasan a la tabla de servicios no completados.	

Tabla 5. Historia de usuario Personalizar Postal

Historia de Usuario	
Número: 5	Nombre: Personalizar Postal
Usuario: Cliente	
Prioridad en Negocio: media	Riesgo en Desarrollo: baja
Puntos Estimados: 2	Iteración Asignada: 2
Descripción: El cliente escoge el tipo de postal que quiere, además de introducir su nombre, primer apellido y destinatarios a los que se quiere que llegue dicha postal.	
Observaciones: Solo se pueden introducir 3 destinatarios.	

Tabla 6. Historia de usuario Crear Orden

Historia de Usuario	
Número: 6	Nombre: Crear Orden
Usuario: Cliente	
Prioridad en Negocio: media	Riesgo en Desarrollo: bajo
Puntos Estimados: 1	Iteración Asignada: 2
Descripción: Con los datos introducidos en la historia de usuario Personalizar Postal se completa la pre-orden y se crea una nueva orden.	
Observaciones: El estado de la pre-orden cambia.	

3.2 Fase de Planificación

Durante la fase de planificación el cliente establece la prioridad de cada historia de usuario, y en correspondencia los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Las estimaciones de esfuerzo asociado a la implementación de las historias las establecen los programadores utilizando como medida la cantidad de puntos. Un punto equivale a una semana ideal de programación.

3.2.1 Estimación de esfuerzo por historias de usuario

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

Tabla 7. Estimación de esfuerzo por historias de usuario

Historias de usuario	Puntos estimados
Gestionar Solicitud de Servicio	2
Gestionar Orden	2
Chequear Existencia de pre-orden	1
Gestionar Servicio	2
Personalizar Postal	2
Crear Orden	1

3.2.2 Plan de iteraciones

Una vez identificadas las Historias de Usuario del sistema y estimado el esfuerzo dedicado a la realización de cada una de estas se procede a la planificación de la etapa de implementación del proyecto. Se decidió realizar tres iteraciones, las cuales se explican a continuación.

3.2.2.1 Primera interacción

El objetivo de esta iteración es la implementación de las historias de usuario de mayor prioridad en el sistema. Durante ésta se creará la mayor parte de la arquitectura del sistema, que será encapsulada en el módulo **mmspostcard-core**. Al final de esta iteración se contará con un prototipo del sistema capaz de gestionar el servicio de postales y enviar las mismas, que será mostrado al cliente con la finalidad de lograr una retroalimentación para el grupo de trabajo (ver figura 8).

3.2.2.2 Segunda iteración

El objetivo de esta iteración es la implementación de las primeras funcionalidades del módulo **mmspostcad-wap**. Durante ésta se gestiona que cuando los usuarios traten de acceder a la página de personalización a través de la URL enviada en el mensaje de servicio, el código asociado corresponda a algunas de la pre-ordenes almacenadas en la base de datos. De no ser así, el usuario no puede acceder la página de personalización. También se tramita el

seguimiento del servicio por parte del usuario. Una vez que se accede a la interfaz de personalización el usuario puede completar el envío de la postal o puede cancelar el mismo. De ser cancelado la pre-orden asociada será eliminada (ver figura 9).

3.2.2.3 Tercera iteración

El objetivo de esta iteración es la implementación de las últimas funcionalidades del módulo **mmspostcard-wap**. Durante la primera parte de la iteración se gestionará todo lo referente a inserción de los datos de la postal (imagen escogida, nombre y apellidos del usuario, destinatarios, mensaje personal). En la segunda parte de la iteración se implementará el completamiento del servicio. Al final de esta iteración se constará con el módulo encargado de realizar la personalización de la postal (ver figura 10).

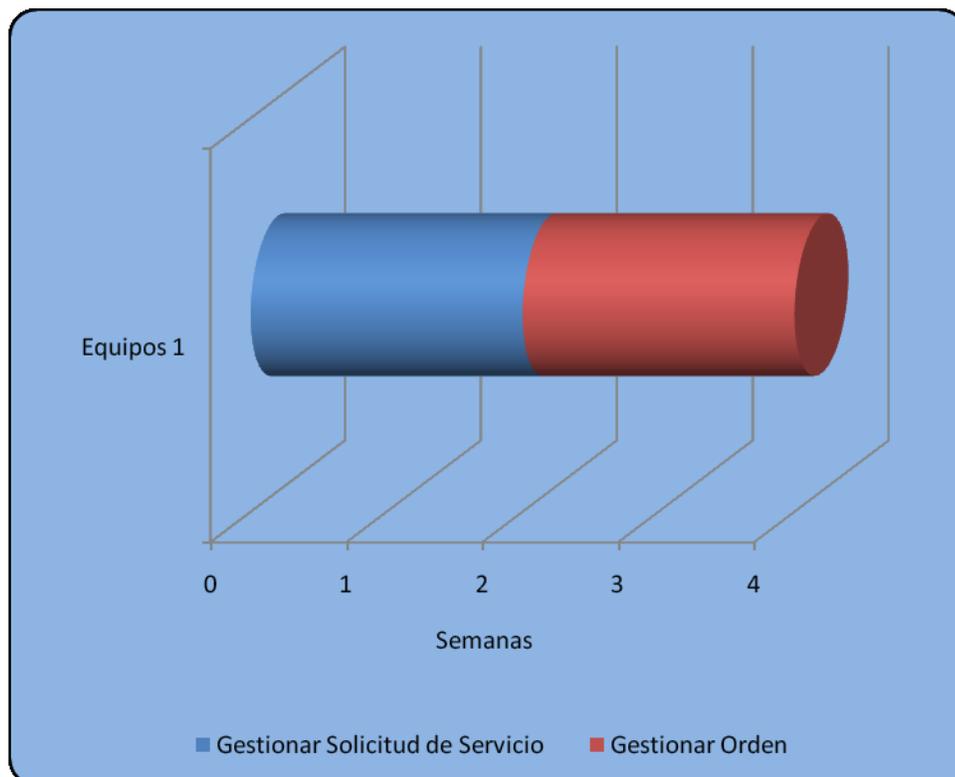


Figura 8. Plan de iteración 1

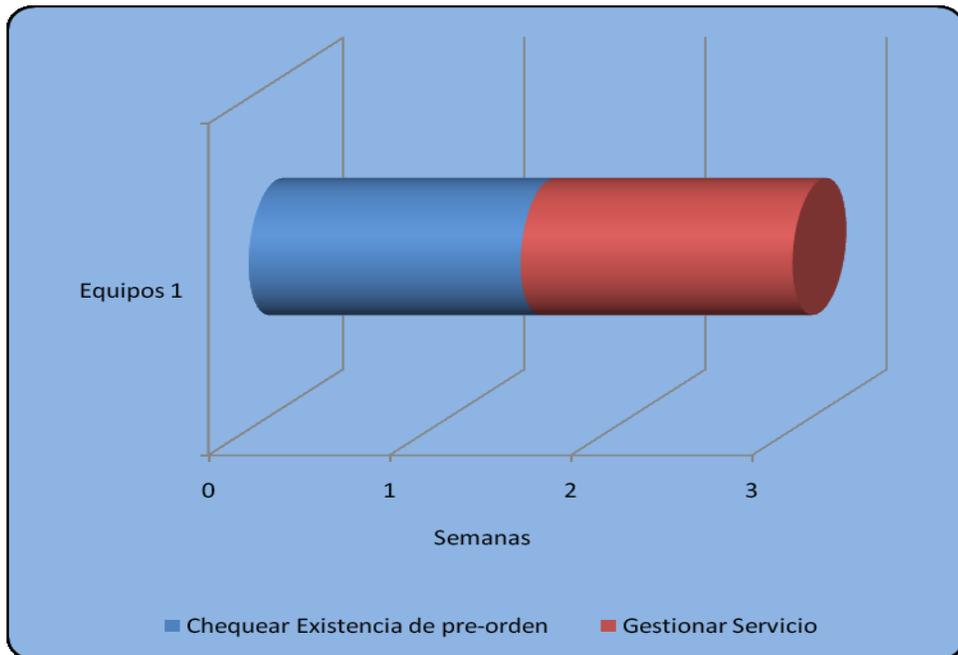


Figura 9. Plan de iteración 2

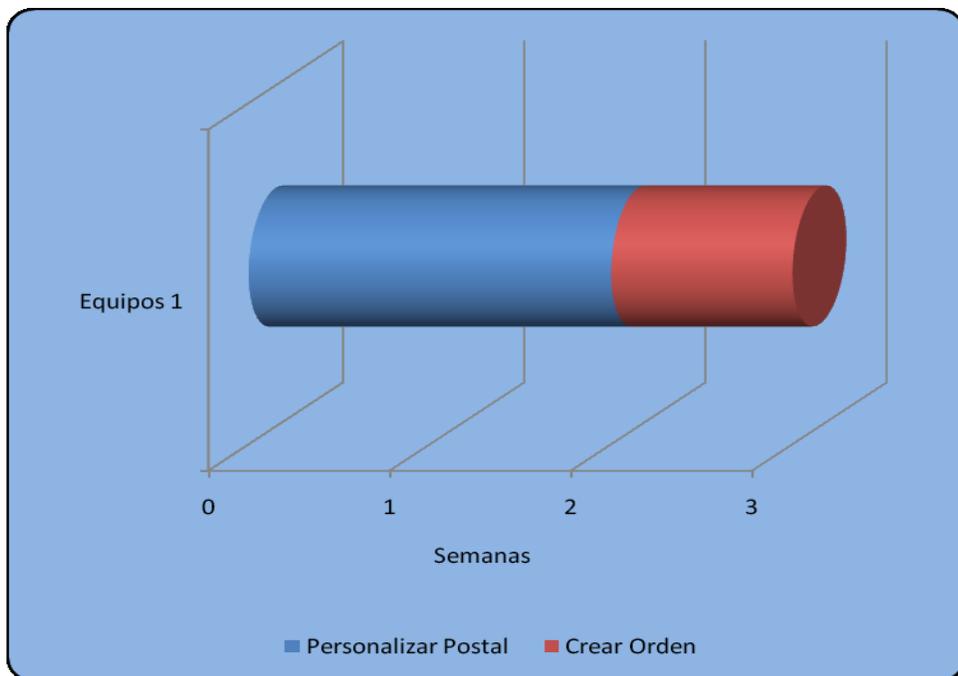


Figura 10. Plan de iteración 3

3.2.3 Plan de Entrega

A continuación se presenta el plan de entregas para la fase de implementación. Como resultado del mismo se pueden ver las fechas reales en que cada versión será entregada. El motivo por el que se crean nuevas versiones de algunos de los módulos puede ser por cambios desde la última versión o simplemente actualización de las dependencias en otros módulos cuando una nueva versión de éstos últimos es entregada.

Tabla 8. Plan de Entrega por fechas

Módulo	2 ^{da} semana de abril	1 ^{ra} semana de mayo	1 ^{ra} semana de junio
mmspostcard-core	0.1	0.2	0.3
mmspostcard-wap		0.1	0.2

3.3 Conclusiones

En el capítulo se abordaron las fases de exploración y planificación de la metodología utilizada para el desarrollo del sistema, se generaron los artefactos de las mismas, se explicaron las historias de usuario y se definió el trabajo a realizar en cada una de las iteraciones del proyecto, así como el plan de entrega de cada uno de los módulos.

CAPÍTULO 4: DISEÑO, IMPLEMENTACIÓN Y PRUEBA

En el presente capítulo se describen las fases de diseño, implementación y prueba propias de la metodología de desarrollo XP. Se detallan las tres iteraciones realizadas durante la etapa de codificación del proyecto, así como las tareas de programación desarrolladas en cada una de estas. También se hace referencia a las pruebas de aceptación realizadas sobre el sistema.

4.1 Fase de Diseño

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra deben ser evitados. En todo momento, el diseño adecuado para el software es aquel que supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos (Fowler, 2004).

Para el diseño de las aplicaciones, la metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando notación UML. En su lugar se usan otras técnicas como las tarjetas CRC como una extensión informal a UML. La técnica de las tarjetas CRC se puede usar para guiar el sistema a través de análisis guiados por la responsabilidad. Las clases se examinan, se filtran y se refinan en base a sus responsabilidades con respecto al sistema, y las clases con las que necesitan colaborar para completar sus responsabilidades (Wake, 1999). No obstante el uso de los diagramas de clases UML puede aplicarse siempre y cuando mejore la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante (Fowler, 2004).

Los diagramas utilizados durante el desarrollo del sistema de envío de postales se encuentran en el Anexo 1.

4.2 Fase de Implementación

Durante el inicio de cada iteración, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario (Fowler, 1999). Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Estas tareas, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente. (Beck, 1999)

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

4.2.1 Primera iteración

Durante el transcurso de esta iteración se implementaron las historias de usuario de mayor prioridad en el sistema, construyéndose la base de la arquitectura. Se logró crear un prototipo del sistema capaz de gestionar el servicio de postales y enviar las mismas.

Tabla 9. Historias de usuario desarrolladas en la primera iteración

Historia de Usuario	Estimación	Real
Gestionar Solicitud de Servicio	2	2
Gestionar Orden	2	2
Total	4	4

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

Historia de usuario **Gestionar Solicitud de Servicio**

Tabla 10. Tarea 1 de la historia de usuario Gestionar Solicitud de Servicio

Tarea	
Número tarea: 1	Número historia: 1
Nombre tarea: Recibir mensaje de solicitud de servicio	
Tipo de tarea : Desarrollo	Puntos estimados: 0,4
Fecha inicio: 19/03/2007	Fecha fin: 20/03/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se creará una cola de entrada, por donde entrarán al sistema los mensajes de solicitud de servicio de los usuarios. Esta cola recibirá los mensajes a través de SAMS-M, el cual se comunica con el centro de mensajería.	

Tabla 11. Tarea 2 de la historia de usuario Gestionar Solicitud de Servicio

Tarea	
Número tarea: 2	Número historia: 1
Nombre tarea: Procesar solicitud de servicio	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 21/03/2007	Fecha fin: 27/03/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: A partir del mensaje de solicitud, se crea una orden de servicio que es almacenada en la base de datos. Como el usuario no ha personalizado la postal todavía, a la orden se le llama pre-orden. Esta pre-orden esta compuesta por la fecha de creación, fecha de expiración, numero telefónico del usuario, estado (“falso” si la orden no esta completa y “verdadero” en caso contrario) y código de la orden (guid). El código de la orden es una cadena de 20 caracteres que se forma a partir de la hora del sistema para que no exista ninguna duplicación de código.	

Tabla 12. Tarea 3 de la historia de usuario Gestionar Solicitud de Servicio

Tarea	
Número tarea: 3	Número historia: 1
Nombre tarea: Enviar mensaje de servicio (Wap Push)	
Tipo de tarea : Desarrollo	Puntos estimados: 0,6
Fecha inicio: 28/03/2007	Fecha fin: 30/03/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: A partir de que se crea la orden, se compone un mensaje de servicio que lleva consigo la url a donde debe conectarse el usuario para personalizar la postal. A esta URL se le agrega como parámetro el código de la orden. Este mensaje será enviado a través de la cola de salida del sistema, por donde saldrán todos los mensajes que genere la aplicación.	

Historia de usuario Gestionar Orden

Tabla 13. Tarea 1 de la historia de usuario Gestionar Orden

Tarea	
Número tarea: 1	Número historia: 2
Nombre tarea: Eliminar ordenes expiradas	
Tipo de tarea : Desarrollo	Puntos estimados: 0,4
Fecha inicio: 02/04/2007	Fecha fin: 03/04/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: A partir de que una orden se crea tiene un tiempo de vida de 24 horas. Si el usuario en este tiempo no personaliza su postal la orden expira. En este caso la orden es eliminada y se almacena en la base de datos el número telefónico del usuario y la fecha en que se elimina.	

Tabla 14. Tarea 2 de la historia de usuario Gestionar Orden

Tarea	
Número tarea: 2	Número historia: 2
Nombre tarea: Enviar mensaje multimedia con la postal	
Tipo de tarea : Desarrollo	Puntos estimados: 0,8
Fecha inicio: 04/04/2007	Fecha fin: 09/04/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se crea un mensaje multimedia a partir de los datos introducidos por el usuario cuando personaliza la postal, al MMS se le adjuntan la imagen, el mensaje personal y el fichero SMIL que es el encargado de la presentación del mensaje. El mensaje es enviado a través de la cola de salida del sistema.	

Tabla 15. Tarea 3 de la historia de usuario Gestionar Orden

Tarea	
Número tarea: 3	Número historia: 2
Nombre tarea: Planificador encargado de gestionar las ordenes	
Tipo de tarea : Desarrollo	Puntos estimados: 0,8
Fecha inicio: 10/04/2007	Fecha fin: 13/04/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se implementará un servicio que se active cada cinco minutos, que será el encargado de ejecutar la eliminación de las ordenes que han expirado y de enviar los mensajes multimedia a los destinatarios correspondientes.	

4.2.2 Segunda iteración

Durante el transcurso de la iteración se implementaron las primeras funcionalidades del módulo **mmspostcard-wap**. Se logró crear una primera vista de la interfaz *Wap* donde el usuario puede cancelar el servicio o continuar con la personalización de la postal.

Tabla 16. Historias de usuario desarrolladas en la segunda iteración

Historia de Usuario	Estimación	Real
Chequear Existencia de pre-orden	1	1
Gestionar Servicio	2	2
Total	3	3

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

Historia de usuario **Chequear Existencia de pre-orden**

Tabla 17. Tarea 1 de la historia de usuario Chequear Existencia de pre-orden

Tarea	
Número tarea: 1	Número historia: 3
Nombre tarea: Crear servlet de escucha	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 16/04/2007	Fecha fin: 20/04/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se implementara un <i>servlet</i> que será el encargado de estar a la escucha de las peticiones que se realicen por los usuarios. Este <i>servlet</i> será el encargado de verificar que la orden existe en la base de datos y re-direccionar a los usuarios a la interfaz <i>Wap</i> para que puedan personalizar la postal. La verificación de la orden consiste en asociar el parámetro de identificación que viene en la petición del usuario con alguno de los códigos de las órdenes que están almacenadas.	

Historia de usuario **Gestionar Servicio**

Tabla 18. Tarea 1 de la historia de usuario Gestionar Servicio

Tarea	
Número tarea: 1	Número historia: 4
Nombre tarea: Crear interfaz Wap para gestionar el seguimiento del servicio	
Tipo de tarea : Desarrollo	Puntos estimados: 0,6
Fecha inicio: 23/04/2007	Fecha fin: 25/04/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se creará una interfaz <i>Wap</i> que le permita al usuario seleccionar si quiere seguir con la personalización de la postal o cancelar el servicio solicitado.	

Tabla 19. Tarea 2 de la historia de usuario Gestionar Servicio

Tarea	
Número tarea: 2	Número historia: 4
Nombre tarea: Cancelar servicio	
Tipo de tarea : Desarrollo	Puntos estimados: 0,8
Fecha inicio: 26/04/2007	Fecha fin: 01/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Si el usuario selecciona la opción de cancelar servicio, la orden asociada se eliminará, será almacenado en la base de datos el número telefónico del usuario y la fecha en que se cancela el servicio.	

Tabla 20. Tarea 3 de la historia de usuario Gestionar Servicio

Tarea	
Número tarea: 3	Número historia: 4
Nombre tarea: Continuar con el servicio	
Tipo de tarea : Desarrollo	Puntos estimados: 0,6
Fecha inicio: 02/05/2007	Fecha fin: 04/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Si el usuario selecciona la opción de darle seguimiento al servicio, será re direccionado a la interfaz que muestra las categorías de imágenes existentes para la confección de la postal.	

4.2.3 Tercera iteración

Durante el transcurso de la tercera iteración se implementaran las últimas funcionalidades del módulo **mmspostcard-wap**. Estas funcionalidades le proporcionarán al usuario la forma de confeccionar su postal para que sean enviadas a los destinatarios especificados por este. Al finalizar esta fase se obtendrá una primera versión del producto. Antes de salir esta versión al mercado primero debe pasar por un periodo de pruebas.

Tabla 21. Historias de usuario desarrolladas en la tercera iteración

Historia de Usuario	Estimación	Real
Personalizar Postal	2	2
Crear Orden	1	1
Total	3	3

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

Historia de usuario **Personalizar Postal**

Tabla 22. Tarea 1 de la historia de usuario Personalizar Postal

Tarea	
Número tarea: 1	Número historia: 5
Nombre tarea: Crear interfaz Wap para mostrar las categorías en que están organizadas las imágenes.	
Tipo de tarea : Desarrollo	Puntos estimados: 0,2
Fecha inicio: 07/05/2007	Fecha fin: 07/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se creará una interfaz <i>Wap</i> donde el usuario pueda visualizar todas las categorías de postales existentes y escoger la que más le interese.	

Tabla 23. Tarea 2 de la historia de usuario Personalizar Postal

Tarea	
Número tarea: 2	Número historia: 5
Nombre tarea: Buscar categorías	
Tipo de tarea : Desarrollo	Puntos estimados: 0,2
Fecha inicio: 08/05/2007	Fecha fin: 08/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se buscarán las categorías existentes en la base de datos, y serán mostradas en la interfaz <i>Wap</i> . Si alguna categoría no tiene ninguna imagen asociada, entonces no se mostrará.	

Tabla 24. Tarea 3 de la historia de usuario Personalizar Postal

Tarea	
Número tarea: 3	Número historia: 5
Nombre tarea: Crear interfaz Wap para la mostrar las imágenes	
Tipo de tarea : Desarrollo	Puntos estimados: 0,2
Fecha inicio: 09/05/2007	Fecha fin: 09/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se creará una Interfaz Wap donde los usuarios puedan visualizar las imágenes de la categoría seleccionada. Las imágenes tendrán un tamaño determinado de acuerdo al dispositivo consultante. Esta interfaz permitirá al usuario seleccionar una imagen para la postal o ver una resolución mayor de esta en otra interfaz.	

Tabla 25. Tarea 4 de la historia de usuario Personalizar Postal

Tarea	
Número tarea: 4	Número historia: 5
Nombre tarea: Buscar imágenes	
Tipo de tarea : Desarrollo	Puntos estimados: 0,2
Fecha inicio: 10/05/2007	Fecha fin: 10/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se buscarán todas las imágenes de la categoría seleccionada por el usuario.	

Tabla 26. Tarea 5 de la historia de usuario Personalizar Postal

Tarea	
Número tarea: 5	Número historia: 5
Nombre tarea: Crear paginado para mostrar las postales	
Tipo de tarea : Desarrollo	Puntos estimados: 0,6
Fecha inicio: 11/05/2007	Fecha fin: 15/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Como una categoría puede tener muchas imágenes, se creará un paginado en la interfaz donde se muestran éstas (máximo cinco imágenes por página).	

Tabla 27. Tarea 6 de la historia de usuario Personalizar Postal

Tarea	
Número tarea: 6	Número historia: 5
Nombre tarea: Crear interfaz Wap para visualizar la imagen en una mayor resolución	
Tipo de tarea : Desarrollo	Puntos estimados: 0,2
Fecha inicio: 16/05/2007	Fecha fin: 16/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Si el usuario escoge la opción de visualizar la imagen en mayor resolución en la interfaz donde se muestran las imágenes, se mostrará una nueva interfaz con la imagen seleccionada con una nueva resolución. Las dimensiones de la imagen dependen de las propiedades del dispositivo consultante. La interfaz tendrá la opción de poder seleccionar la imagen y añadirla a la postal.	

Tabla 28. Tarea 7 de la historia de usuario Personalizar Postal

Tarea	
Número tarea: 7	Número historia: 5
Nombre tarea: Crear interfaz Wap para insertar los datos de la postal	
Tipo de tarea : Desarrollo	Puntos estimados: 0,4
Fecha inicio: 17/05/2007	Fecha fin: 18/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se creará una interfaz Wap con un formulario donde el usuario tiene que insertar el nombre y primer apellido, los destinatarios a los que irá dirigida la postal (máximo tres destinatarios) y el mensaje personal.	

Historia de usuario **Crear Orden**

Tabla 29. Tarea 1 de la historia de usuario Crear Orden

Tarea	
Número tarea: 1	Número historia: 6
Nombre tarea: Crear interfaz Wap para visualizar una vista previa de la postal	
Tipo de tarea : Desarrollo	Puntos estimados: 0,4
Fecha inicio: 21/05/2007	Fecha fin: 22/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Se creará una interfaz Wap donde se muestre una vista previa de la postal creada, además los destinatarios a los que será enviada. La interfaz tendrá la opciones de cancelar la postal (comenzar de nuevo el proceso de personalización), modificar datos (modificar los datos insertados) y guardar la postal (terminar la postal).	

Tabla 30. Tarea 2 de la historia de usuario Crear Orden

Tarea	
Número tarea: 2	Número historia: 6
Nombre tarea: Cancelar postal	
Tipo de tarea : Desarrollo	Puntos estimados: 0,4
Fecha inicio: 23/05/2007	Fecha fin: 24/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Si el usuario decide cancelar la postal, será re-direccionado a la interfaz donde se muestran las categorías, dándole la posibilidad de comenzar el proceso de personalización de la postal desde el principio.	

Tabla 31. Tarea 3 de la historia de usuario Crear Orden

Tarea	
Número tarea: 3	Número historia: 6
Nombre tarea: Modificar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0,4
Fecha inicio: 25/05/2007	Fecha fin: 28/05/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Si el usuario no está de acuerdo con los datos mostrados cuando se visualiza la vista previa de la postal puede modificar los datos. El usuario será re-direccionado a la interfaz donde se insertan los datos de la postal y puede realizar los cambios que considere.	

Tabla 32. Tarea 4 de la historia de usuario Crear Orden

Tarea	
Número tarea: 4	Número historia: 6
Nombre tarea: Salvar postal	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 29/05/2007	Fecha fin: 01/06/2007
Programador responsable: Lex Karel Zayas Hernández – Yudier Villalta Villar	
Descripción: Si el usuario considera que esta de acuerdo con el contenido de la postal, ésta será almacenada en la base de datos y quedará completada la orden asociada a dicha postal (el estado de la orden cambia a “verdadero”).	

4.3 Fase de Prueba

Uno de los pilares de XP es el proceso de pruebas (Beck, 1999). XP anima a probar tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final (Crispin y House, 2002).

4.3.1 Pruebas de Aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario (Crispin y House, 2002). Durante las iteraciones las historias de usuario seleccionadas

serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. Las pruebas de aceptación significan la satisfacción del cliente con el producto desarrollado, el final de una iteración y el comienzo de la siguiente (Miller, 2001). El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable (Wells, 2006).

Las pruebas de aceptación realizadas al sistema se encuentran en el Anexo 2.

4.4 Conclusiones

En este capítulo se describieron las fases de diseño, implementación y prueba. Se realizó una breve descripción del diseño del sistema, el cual se realizó basado en la simplicidad. Se detallaron cada una de las tres iteraciones del proyecto haciendo énfasis en los artefactos generados en la fase de implementación y se abordó acerca de la importancia de las pruebas en la metodología XP, en especial las pruebas de aceptación, ya que estas miden la satisfacción del cliente con el producto desarrollado.

CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

En la planificación del proceso de desarrollo de software es de vital importancia la estimación, la cual consiste en determinar con cierto grado de certeza los recursos necesarios para el desarrollo del mismo, ya sean recursos de *hardware*, *software*, esfuerzo, tiempo y costo. En el presente capítulo se realizará un estudio de factibilidad para la realización del sistema propuesto mediante una estimación de tamaño, esfuerzo y planificación necesaria para llevar a cabo el mismo.

5.1 Características del proyecto

Tabla 33. Entradas externas

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
Solicitar servicio	1	1	Simple
Personalizar postal	1	6	Simple
Total		7	

Tabla 34. Salidas externas

Nombre de la salida externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
Categorías	1	5	Simple
Postales por categoría	1	5	Simple
Vista Agrandada de postal	1	1	Simple
Vista previa de postal	1	3	Simple
Total		14	

Tabla 35. Ficheros internos

Nombre del fichero interno	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
Colas de mensajes	1	1	Simple
Temporalizadores de órdenes	1	1	Simple
Interacción con Gateway	1	1	Simple
Total		3	

Tabla 36. Peticiones

Nombre de la petición	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
Total		0	

Tabla 37. Interfaces externas

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
MMS	1	1	Simple
Total		1	

Tabla 38. Puntos de función desajustados

Elementos	Simple		Medio		Complejo		Subtotal
	No.	Peso	No.	Peso	No.	Peso	
Entradas externas	7	3	0	4	0	6	21
Salidas externas	14	4	0	5	0	7	56
Ficheros internos	3	7	0	10	0	15	21
Peticiones	0	3	0	4	0	6	0
Interfaces externas	1	5	0	7	0	10	5
Total							103

5.1.2 Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombre y costo

El modelo COCOMO II usa Puntos Función (FP) y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura. Los Puntos de Función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema. Son útiles estimadores ya que están basados en información que está disponible en las etapas tempranas del ciclo de vida del desarrollo de software.

Tabla 39. Características

Características	Valor
Puntos de función desajustados	103
Lenguaje (Java)	53
Instrucciones fuentes por puntos de función	5459
Instrucciones fuentes	5.45 ksloc

Tabla 40. Factores de escala

Nombre	Valor	Justificación
Precedentes	1.24	Existen varios sistemas similares a nivel mundial, pero no a nivel nacional.
Flexibilidad de desarrollo	1.01	Cuenta con una flexibilidad de desarrollo muy alta.
Cohesión del equipo	1.10	El equipo de desarrollo presenta una muy alta cohesión.
Solución de riesgos	4.24	No se identificaron grandes riesgos.
Madurez del proceso	1.56	Se tiene una experiencia previa en el desarrollo de aplicaciones de este tipo.
Total(SF)	9.15	

Tabla 41. Multiplicadores de esfuerzo

Nombre	Valor	Justificación
RCPX	1.05	El sistema presenta un nivel alto de complejidad
RUSE	1.00	El nivel de reutilización del producto es alto.
PDIF	0.96	Uso de memoria y almacenamiento normal, plataforma estable.
PREX	0.90	Alto grado de experiencia en el área de la aplicación, plataforma, lenguaje y herramienta.
PERS	0.52	Alta capacidad del personal.
FCIL	0.82	Se utilizaron entornos de desarrollo integrados, herramientas de modelación automatización que facilitan el trabajo.
SCED	1.00	Se empleó el tiempo planificado para el desarrollo del sistema.
Total(EM)	0.89	

Cálculos

$$A = 1.94 \quad B = 1.91 \quad C = 1.67 \quad D = 1.24$$

$$E = B + 0.01 * SF = 1.91 + 0.01 * 9.15 = 1.9115$$

$$PM = A * Size^E * \prod EM = 2.94 * 5.45^{1.0015} * 0.89 = 13.33$$

$$F = D + 0.2 * 0.01 * \sum SF = 1.24 + 0.2 * 0.01 * 9.15 = 1.219$$

$$TDEV = C * PM^F = 1.67 * 13.33^{1.219} = 7.20$$

$$CH = \frac{PM}{TDEV} = \frac{13.33}{7.20} \approx 1.85$$

$$C = CH * Sal * PM = 1.85 * 100 * 13.33 = 2466$$

Tabla 41. Resultados

Cálculo de:	Valor
Esfuerzo	13.33 hombres/mes
Tiempo de desarrollo	6.6 meses
Cantidad de hombres	2 hombres
Salario medio	\$100
Costo	\$2666

5.2 Beneficios tangibles e intangibles

El sistema de envío de postales MMS es un producto con fines comerciales, puede ser añadido a la plataforma *MobiStore* como un servicio adicional, aumentando su costo en el mercado. El beneficio fundamental es contar con un sistema que a través del API SAMS-M sea capaz de la personalización y el envío de contenidos mediante MMS.

Como beneficio intangible aportado por el sistema desarrollado se puede mencionar que servirá de base para otros servicios de valor agregado que serán desarrollados por la entidad Procyon.

5.3 Análisis de costo

El desarrollo de un producto siempre presenta un costo de producción, el cual tiene que ser justificado en base a los beneficios reportados por el mismo. Y un producto informático no esta exento de esto. El sistema presentado en este trabajo no genera grandes gastos debido en gran medida a la utilización de plataformas, tecnologías y herramientas libres que no requieren del pago de licencia por su uso o comercialización. Su costo se ve influido por el salario de los desarrolladores.

5.5 Conclusiones

En este capítulo se realizó un análisis de factibilidad de la solución propuesta, obteniendo los estimados de tamaño, esfuerzo y planificación. Se arribó a la conclusión de que es viable el desarrollo del sistema. Esta afirmación se basa en la comparación de los costos de producción con los beneficios que reportará el sistema al incorporarse a la plataforma *MobiStore*.

CONCLUSIONES

Con el desarrollo de este trabajo se profundizó en el conocimiento de la mensajería móvil y los servicios de valor agregado. Se aplicó la metodología XP para guiar el proceso de desarrollo de software, mediante el cual se obtuvo la implementación de un sistema de envío de postales MMS para usuarios de teléfonos móviles con las funcionalidades previstas para el primer ciclo de desarrollo del software. El diseño y la implementación se rigieron por los estándares de la plataforma Java EE, aplicando diferentes patrones de diseño. Se definieron los parámetros de configuración e instalación con la documentación correspondiente. Se puede concluir que se ha cumplido satisfactoriamente el objetivo trazado por este trabajo enfatizando en los siguientes puntos:

- Se implementó un sistema de envío de postales virtuales por MMS, como servicio de valor agregado a la plataforma de manejo de contenidos *MobiStore*, de la entidad Procyon.
- Se implementó una aplicación servidora que maneja todo lo referente a la solicitud del servicio de envío de postales virtuales, así como la entrega de los mensajes MMS a los destinatarios de las postales.
- Se desarrolló una interfaz *Wap* amigable para la personalización de las postales por parte del usuario.

RECOMENDACIONES

A continuación se listan una serie de recomendaciones con el objetivo de realizar un seguimiento a este trabajo:

- Integrar el sistema a la plataforma *MobiStore*.
- Implementar una interfaz de administración con el objetivo de poder controlar el servicio de manera independiente.
- Utilizar este sistema como base para la implementación de otros servicios de valor agregado en la plataforma *MobiStore*.
- Realizar pruebas de este sistema en condiciones reales.

BIBLIOGRAFÍA

Bayern, Shawn. 2001. *Web Development with JavaServer Pages*. s.l. : Manning Publications Company, 2001. 193011012X.

Beck, Kent. 1999. *Extreme Programming Explained*. s.l. : Addison-Wesley, 1999. 0201616416.

Crispin, Lisa y House, Tip. 2002. *Testing Extreme Programming*. s.l. : Addison-Wesley, 2002. 0321113551.

Eclipse Magazine. 2007. JBoss Eclipse IDE 2.0.0. *Eclipse Magazine*. [En línea] 2007. [Citado el: 15 de Abril de 2007.]

<http://www.eclipsemag.net/ecm/ecmnews/psecom,id,192,nodeid,2.html?PHPSESSID=4>.

Ericsson. 2006. WAP push. *Ericsson*. [En línea] 2 de Mayo de 2006. [Citado el: 2 de Febrero de 2007.]

http://www.ericsson.com/mobilityworld/sub/open/technologies/open_development_tips/tools/wap_push_sdk.

Fowler, Martin. 2004. Is Design Dead. *martinfowler.com*. [En línea] Mayo de 2004. [Citado el: 14 de Mayo de 2007.] <http://martinfowler.com/articles/designDead.html>.

—. 1999. *Planning Extreme Programming*. s.l. : Addison-Wesley, 1999. 0201710919.

Guemes, Celetino. 2006. Wurlf-Wall. *FRAMEWORK AME*. [En línea] 6 de Abril de 2006. [Citado el: 16 de Febrero de 2007.] <http://ame.endesa.es/confluence/display/AMEBASE/wurlf-wall>.

Hall, Marty y Brown, Larry. 2003. *Core Servlets and Java Server Pages*. s.l. : Prentice Hall, 2003. 0130092290.

IBM. 2007. Eclipse Platform Technical Overview. *Eclipse*. [En línea] 2007. [Citado el: 12 de Abril de 2007.] <http://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.pdf>.

International Engineering Consortium. 2007. Global System for Mobile Communication (GSM). *IEC*. [En línea] 2007. [Citado el: 26 de Enero de 2007.] <http://www.iec.org/online/tutorials/gsm/index.html>.

—. 2007. Wireless Internet Network Communications Architecture. *IEC*. [En línea] 2007. [Citado el: 26 de Enero de 2007.] <http://www.iec.org/online/tutorials/winternet/topic04.html>.

JBoss. 2007. JBoss Application Server. *JBoss a division of Red Hat*. [En línea] 2007. [Citado el: 1 de Marzo de 2007.] http://docs.jboss.com/jbossas/whatsnew40/html_single/.

Jendrock, Eric, y otros. 2006. *The Java™ EE 5*. Los Angeles : Addison-Wesley, 2006. 0–321–49029–0.

Kofler, Michael. 2005. *The Definitive Guide to MySQL 5, Third Edition (Definitive Guide)*. s.l. : Apress, 2005. 1590595351.

Miller, Roy. 2001. *Extreme Programming Applied*. s.l. : Addison-Wesley, 2001. 0201616408.

Monson-Haefel, Richard. 2006. *Enterprise JavaBeans 3.0 (5th Edition)*. s.l. : O'Reilly, 2006. 059600978X.

MySQL. 2006. The Main Features of MySQL. *MySQL developer zone*. [En línea] 2006. [Citado el: 7 de Marzo de 2007.] <http://mysql.org/doc/refman/5.0/en/features.html>.

Ort, Ed y Mehta, Bhakti. 2003. Java Architecture for XML Binding (JAXB). *Sun Developer Network*. [En línea] Marzo de 2003. [Citado el: 12 de Febrero de 2007.] <http://java.sun.com/developer/technicalArticles/WebServices/jaxb/>.

- Palos, Jose Antonio. 2004.** El API JAXB. *Java en castellano*. [En línea] 2 de Noviembre de 2004. [Citado el: 13 de Febrero de 2007.] <http://programacion.net/java/tutorial/jaxb/>.
- Pannon. 2007.** MMS Postcard. *Pannon*. [En línea] 2007. [Citado el: 24 de Enero de 2007.] http://www.pannon.hu/en/private/services/mms/mms_postcard/.
- Passani, Luca. 2002.** WALL - The Wireless Abstraction Library. *Wireless Universal Resource File*. [En línea] 2002. [Citado el: 15 de Febrero de 2007.] <http://wurfl.sourceforge.net/index.php>.
- Patel, Rima, Brose, Gerald y Silverman, Micah. 2006.** *Mastering Enterprise JavaBeans 3.0*. Indianapolis : Wiley, 2006. 0-321-49029-0.
- Perry, Bruce. 1999.** *Java Servlet & JSP Cookbook*. s.l. : O'Reilly, 1999. 0596005725.
- Routt, William. 2003.** *Wireless Markup Language (WML) Scripting and Programming using WML, cHTML, and xHTML*. s.l. : Althos, 2003. 0974278750.
- Sun Microsystems. 2004.** JSR 212: Server API for Mobile Services: Messaging - SAMS: Messaging. *Java Community Process*. [En línea] 15 de Julio de 2004. [Citado el: 26 de Febrero de 2007.] <http://jcp.org/en/jsr/detail?id=212>.
- Swisscom Mobile. 2007.** MMS Postcard. *Swisscom Mobile*. [En línea] 2007. [Citado el: 23 de Enero de 2007.] http://www.swisscom-mobile.ch/scm/prv_mms_postkarte-en.aspx.
- TechTarget. 2005.** Multimedia Message Service. *Search Tech Target*. [En línea] 2 de Mayo de 2005. [Citado el: 2 de Febrero de 2007.] http://searchmobilecomputing.techtarget.com/sDefinition/0,290660,sid40_gci943702,00.html.
- . 2005.** Short Message. *Search Tech Target*. [En línea] 2 de Mayo de 2005. [Citado el: 2 de Febrero de 2007.] http://searchmobilecomputing.techtarget.com/sDefinition/0,290660,sid40_gci773769,00.html.

—. 2005. SMIL. *Search Tech Target*. [En línea] 2 de Mayo de 2005. [Citado el: 4 de Febrero de 2007.] http://searchwebservices.techtarget.com/sDefinition/0,290660,sid26_gci214217,00.html.

Telefónica I+D. 2005. *Las Telecomunicaciones y la Movilidad en la Sociedad de la Información*. Madrid : AHCINET, 2005. 84-89900-37-X.

Telefónica Movistar. 2007. Postales. *Movistar*. [En línea] 2007. [Citado el: 23 de Enero de 2007.] <http://www.movistar.es/particulares/servicios/multimedia/postales>.

Tigris.org. 2007. Subclipse. *Tigris.org Open Source Software Engineering Tools*. [En línea] 2007. [Citado el: 15 de Abril de 2007.] <http://subclipse.tigris.org/>.

—. 2007. Subversion. *Tigris.org Open Source Software Engineering Tools*. [En línea] 2007. [Citado el: 16 de Abril de 2007.] <http://subversion.tigris.org/>.

Wake, William C. 1999. *Extreme Programming Explored*. s.l. : Addison-Wesley, 1999. 0201733978.

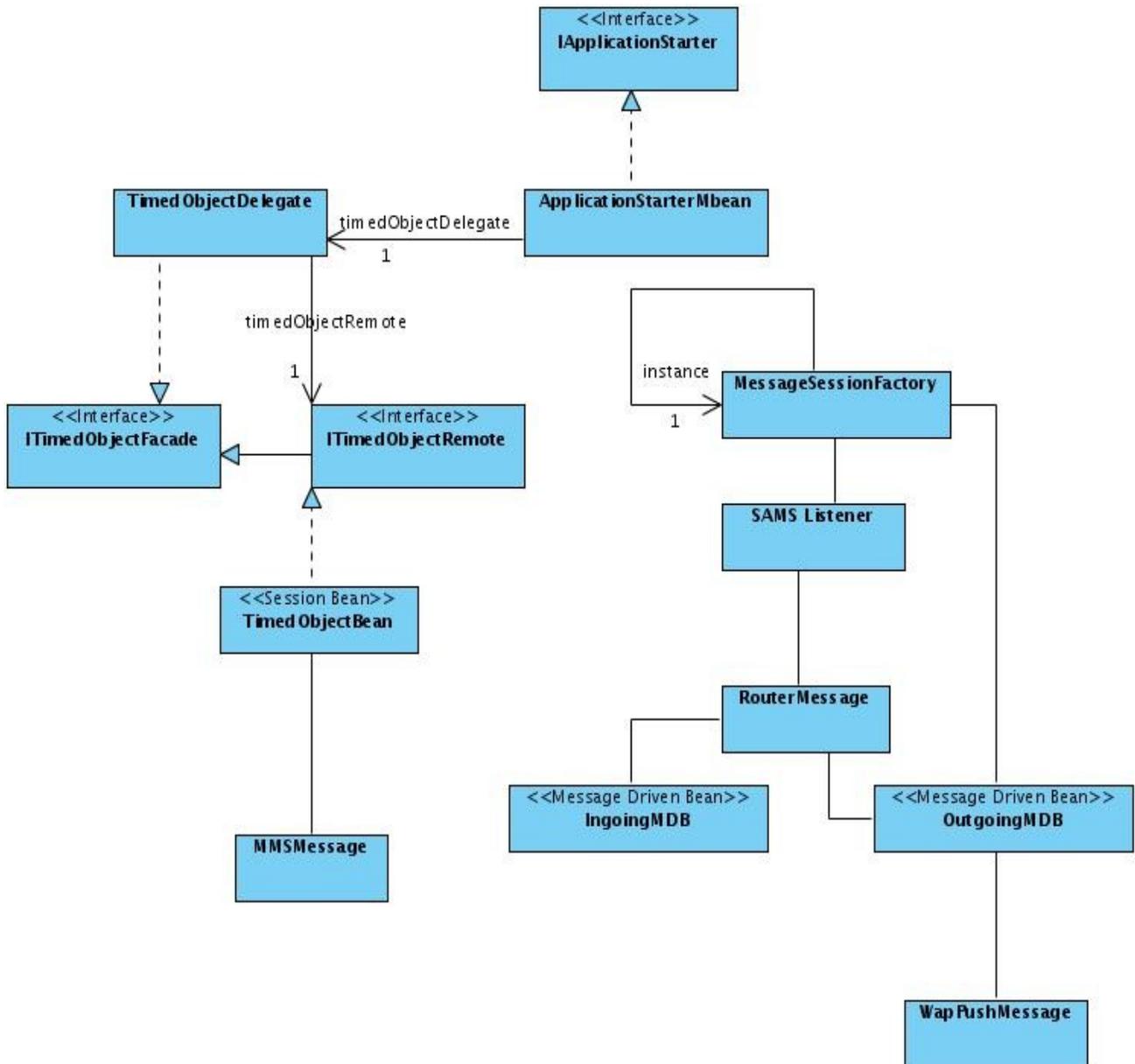
Wells, Don. 2006. Acceptance Tests. *Extreme Programming*. [En línea] 2006. [Citado el: 20 de Mayo de 2007.]

Wikipedia. 2007. Control de versiones. *Wikipedia enciclopedia libre*. [En línea] 2007. [Citado el: 11 de Abril de 2007.] http://es.wikipedia.org/wiki/Control_de_versiones.

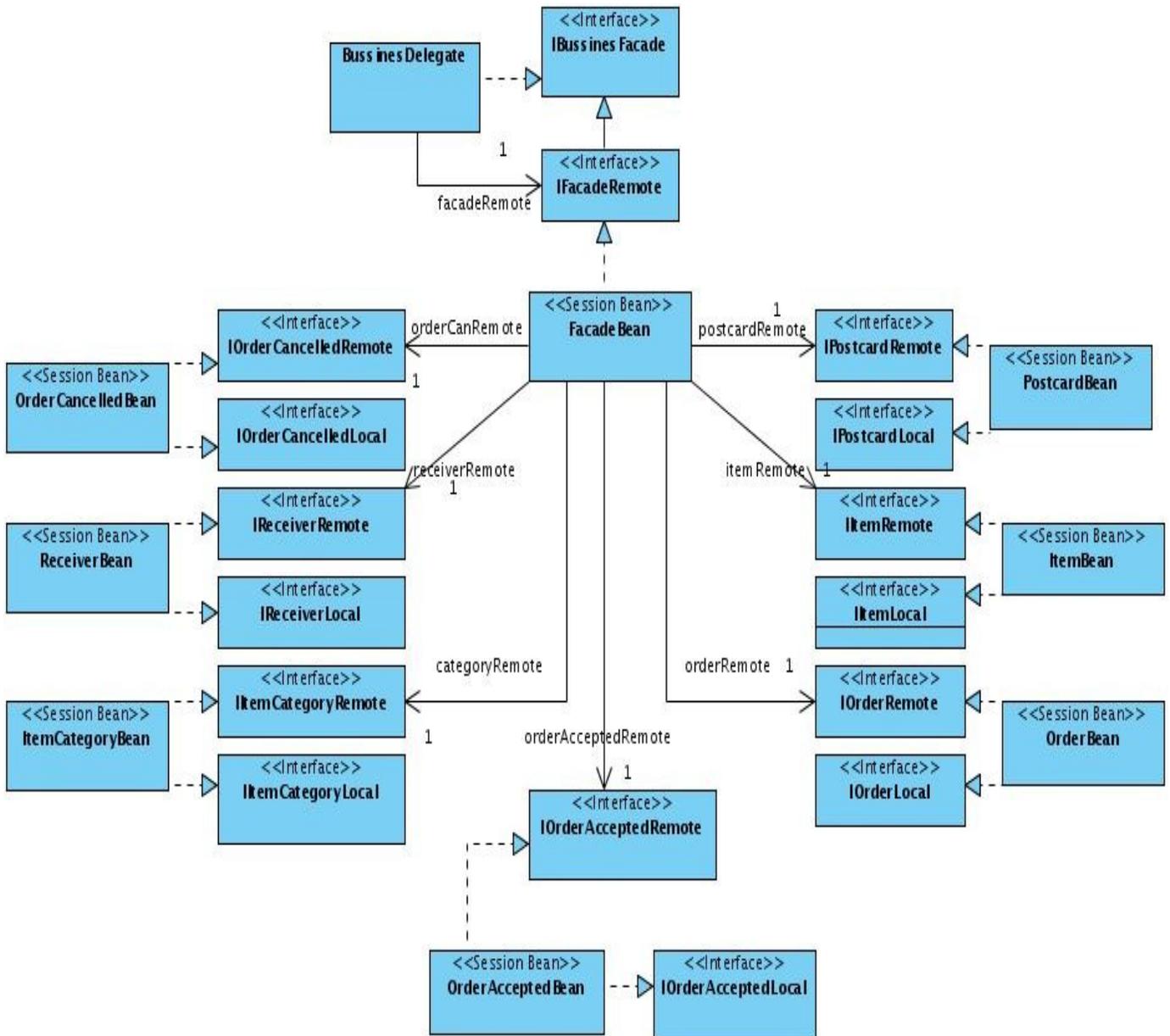
—. 2007. Integrated Development Environment. *Wikipedia encilopedia libre*. [En línea] 2007. [Citado el: 11 de Abril de 2007.] http://en.wikipedia.org/wiki/Integrated_Development_Environment.

ANEXOS

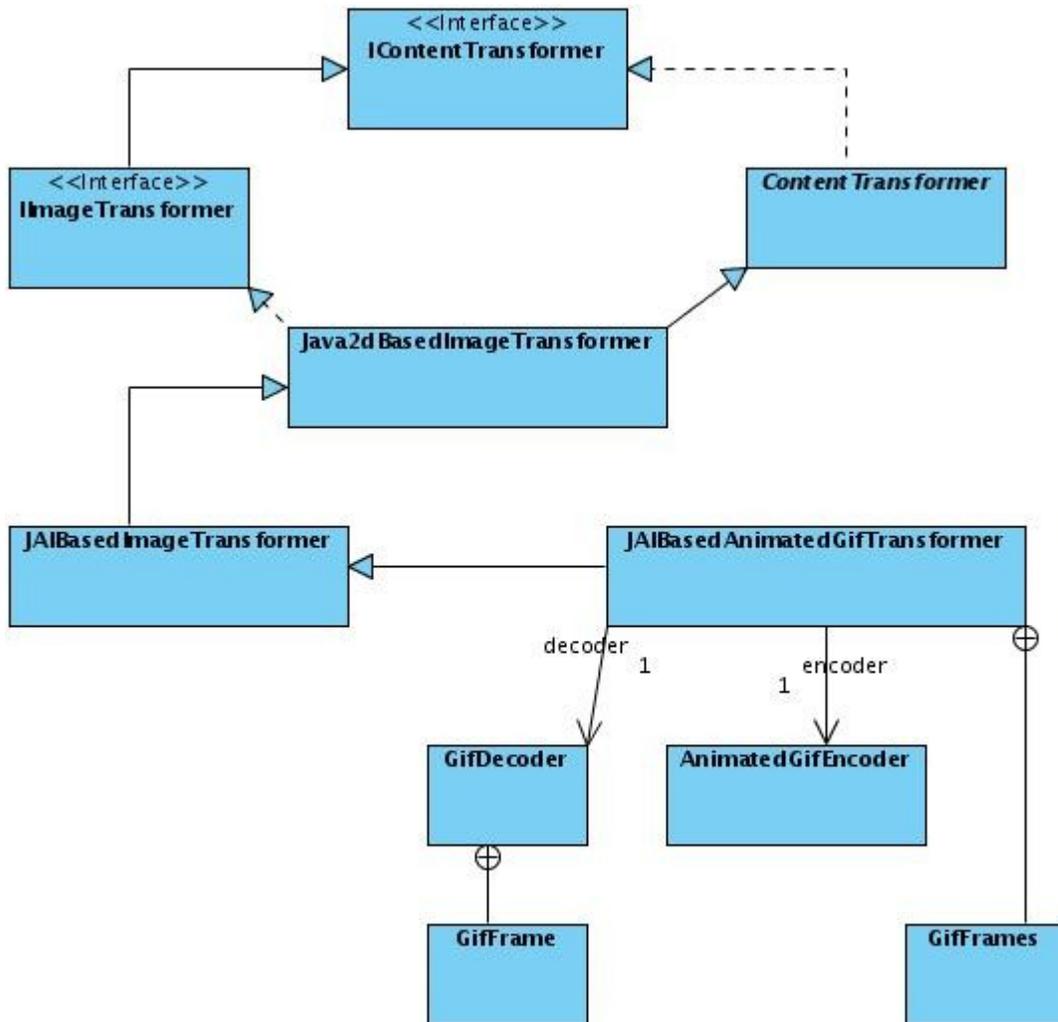
ANEXO 1. Diagramas de clases del Sistema



Anexo 1.1. Diagrama de clases del paquete *Messaging*



Anexo 1.2. Diagrama de clases del paquete *Dao*



Anexo 1.3. Diagrama de clases del paquete *Transcoder*

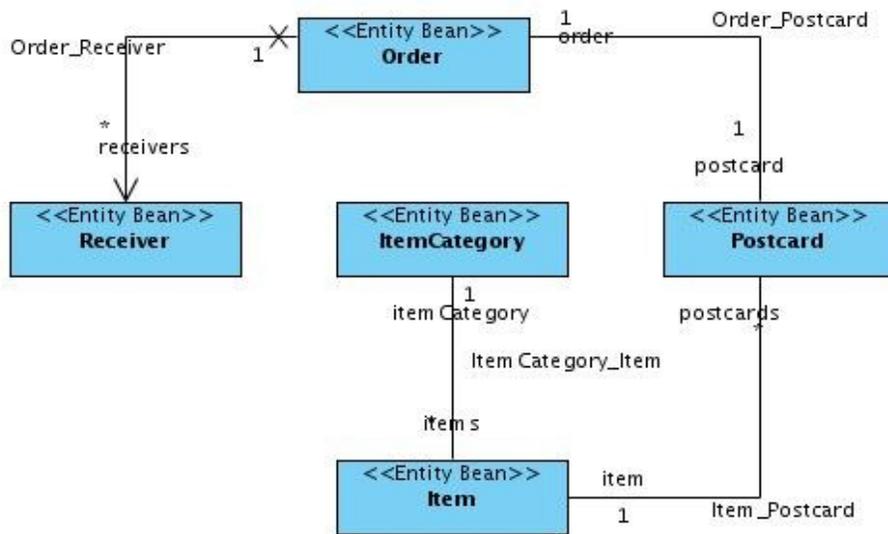
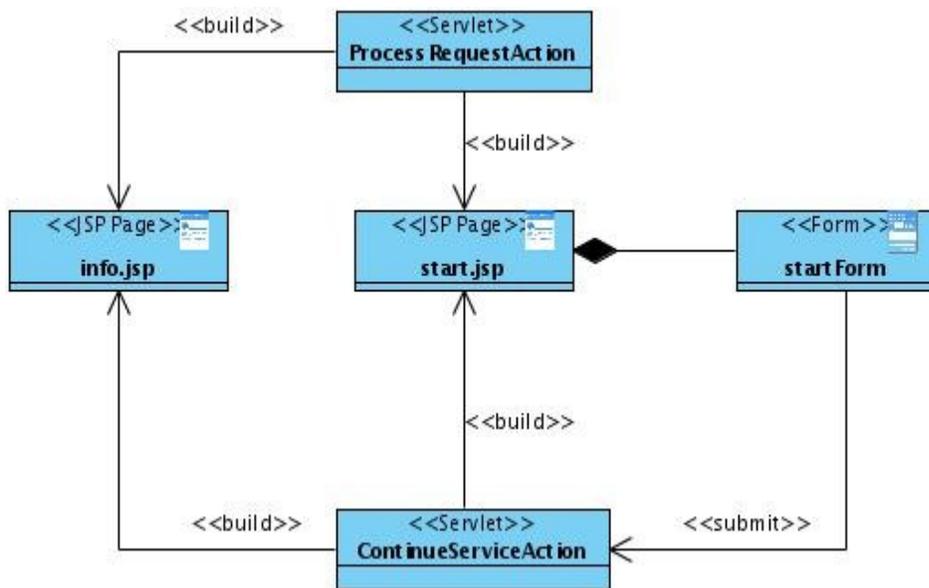
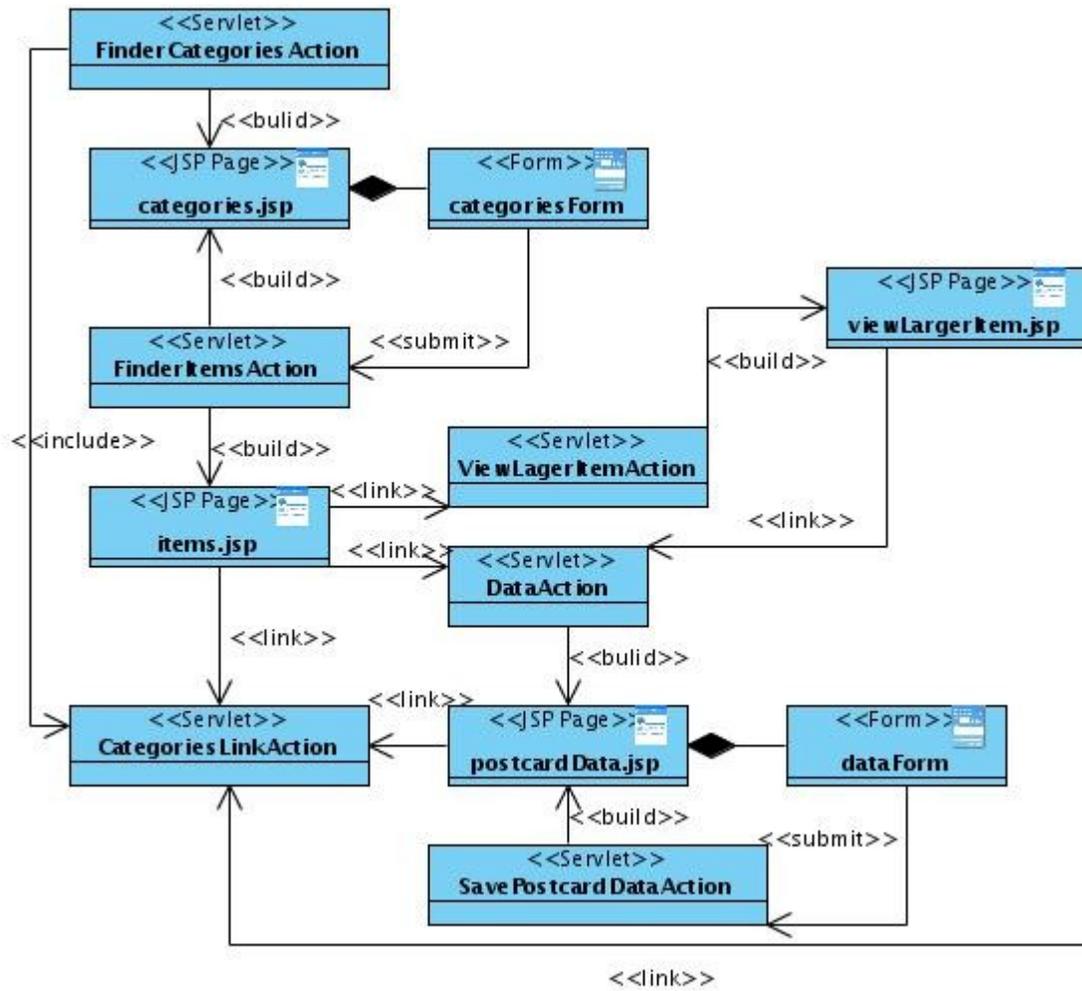


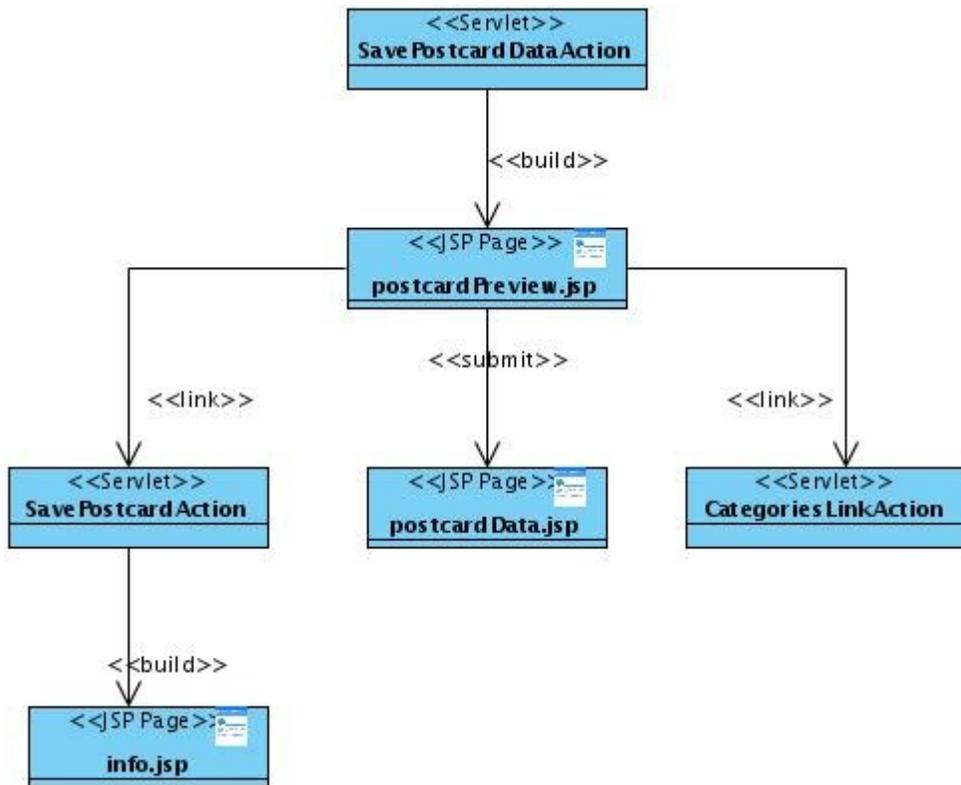
Figura 14. Diagrama de clases. Modelo de persistencia de datos



Anexo 1.4. Diagrama de clases WAP (parte I)



Anexo 1.5. Diagrama de clases WAP (parte II)



Anexo 1.6. Diagrama de clases WAP (parte III)

ANEXO 2. Pruebas de Aceptación

Pruebas de aceptación para la historia de usuario **Gestionar Solicitud de Servicio**

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1 – Gestionar solicitud de servicio.
Nombre: Gestionar solicitud de servicio.	
Descripción: Prueba para verificar la funcionalidad de recibir los mensajes de solicitud.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. La sesión de SAMS debe estar creada. Se enviará un SMS con la palabra reservada válida (<i>postcard</i>).	
Entrada / Pasos de ejecución: Se intenta recibir el mensaje enviado con los datos válidos.	
Resultado Esperado: El mensaje es recibido sin errores.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.1. Gestionar solicitud de servicio

Caso de Prueba de Aceptación	
Código: HU1_P2	Historia de Usuario: 1 – Gestionar solicitud de servicio.
Nombre: Generar pre-orden.	
Descripción: Prueba para verificar la funcionalidad de generar la pre-orden.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se utilizará un mensaje con datos válidos.	
Entrada / Pasos de ejecución: Se intenta generar una pre-orden con los datos recibidos.	
Resultado Esperado: La pre-orden es generada sin errores.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.2. Generar pre-orden

Caso de Prueba de Aceptación	
Código: HU1_P3	Historia de Usuario: 1 – Gestionar solicitud de servicio.
Nombre: Enviar <i>Wap Push</i> .	
Descripción: Prueba para verificar la funcionalidad de enviar un <i>Wap Push</i> con la dirección URL para la personalización de la postal.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. La sesión de SAMS debe estar creada. Se utilizará un mensaje <i>Wap Push</i> con datos válidos.	
Entrada / Pasos de ejecución: Se intenta enviar un mensaje <i>Wap Push</i> con la dirección URL a un número válido.	
Resultado Esperado: El mensaje <i>Wap Push</i> es enviado y recibido sin errores.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.3. Enviar *Wap Push*

Pruebas de aceptación para la historia de usuario **Gestionar Orden**

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2 – Gestionar orden.
Nombre: Chequear ordenes completadas.	
Descripción: Prueba para verificar la funcionalidad de chequear las órdenes que han sido completadas.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se establece cada que tiempo se efectuará el chequeo. Se completarán órdenes.	
Entrada / Pasos de ejecución: Se intenta encontrar las órdenes que han sido completadas.	
Resultado Esperado: Se encuentra las órdenes completadas.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.4. Chequear órdenes completadas

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2 – Gestionar orden.
Nombre: Conformar MMS.	
Descripción: Prueba para verificar la funcionalidad de conformar MMS con los datos de la postal.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se utilizarán datos de postal válidos.	
Entrada / Pasos de ejecución: Se intenta confeccionar un mensaje MMS.	
Resultado Esperado: Se genere correctamente el mensaje MMS.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.5. Conformar MMS

Caso de Prueba de Aceptación	
Código: HU2_P3	Historia de Usuario: 2 – Gestionar orden.
Nombre: Enviar MMS.	
Descripción: Prueba para verificar la funcionalidad de enviar el mensaje MMS.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. La sesión de SAMS debe estar creada. Se utilizarán un mensaje MMS válido.	
Entrada / Pasos de ejecución: Se intenta enviar un mensaje MMS.	
Resultado Esperado: Se envíe correctamente el mensaje MMS.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.6. Enviar MMS

Pruebas de aceptación para la historia de usuario **Chequear existencia de pre-orden**

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 3 – Chequear existencia de pre-orden.
Nombre: Chequear existencia de pre-orden.	
Descripción: Prueba para verificar la funcionalidad de chequear que el código introducido por un usuario corresponde a una pre-orden válida.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se utilizará un código de pre-orden válido.	
Entrada / Pasos de ejecución: Se intenta encontrar la pre-orden solicitada y mostrar la interfaz Wap para la personalización.	
Resultado Esperado: Se encuentre la pre-orden solicitada sin errores y se muestre la interfaz Wap correctamente.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.7. Chequear existencia de orden

Pruebas de aceptación para la historia de usuario **Gestionar servicio**

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4 – Gestionar servicio.
Nombre: Continuar con la personalización.	
Descripción: Prueba para verificar la funcionalidad de chequear cuando el usuario escoja continuar con la personalización de la postal.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se utilizará un código de pre-orden válido.	
Entrada / Pasos de ejecución: Se intenta re direccionar al usuario para que continúe con la personalización de su postal.	
Resultado Esperado: Se re direcciona correctamente al usuario.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.8. Continuar con la personalización

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: 4 – Gestionar servicio.
Nombre: Eliminar pre-orden.	
Descripción: Prueba para verificar la funcionalidad de eliminar una pre-orden por parte del usuario.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se utilizará un código de pre-orden válido.	
Entrada / Pasos de ejecución: Se intenta eliminar pre-orden por parte del usuario.	
Resultado Esperado: Se elimine correctamente la pre-orden.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.9. Eliminar pre-orden

Caso de Prueba de Aceptación	
Código: HU4_P3	Historia de Usuario: 4 – Gestionar servicio.
Nombre: Servicio no completado.	
Descripción: Prueba para verificar la funcionalidad de que las órdenes que han sido canceladas por parte del usuario se registren.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se utilizará datos de un pre-orden eliminada válidos.	
Entrada / Pasos de ejecución: Se intenta registrar datos de la pre-orden eliminada.	
Resultado Esperado: Se registre la pre-orden eliminada sin errores.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.10. Servicio no completado

Pruebas de aceptación para la historia de usuario **Crear orden**

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5 – Crear orden.
Nombre: Crear orden.	
Descripción: Prueba para verificar la funcionalidad de confeccionar la orden con los datos introducidos por el usuario.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se utilizará datos válidos.	
Entrada / Pasos de ejecución: Se intenta confeccionar la orden con los datos introducidos.	
Resultado Esperado: Se confeccione la orden sin errores.	
Evaluación de la Prueba: Prueba satisfactoria	

Anexo 2.11. Crear Orden

GLOSARIO

API: (*Application Programming Interface*) Interfaz de Programación de Aplicaciones.

CHTML: (*Compact HTML*) HTML compacto. Es un subconjunto de HTML para dispositivos pequeños de información.

GPRS: (*General Packet Radio Service*) Servicio General de Paquetes de Radio. Es un estándar de comunicación para teléfonos móviles que transmite la información por grupos significativos o paquetes.

GSM: (*Global System for Mobile Communications*) Sistema Global para comunicaciones Móviles. Sistema de telefonía celular digital para comunicaciones móviles de segunda generación.

HTML: (*Hyper Text Markup Language*) Lenguaje de Etiquetas de Hipertexto. Es un lenguaje de etiquetas diseñado para estructurar textos y presentarlos en forma de hipertexto, es el formato estándar de las páginas web.

IDE: (*Integrated Development Environment*) Entorno Integrado de Desarrollo.

iMode: Conjunto de tecnologías y protocolos diseñados para poder navegar a través de mini páginas diseñadas específicamente para dispositivos móviles.

JAVA: Lenguaje de programación multiplataforma desarrollado por Sun Microsystem.

JDBC: (*Java Database Connectivity*). Es un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

JMS: (*Java Message Service*) Servicios de Mensajería de Java. Es un estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma de Java EE crear, enviar, recibir y leer mensajes.

MMS: (*Multimedia Messaging System*) Servicio de Mensajería Multimedia. Es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia.

MMSC: (*Multimedia Messaging System Center*) Centro de Mensajería Multimedia. Es un elemento de la red de telefonía móvil cuya función es la de enviar y recibir mensajes MMS.

MySQL: Sistema gestor de base de datos.

Plugins: Es un programa adicional que puede ser añadido a una aplicación para aumentar la funcionalidad de esta.

PPG: (*Push Proxy Gateway*). Elemento encargado de soportar la funcionalidad Push de contenidos hacia los terminales móviles.

Portlets: Son componentes de interfaz de usuario que son gestionadas y visualizadas en un portal web.

RMI: (*Java Remote Method Invocation*). Es un mecanismo ofrecido en Java para invocar un método remotamente.

SMS: (*Short Message Service*) Servicio de Mensajes Cortos. Es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos.

SMSC: (*Short Message Service Center*) Centro de Servicios de Mensajes Cortos. Es un elemento de la red de telefonía móvil cuya función es la de enviar y recibir mensajes SMS.

SVN: Sistema de Control de Versiones.

URL: (*Uniform Resource Locator*) Localizador Uniforme de Recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

VAS: (Value Added Services) Servicios de Valor Agregado.

VASP: (Value Added Services Provider) Proveedor de Servicios de Valor Agregado.

WALL: (Wireless Abstraction Library). Es la combinación de una API Java, y una serie de librerías de tags (taglibs JSP) que permiten construir páginas JSP portables entre dispositivos.

WURFL: (Wireless Universal Resource File). Es un fichero XML donde se expresan, de manera compacta, las características de diversos dispositivos móviles.

XHTML: (eXtensible Hypertext Markup Language) Lenguaje Extensible de Marcado de Hipertexto. XHTML es la versión XML de HTML.

XML: (eXtensible Markup Language) Lenguaje de Marcas Extensible. Es un metalenguaje extensible de etiquetas.

XP: (Extreme Programming) Programación Extrema. Es una metodología ágil de desarrollo de software.