

Universidad de las Ciencias Informáticas
Facultad 2



Título: Sistema de información sobre la
Serie Nacional de Béisbol

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

Autores: Jaisniel Pérez Evora

Marcos Cruz Cué

Tutores: Ing. Dariena Ramirez Luján

Ing. Lex Karel Zayas Hernández

Consultante: Lic. Noel García Guimeras

Ciudad de la Habana, Junio de 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Jaisniel Pérez Evora
Firma del Autor

Marcos Cruz Cué
Firma del Autor

Ing. Dariena Ramirez Luján
Firma del Tutor

Ing. Lex Karel Zayas Hernández
Firma del Tutor

Lic. Noel García Guimeras
Firma del Consultante

AGRADECIMIENTOS

A nuestro Comandante en Jefe Fidel Castro Ruz, por darnos la posibilidad de llevar adelante el sueño de construir un mundo mejor. A la Revolución por confiar en nosotros para construir una universidad de excelencia. A la Universidad de las Ciencias Informáticas, por hacer de sus jóvenes hombres del futuro. A nuestros tutores Dariena y Lex Karel, a nuestro consultor Noel, por su dedicación; al equipo de Procyon, Almeida, York, Abel, Glicerio, etc.

Marcos y Jaisniel

A todos mis compañeros de aula durante todos estos años. A todos los que de una forma u otra me han ayudado en mi vida como estudiante. A toda mi familia y amigos, en especial a mis padres, a mi hermano, a Sergo, Ailyn, Yare y todos los que siempre me han apoyado y alentado para seguir adelante.

Jaisniel

A mis compañeros de aula durante todos estos años. A mis padres y abuelos por su paciencia y apoyo. A mi compañera por su apoyo y comprensión. A mis amigos y a todos los que de una forma u otra me apoyaron o influyeron en la realización de este trabajo.

Marcos

DEDICATORIA

Quisiera dedicarle este trabajo primeramente a la Revolución por darnos la oportunidad de estudiar, al Comandante en Jefe por confiar en nosotros. A mi familia por su apoyo y paciencia. A mi compañera por estar siempre ahí para apoyarme. Al colectivo de trabajadores de la entidad Procyon, porque sin ellos esta tesis no hubiera sido posible.

Marcos

A la Revolución y al Comandante en Jefe. A toda mi familia, muy especial para mi Mamá, a todos los que me han apoyado en los momentos difíciles y a mis compañeros de Procyon.

Jaisniel

RESUMEN

En la actualidad existe un gran auge de las comunicaciones inalámbricas a través de dispositivos móviles. Este acontecimiento ha provocado el surgimiento y desarrollo de una amplia gama de servicios y aplicaciones llevado a cabo por las diferentes empresas que laboran en este campo. La entidad cubana Procyon, se dedica al desarrollo de aplicaciones destinadas a la telefonía móvil. Hoy día se encuentra enfrascada en la creación de una plataforma de desarrollo (Blueeye), la cual podrá ser expandida a interés de esta entidad para brindar servicios y obtener ganancias a través de los mismos. De tal situación nació la idea de desarrollar una aplicación que brinde el servicio de información sobre la Serie Nacional de Béisbol, el cual será adicionado a dicha plataforma. Blueeye brinda un sistema de gestión de información, que es ofrecida a los clientes por solicitudes realizadas a través de un SMS. Los mismos pueden definir la vía de obtención de la información solicitada, que es presentada a través de un sitio WAP o por medio de un SMS de respuesta que contiene la información requerida. El sistema posee un sitio de administración Web que posibilita la actualización de la información en la base de datos. Después de un estudio y análisis de las herramientas a utilizar así como de las aplicaciones similares existentes, se decidió proceder con la creación de tal aplicación. La aplicación presenta gran aceptabilidad y constituye una fuente de información sencilla y de inmediato acceso como lo exige el contenido que será brindado.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN.....	7
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	10
1.1 INTRODUCCIÓN.....	10
1.2 ESTADO DEL ARTE DE LOS SVA.....	10
1.3 TELEFONÍA MÓVIL.....	13
1.3.1 Servicios de Valor Agregado.....	14
1.4 METODOLOGÍA DE DESARROLLO.....	15
1.5 PLATAFORMA DE DESARROLLO.....	18
1.6 TECNOLOGÍAS UTILIZADAS.....	19
1.6.1 Enterprise JavaBeans (EJB).....	19
1.6.3 WURFL-WALL.....	20
1.6.4 JSP y Servlet.....	20
1.7 SERVIDOR DE APLICACIONES.....	22
1.8 GESTOR DE BASES DE DATOS.....	22
1.8.1 Entorno Integrado de Desarrollo (IDE).....	23
1.8.1.1 ¿Por qué Eclipse?.....	23
1.8.2 Apache Ant.....	23
1.8.3 SVN (Subversion).....	24
1.9 CONCLUSIONES.....	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	25
2.1 INTRODUCCIÓN.....	25
2.2 OBJETIVOS ESTRATÉGICOS DE PROCYON.....	25
2.3 OBJETO DE AUTOMATIZACIÓN.....	25
2.4 PROPUESTA DEL SISTEMA.....	25
2.4.1 Flujo de procesos del sistema propuesto.....	27
2.5 ANÁLISIS COMPARATIVO DE OTRAS SOLUCIONES EXISTENTES CON LA PROPUESTA.....	27
2.6 CONCLUSIONES.....	28
CAPÍTULO 3: EXPLORACIÓN Y PLANIFICACIÓN	29
3.1 INTRODUCCIÓN.....	29
3.2 EXPLORACIÓN.....	29
3.2.1 Historias de usuario.....	29

3.3	PLANIFICACIÓN.....	32
3.3.1	Estimación de esfuerzo por historias de usuario.....	32
3.3.2	Plan de Iteraciones.....	33
3.3.2.1	Primera Iteración.....	33
3.3.2.2	Segunda Iteración.....	33
3.3.2.3	Tercera Iteración.....	34
3.4	PLAN DE DURACIÓN DE LAS ITERACIONES.....	34
3.5	PLAN DE ENTREGA.....	34
3.6	CONCLUSIONES.....	35
CAPÍTULO 4: DISEÑO, IMPLEMENTACIÓN Y PRUEBA.....		36
4.1	INTRODUCCIÓN.....	36
4.2	DISEÑO.....	36
4.2.1	Patrones de Diseño.....	36
4.2.1.1	Modelo Vista Controlador.....	37
4.2.1.2	Fábrica Abstracta.....	38
4.2.1.3	Adaptador.....	38
4.2.1.4	Iterador.....	38
4.3	IMPLEMENTACIÓN.....	38
4.3.1	Primera Iteración.....	39
4.3.1.1	Tareas de las historias de usuario desarrolladas en la primera iteración.....	39
4.3.2	Segunda Iteración.....	41
4.3.2.1	Tareas de las historias de usuario desarrolladas en la segunda iteración.....	42
4.3.3	Tercera Iteración.....	44
4.3.3.1	Tareas de las historias de usuario desarrolladas en la tercera iteración.....	45
4.4	FASE DE PRUEBA.....	47
4.4.1	Pruebas de Aceptación.....	47
4.5	CONCLUSIONES.....	47
CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.....		48
5.1	INTRODUCCIÓN.....	48
5.2	CARACTERÍSTICAS DEL PROYECTO.....	48
5.2.1	Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.....	50
5.3	ESTIMACIÓN.....	51
5.4	BENEFICIOS TANGIBLES E INTANGIBLES.....	53
5.5	ANÁLISIS DEL COSTO.....	53
5.6	CONCLUSIONES.....	54
CONCLUSIONES GENERALES.....		55
RECOMENDACIONES.....		56
BIBLIOGRAFÍA.....		57

ANEXOS..... 59

 Anexo 1. Diagrama de Clases Del Subsistema..... 59

 Anexo 2. Pruebas de Aceptación..... 67

GLOSARIO 73

INTRODUCCIÓN

Las tecnologías inalámbricas han revolucionado el mundo de las comunicaciones, experimentando un auge y desarrollo de forma exponencial en los últimos años, esto se debe especialmente a la telefonía móvil.

La telefonía móvil desde sus inicios, a finales de los 70, ha transformado notablemente las actividades que se realizan a diario. En un principio fue concebida sólo para la transmisión de voz; sin embargo en la actualidad este concepto se ha ampliado, siendo capaz de brindar otros servicios como transferencia de datos, audio y video, aunque con algunas limitaciones.

La telefonía móvil ha impactado y revolucionado el mundo. Dadas las diferentes prestaciones y servicios que nos brinda. Decir que se puede mantener una conversación con alguien a cualquier distancia mientras conducimos, revisar el correo en cualquier momento desde el lugar que te encuentres, ha convertido a los dispositivos móviles en un instrumento imprescindible para la sociedad, siendo reflejo de seguridad, productividad y conectividad inmediata ante cualquier necesidad.

Todo esto ha conducido a un inevitable aumento del uso de la telefonía celular, convirtiéndose en el principal medio de comunicación y en el que más espacios y entornos abarca por sus características trascendentales de tamaño y manejabilidad, las mismas le han dado el seudónimo de móvil, el cual es altamente descriptivo.

El crecimiento acelerado del uso de esta tecnología, ha sido objeto de atención de muchas empresas por todo el mundo, las cuales se han convertido en fieles proveedores de Servicios de Valor Agregado(SVA), donde la competencia por el liderazgo y el éxito financiero es muy difícil, existiendo por ello un constante aumento de los servicios y calidad de los mismos, entre estos servicios se encuentran: correo electrónico, ranking por votaciones, llamadas a emergencias, horóscopo, estado del tiempo, localización, video llamada, Internet y televisión digital.

En la actualidad, la entidad cubana Procyon lleva a cabo el desarrollo de una serie de proyectos entre los cuales se encuentra la creación de una plataforma destinada a operadores móviles (Blueeye), que provee una serie de SVA como EtecsaPhone: guía telefónica para consultas, WAPWeather: brinda información actualizada del estado del tiempo, Voting: Servicio para votaciones; todos estos basados en solicitudes a través del servicio de mensajes cortos (*del inglés Short Messaging Service o SMS*), además de una serie de servicios avanzados. Debido a la importancia que tiene dicha plataforma y la

necesidad de que brinde la mayor cantidad de servicios con el objetivo de entrar a competir en el mercado internacional y aumentar su prestigio en el mismo, se ha decidido seguir trabajando en ella y desarrollar un sistema que brinde el servicio de envío de informaciones sobre la Serie Nacional de Béisbol, ante tal situación surge el siguiente **problema a resolver**.

¿Cómo brindar información actualizada sobre la Serie Nacional de Béisbol accediendo desde dispositivos móviles?

Para dar solución al mismo se plantea como **objeto de estudio**, los servicios de información para dispositivos móviles.

El **campo de acción** queda enmarcado en el servicio de información sobre la Serie Nacional de Béisbol para dispositivos móviles.

Para darle solución a nuestro sistema se planteó como **objetivo general** analizar, diseñar e implementar una aplicación JAVA EE que brinde información sobre la Serie Nacional de Béisbol a los dispositivos móviles.

De donde resultaron los siguientes **objetivos específicos**:

- Implementar una aplicación de envío de información utilizando Blueeye.
- Implementar un administrador Web para la actualización de la información.
- Implementar un portal WAP para el despliegue de la información.

Para cumplir los objetivos trazados se crearon las siguientes tareas:

- Análisis de los sistemas similares existentes.
- Selección de las tecnologías necesarias para implementar el sistema propuesto.
- Selección de una metodología que se acondicione a las características de nuestro proyecto para guiar el desarrollo del software.
- Captura de las necesidades del cliente.
- Definición de un plan de entrega.

- Definición de las prioridades de desarrollo.
- Creación de un sitio Web de administración.
- Creación de un sitio WAP.

Este trabajo está conformado por cinco capítulos que describen los estudios realizados, la propuesta del sistema y detalla la realización del mismo.

En el capítulo 1 se abordan los conceptos y términos elementales de la investigación, se hace un estudio de los sistemas similares, se seleccionan y exponen las tecnologías y herramientas utilizadas además de la metodología que guiará el proceso de desarrollo del software. En el capítulo 2 se plantean los objetivos estratégicos de la entidad Procyon, se presentan los procesos que serán objetos de automatización, se hace una propuesta del sistema y se expone el flujo de procesos del mismo. En el capítulo 3 se hace una exploración del software que se quiere implementar, se capturan las necesidades del cliente, se programan las iteraciones para la creación del sistema y se crea una planificación de estrategia del producto. En el capítulo 4 se generan los diagramas que se utilizan en el desarrollo del Sistema y se especifican las iteraciones, donde se describen las tareas de programación a desarrollar. En el capítulo 5 se hace un estudio de factibilidad del software a desarrollar.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 INTRODUCCIÓN

En el presente capítulo se tratarán los principales conceptos y términos que se utilizan como soporte teórico del sistema a diseñar e implementar, se realizará un análisis de los SVA de información a través de SMS y SMS WAP Push y se expondrán las metodologías, tecnologías y herramientas utilizadas en el desarrollo de la solución.

1.2 ESTADO DEL ARTE DE LOS SVA.

En el mundo de hoy muchas empresas dedican su labor a la prestación de servicios de telefonía celular. Sumado a esto, viene una serie de SVA para explotar al máximo las posibilidades que brindan las conexiones. Algunos de estos servicios son los que envían información a los usuarios a través de SMS, de forma tal que estos reciban toda la información pertinente que solicitaron. A continuación describiremos algunos de los servicios que se brindan a través del envío de SMS.

Estado del Tiempo: El usuario desea conocer el estado del tiempo para las próximas horas o el día siguiente, entonces envía un SMS hacia el número que se encuentra a la escucha para brindar servicios, especificando en este qué servicio desea recibir a través de palabras claves designadas por el proveedor. Una vez que el proveedor recibe el SMS lo procesa y genera un SMS WAP Push que contiene la dirección donde se encuentra desplegada la información que él necesita, posteriormente el usuario acepta dicho mensaje y es redireccionado al sitio WAP que contiene la información que fue solicitada. (Castillo, 2006)

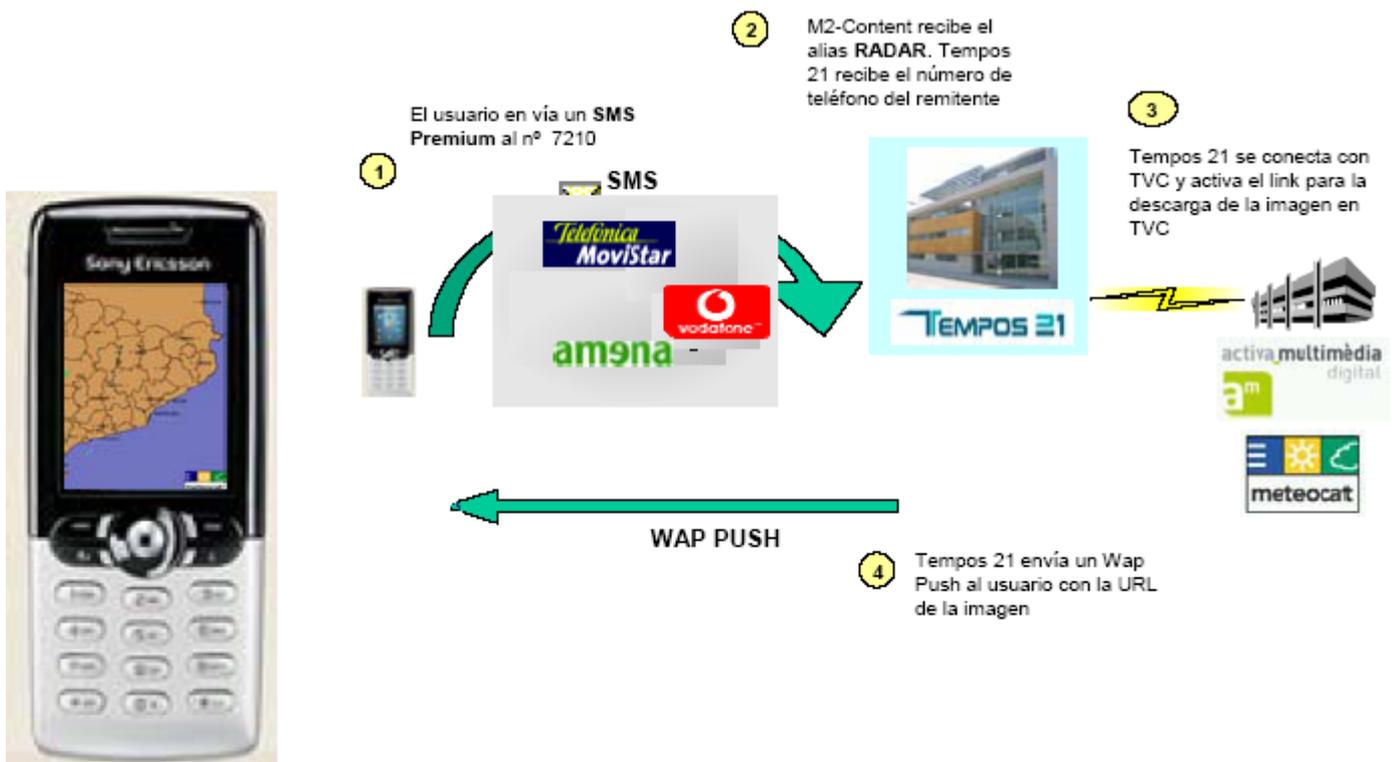


Figura 1: Dinámica desarrollada por Tempos 21 en el servicio del tiempo.

Deporte: Un servicio muy parecido al que se tiene como objetivo desarrollar es brindado por la empresa de servicios La Tercera. A continuación la figura 2 muestra una idea de cómo funciona dicho servicio.



Figura 2: Servicio de suscripción de Deportes.

Horóscopo: El horóscopo es un servicio muy común que siempre ha llamado la atención y ha sido brindado en las diferentes formas en que lo ha permitido la tecnología a lo largo del desarrollo de la humanidad, en la figura 3 se brinda una idea de como se comporta tal servicio.



Figura 3: Servicio de información de Horóscopo.

El costo de recibir estos servicios es muy pequeño y oscila generalmente entre los 0.15 y 0.25 USD, sin embargo, las ganancias para los proveedores y operadores son notablemente grandes. Esto se debe a que los usuarios son muchos y se encuentran en constante crecimiento.

Gran cantidad de los sistemas de envío de información a través de SMS y SMS WAP Push son muy similares, dando como resultado un esquema-protocolo que representa la secuencia de pasos que se llevan a cabo para cumplir este objetivo, ver figura 4.

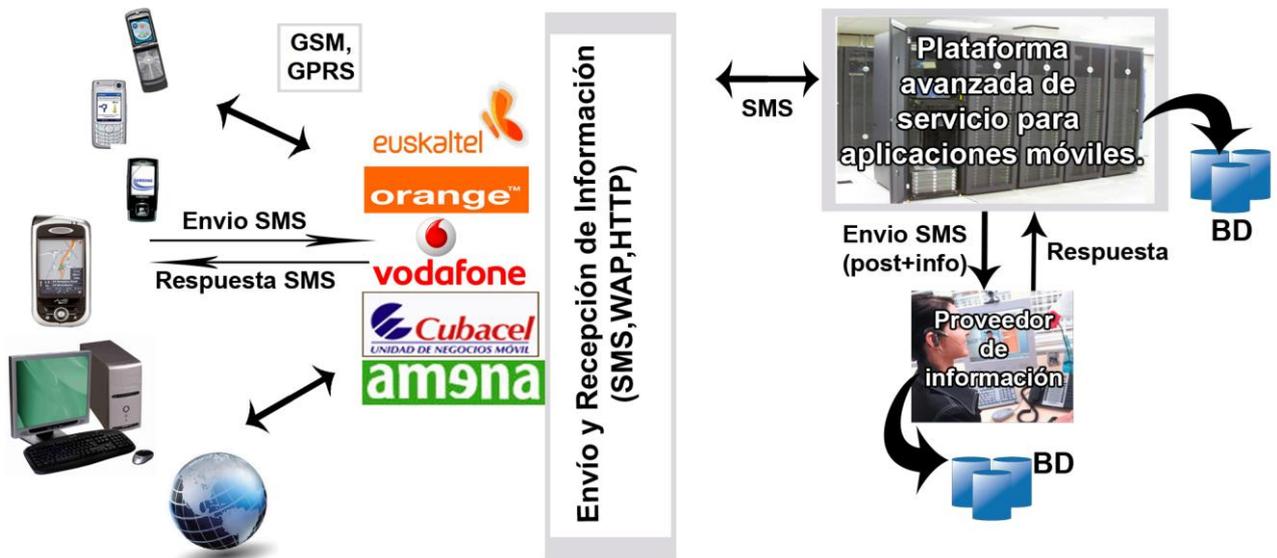


Figura 4: Esquema del Servicio de envío de información.

1.3 TELEFONÍA MÓVIL.

El auge que ha generado la telefonía celular se debe a la gran cantidad de servicios que hoy día es capaz de brindar. Esto se debe al desarrollo que esta ha experimentado, comenzando desde sus inicios sólo por la transmisión de voz, hasta nuestros días con la incorporación de transferencia de información y de video. Todo esto ha sido posible gracias al avance de las prestaciones de los dispositivos móviles y los diferentes estándares de comunicación.

Entre los estándares más difundidos por todo el mundo se encuentra el Sistema Global para las Comunicaciones Móviles (del inglés *Global System for Mobile Communications* o *GSM*). GSM define un sistema completo que incluye no sólo la interfaz radio, sino también una completa arquitectura de red, lo que permite el desarrollo de multitud de nuevos servicios sobre este estándar.

GSM tiene como funcionalidad principal la transmisión de voz, aunque también permite la transmisión de datos a baja velocidad (9.6 Kbps), aún así ha permitido el éxito absoluto del servicio de un sistema de mensajes cortos (SMS).

Los SMS permiten enviar y recibir mensajes de texto de hasta 160 caracteres alfanuméricos como máximo a otro teléfono a través del centro de mensajes de un operador de red; si el teléfono al que se envía el mensaje está apagado o fuera de cobertura el mensaje se almacena en la red y se entrega en cuanto el teléfono se conecta de nuevo.

Con el desarrollo del GSM surge entonces el Servicio General de Radio-Paquete (del inglés *General Packet Radio Service* o *GPRS*). La ventaja trascendental fue el cambio de conmutación de circuito a conmutación de paquetes, este servicio modificó una serie de factores que influyeron positivamente en la comunicación de las redes GSM. Ahora soporta teóricamente un mínimo de 40 Kbps hasta 110 Kbps para la transferencia de información, lo que posibilitó una extensión del servicio SMS surgiendo el Servicio de Mensajes Multimedia para permitir el intercambio de contenido multimedia entre los teléfonos móviles y otros dispositivos habilitados (adicional al texto, imágenes, audio y video).

El Protocolo de Aplicaciones Inalámbricas (del inglés *Wireless Application Protocol* o *WAP*) es un estándar abierto a nivel internacional para aplicaciones que utilizan las comunicaciones de este tipo, por ejemplo el acceso a servicios de Internet desde un teléfono móvil. Para la presentación de contenidos utiliza el Lenguaje de Marcado para Telefonía Inalámbrica (del inglés *Wireless Markup Language* o *WML*). La pila de protocolos de WAP 1 no es compatible directamente con la de Internet, por lo que se exige la presencia de un nodo pasarela para hacer de intermediario en la comunicación

entre un terminal WAP y un servidor de contenidos WAP residente en Internet. WAP 1 ha sido objeto de fuertes críticas por diversos motivos, que incluyen la pobreza del soporte de los Gráficos Monocromos (del inglés *Wireless Bitmap* o WBMP) y las diferencias en las implantaciones de WAP en los terminales de distintos fabricantes, que constituyen un potencial problema de seguridad debido a que la Capa de Seguridad de Transporte Inalámbrica (del inglés *Wireless Transport Layer Security* o WTLS) no es muy robusta y además, por no ser compatible con las capas de seguridad usadas en Internet, los contenidos deben estar en claro en la pasarela WAP.

WAP 2.0 surge en el 2004, esta versión es una reingeniería de WAP que utiliza el Lenguaje de Marcado de Hipertexto Extensible del perfil Móvil (del inglés *Extensible Hypertext Markup Language Mobile Profile* o XHTML MP), que es un subconjunto de Lenguaje de Marcado de Hipertexto Extensible (del inglés *Extensible Hypertext Markup Language* o XHTML) que incluye el XHTML básico, Hoja de estilo en cascada WAP (del inglés *WAP Cascading Style Sheets* o WCSS), y un subconjunto de Hoja de Estilo en Cascada 2 (del inglés *Cascading Style Sheets 2* o CSS2) más ciertas extensiones específicas para móviles, como lenguajes para la presentación de contenidos, mejorando, por ejemplo, el soporte de los gráficos (incluye color). De esta forma se consigue que el diseño de contenidos con WAP 2.0 sea muy similar al diseño de contenidos para la WEB para navegadores de otros dispositivos. (Daniel Meyer #97-29843) y (Ing. Scándolo)

Algo que está estrechamente relacionado con WAP es la Tecnología WAP Push, que es un tipo de SMS que se utiliza para acceder de una manera conocida, sencilla y fácil a un sitio o página WAP sin necesidad de introducir la dirección de la página en el navegador del dispositivo móvil de forma parecida al sistema de descarga de melodías y tonos vía SMS. El usuario envía un SMS y recibe otro de vuelta con un mensaje. Al aceptar el SMS se abre el navegador y lo dirige a la dirección WAP que ha solicitado. Muy parecido a cuando en nuestro correo electrónico recibimos un mensaje que contiene una dirección Web y hacemos clic sobre ella. (entumovil.net)

1.3.1 Servicios de Valor Agregado

En la telefonía móvil los SVA aparecen cuando los servicios básicos de SMS, Servicio de Mensajes Multimedia (del inglés *Multimedia Messaging Service* o MMS) y WAP- PUSH comienzan a utilizarse con el objetivo de obtener un servicio adicional. Hoy existe un amplio espectro de estos, entre los que podemos mencionar: estado del tiempo, horóscopo, noticias, estrenos de cine, deportes, envíos de postales, concursos, sorteos, etc.

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

- **Servicios basados en el contenido:** No poseen ningún tipo de procesamiento, encapsulado o tratamiento; en otras palabras, son los mensajes intercambiados entre los usuarios.
- **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.
- **Servicios complementarios:** Permiten la creación de contenidos por el usuario, como el envío de tarjetas postales o felicitaciones, también los compositores de mensajes completos con repositorios de contenidos elementales como imágenes y sonidos.
- **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, los de información meteorológica, los anuncios musicales o videos promocionales, los servicios de envío de fotografías capturadas por cámaras Web y los juegos.

1.4 METODOLOGÍA DE DESARROLLO

En la actualidad no existe una metodología general para desarrollar cualquier proyecto de desarrollo de software, sino que estas tienen prácticas específicas que las hacen más óptimas o no dependiendo de las características del proyecto.

Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo del software, los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio de orden y documentación, los métodos ligeros tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso. (Molpeceres, 2002)

Una de estas tendencias está representada por el Proceso Unificado de Desarrollo (del inglés *Rational Unified Process* o RUP), que es una metodología pesada y constituye uno de los procesos más generales de los existentes en la actualidad, ya que en realidad está pensado para adaptarse a cualquier proyecto. Se caracteriza por ser muy organizativo, se basa en roles y tiene una vasta documentación.

Por otro lado encontramos la Programación Extrema (del inglés *Extreme Programming* o XP), que pertenece a las metodologías ligeras (también conocida como ágiles) e intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. XP se caracteriza por desarrollar en pareja, mantener el mínimo de organización, el código pertenece a todos y es pobre en documentación. (Molpeceres, 2002)

Mientras que RUP intenta reducir la complejidad del software por medio de estructuras y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. (Molpeceres, 2002)

A continuación se expone en la figura 5 diferentes metodologías, que son caracterizadas en una gráfica de flexibilidad contra calidad. (Baird, 2003)

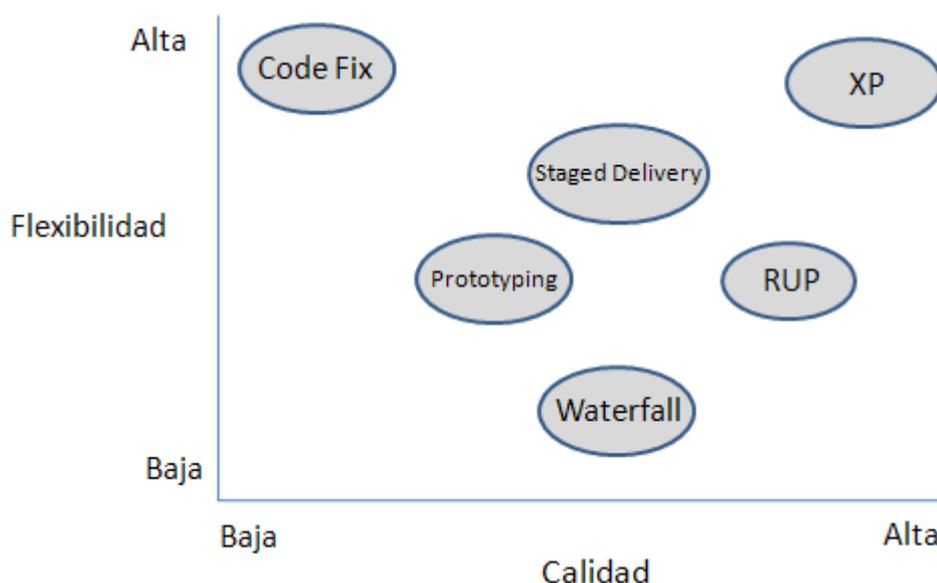


Figura 5 Comparación de las metodologías de desarrollo.

Para el desarrollo del proyecto se escogió como metodología de desarrollo XP, ya que se ajusta en gran medida al proyecto, promoviendo el trabajo en equipo y preocupándose por el aprendizaje de los desarrolladores. Se basa en la realimentación 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. Es adecuada especialmente para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico, es una metodología ágil especialmente pensada para proyectos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos y basados en nuevas tecnologías. (Beck, 2002)

A continuación se muestran varios aspectos definidos en la programación extrema que influyeron en la decisión de utilizar esta metodología para el desarrollo del sistema.

XP:

- Destacada en los procesos ágiles de desarrollo del software (desarrollo a corto plazo).
- Pone más énfasis en la adaptabilidad que en la previsibilidad.
- Considera los cambios de requisitos sobre la marcha como un aspecto natural, inevitable e incluso deseable en el desarrollo de proyectos.
- Recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- Programación en parejas (cada miembro realiza una acción que el otro no está haciendo actualmente: mientras que uno codifica las pruebas de unidad el otro piensa en la clase que satisfecerá la prueba, se sugiere a menudo que los dos socios cambien de papeles por lo menos cada media hora o después de que se haga una prueba de unidad).
- Desarrollo iterativo e incremental.
- Grupo pequeño y muy integrado (máximo 12 personas).
- Simplicidad (Propone el principio de hacer que la cosa más simple pueda funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana).

1.5 PLATAFORMA DE DESARROLLO.

Para el desarrollo de aplicaciones empresariales existen varias tendencias hoy día como son Microsoft. Net, LAMP, Java EE entre otras, una de las de más aceptación es la plataforma Java EE (del inglés Java Enterprise Edition) de Sun Microsystems.

Para el desarrollo de la aplicación se escogió la plataforma JAVA EE debido al vínculo que existe entre esta y las aplicaciones para la telefonía móvil. A nivel mundial 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 la utilización de JAVA EE es una política establecida de la entidad Procyon.

La arquitectura JAVA EE define un modelo dirigido a n-capas, que se seleccionan en dependencia del tipo de sistema, permitiendo mayor flexibilidad, que sean más robustas debido al encapsulamiento, y permite el desarrollo paralelo y una alta escalabilidad. (Corporation, 2003)

La plataforma JAVA EE es una tecnología para el diseño de componentes de transacción escalables y basados en servidor. Comprende un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones distribuidas. JAVA EE incluye varias especificaciones de Interfaz de Programación de Aplicaciones (del inglés *Application Programming Interface* o API), tales como Conectividad de la Base de Datos de Java (del inglés *Java Database Connectivity* o JDBC), Invocación de Métodos Remotos (del inglés *Java Remote Method Invocation* o RMI), Servicio de Mensajería Java (del inglés *Java Message Services* o JMS), Servicios Web y Lenguaje de Marcas Extensible (del inglés *Extensible Markup Language* o XML), todos estos definidos 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), Páginas del Servidor Java (del inglés *Java Server Pages* o JSP) 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, seguridad, escalabilidad, concurrencia, gestión de los componentes desplegados y que el servidor de aplicaciones puede manejar transacciones, lo cual significa que los desarrolladores pueden concentrarse más en la lógica del 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 el servidor de aplicaciones, hay muchas herramientas de código abierto disponibles para extender la plataforma o para simplificar el desarrollo.

1.6 TECNOLOGÍAS UTILIZADAS.

Este epígrafe detalla las tecnologías utilizadas en el desarrollo del sistema. Las mismas fueron seleccionadas valorando sus ventajas con las necesidades del sistema a desarrollar para su óptimo funcionamiento una vez que sea puesto en práctica, y también cumpliendo con las políticas de desarrollo establecidas en la entidad Procyon.

1.6.1 Enterprise JavaBeans (EJB).

Como tecnología del lado del servidor para la arquitectura de componentes de JAVA EE, se utilizó EJB, ello permite agilizar y simplificar el desarrollo distribuido, transaccional, seguro y portátil de aplicaciones así como el manejo de la concurrencia, la gestión de mensajes y la adaptación en tiempo de despliegue. Existen varios tipos de EJB los cuales se describen a continuación:

- 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).

1.6.2 WML.

En el desarrollo de interfaces WAP se utiliza el lenguaje WML, su origen es del XML. Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (del inglés *Personal Digital Assistant* o PDA) dotados de tecnología

WAP. Es una versión reducida del Lenguaje de Marcado de Hipertexto (del inglés *HyperText Markup Language* o 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, utilizado para la creación de páginas Web convencionales. Un consorcio formado por Nokia, Phone.com, Motorola y Ericsson, conocido como el Foro WAP, define la sintaxis, las variables y los 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 WML Script que es un lenguaje bastante similar a JavaScript.

1.6.3 WURFL-WALL.

En el mundo existe una gran diversidad de dispositivos móviles, donde las características de los mismos difieren en gran medida, por ejemplo en el tamaño de la pantalla, en el soporte de audio e imágenes compatibles. Debido a toda esta irregularidad, para lograr estandarizar los servicios a todos los tipos de dispositivos móviles se utilizó el fichero universal de descripción de dispositivos móviles (del inglés *Wireless Universal Resource File* o WURFL) y un API nombrado WALL (del inglés *Wireless Abstraction Library*), donde WURFL es un repositorio que contiene un XML con información de los dispositivos móviles que existen y WALL es un conjunto de librerías que permite utilizar la información brindada por el WURFL (Passani, 08), para poder lograr enviar información que se adapte a las características de los dispositivos móviles y lograr así mostrar toda la información deseada sin problema alguno.

1.6.4 JSP y Servlet.

JSP es una tecnología que permite mezclar HTML estático con HTML generado dinámicamente. Muchas páginas Web construidas con programas de Interfaz de Entrada Común (del inglés *Common Gateway Interface* o CGI) son casi estáticas, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI, incluyendo los Servlets, hacen que se genere la página completa mediante el programa, incluso aunque la mayoría de ellas sea siempre lo mismo. JSP permite crear dos partes de forma separada. (Hall, 08)

Ventajas de la tecnología JSP:

- La parte dinámica está escrita en Java, no en Visual Basic, otro lenguaje específico de Microsoft, por eso es mucho más poderosa y fácil de usar.
- Es portable a otros sistemas operativos y servidores Web.
- Pueden construir el HTML, dejando espacio para que otro programe el Servlet e inserte el contenido dinámico.
- Permite usar Servlet en vez de un programa separado para generar las partes dinámicas. (Hall, 08)

Los Servlets son las respuestas de la tecnología Java a la programación CGI. Son programas que se ejecutan en un servidor Web y construyen páginas Web. Los Servlets Java son más eficientes, fáciles de usar, más poderosos, más portables, y más baratos que el CGI tradicional y otras muchas tecnologías del tipo CGI. (Hall, 08)

Ventajas de la tecnología Servlets:

- **Eficiencia:** Con CGI tradicional se arranca un nuevo proceso para cada solicitud del Protocolo de Transferencia de Hipertexto (del inglés *HyperText Transfer Protocol* o HTTP). Si el programa CGI hace una operación relativamente rápida, la sobrecarga del proceso de arrancada puede dominar el tiempo de ejecución. Con los Servlets, la Máquina Virtual Java (del inglés *Java Virtual Machine* o JVM) permanece arrancada, y cada petición es manejada por un hilo Java de peso ligero y no por un pesado proceso del sistema operativo.
- **Conveniencia:** Junto con la conveniencia de poder utilizar un lenguaje familiar, los Servlets tienen una gran infraestructura para análisis automático y decodificación de datos de formularios HTML, leer y seleccionar cabeceras HTTP, manejar cookies, seguimiento de sesiones, y muchas otras utilidades.
- **Potencia:** Los Servlets Java permiten fácilmente hacer muchas cosas que son difíciles o imposibles con CGI normal. Por algo los Servlets pueden hablar directamente con el servidor Web, esto simplifica las operaciones que se necesitan para buscar imágenes y otros datos almacenados en situaciones estándar.
- **Portable:** Los Servlets están escritos en Java y siguen un API bien estandarizado. Consecuentemente, los Servlets escritos en el servidor JBOSS se pueden ejecutar sin

modificarse en Apache. Los Servlets están soportados directamente o mediante Plug-in en la mayoría de los servidores Web.

- **Barato:** Hay un número de servidores Web gratuitos o muy baratos que son buenos para publicar sitios Web de bajo nivel. Sin embargo, con la excepción del servidor Apache Tomcat, que es gratuito, la mayoría de los servidores Web comerciales son relativamente caros. Una vez que tengamos un servidor Web, no importa el coste del servidor, añadirle soporte para Servlets (si no viene pre configurado para soportarlos) es gratuito o muy barato. (Hall, 08)

1.7 SERVIDOR DE APLICACIONES.

Se utiliza el JBoss 4.2.1.GA como servidor de aplicaciones, es un servidor de código abierto que permite a los clientes crear, desplegar, integrar, organizar y presentar aplicaciones y servicios Web en una arquitectura orientada a servicios, proporciona una manera sencilla, abierta y asequible de modernizar la antigua infraestructura de aplicaciones. Disminuye la complejidad de desarrollo, mejora la experiencia del usuario final y resuelve la fricción en los procesos comerciales. Además, al ser de código abierto, JBoss Enterprise proporciona una flexibilidad que no se puede comparar con la de sus competidores y disminuye notablemente el coste total de su propiedad. (Red Hat, 2007)

1.8 GESTOR DE BASE DE DATOS.

Como gestor de base de datos se utiliza MySQL, el cual fue diseñado principalmente para la velocidad, muy necesario para brindar información a los usuarios de forma inmediata. Una de las mayores ventajas es que cuando MySQL encuentra un índice en la búsqueda evita un escaneo completo de la tabla lo que hace que cuando se tienen grandes cantidades de datos en la misma, la mejora puede ser muy importante. Con los índices se evita que MySQL tenga que hacer lecturas secuenciales. A diferencia de PostgreSQL este gestor no necesita disponibilidad de grandes recursos de CPU ni de memoria para ser utilizado con eficacia y tiene mejores utilidades de administración (Hispano, 2008) (Gilfillan, 2003)

MySQL si cae o tiene problemas de hardware, puede recuperar los datos, mediante recuperación automática o desde una copia de seguridad más el log de transacciones, puede combinar varios comandos y aceptarlos todos al mismo tiempo con el comando COMMIT (si autocommit está desactivado). Si falla una actualización, todos los cambios se deshacen. Presenta motores de almacenamiento transaccionales que pueden proporcionar mejor concurrencia para tablas que tienen varias actualizaciones concurrentes con lecturas. (Microsystem, 2008)

Procyon establece como política el uso de MySQL como gestor de base de datos, tal motivo influyó a la selección del mismo, no obstante el sistema manipula la base de datos mediante EJB, lo que libera al sistema de someterse profundamente a un gestor de base de datos específico.

HERRAMIENTAS DE DESARROLLO.

En este epígrafe se fundamentan las herramientas de desarrollo utilizadas para el desarrollo del sistema. Estas fueron seleccionadas de acuerdo con las características del proyecto además de que en la entidad Procyon han sido puestas en prácticas obteniendo resultados favorables y han sido establecidas como política para la configuración del ambiente de desarrollo.

1.8.1 Entorno Integrado de Desarrollo (IDE).

Como IDE se toma Eclipse, el cual se adapta en gran medida al proyecto y cumple con las estrategias de la entidad Procyon.

1.8.1.1 ¿Por qué Eclipse?

Existen potentes entornos de desarrollo de software por los que había que pagar costosas licencias como Visual Studio de Microsoft o Visual Age de IBM. Desde hace un tiempo se tiene la oportunidad de tener una herramienta con potencia similar y de libre distribución: Eclipse. Esta es una plataforma de herramientas universal, IDE abierto y extensible, para cualquier cosa y nada en particular. La característica clave de Eclipse es la extensibilidad. Eclipse es una gran estructura formada por un núcleo y muchos plug-ins que van conformando la funcionalidad final. La forma en que los enchufes interactúan es mediante interfaces o puntos de extensión; así, las nuevas aportaciones se integran sin dificultad ni conflictos. Eclipse maneja los recursos del usuario, organizados en uno o más proyectos. Cada proyecto corresponde a un directorio en el directorio de trabajo de Eclipse, y contienen archivos y carpetas, también hace uso de un sistema de control de versiones para manejar los recursos en un proyecto del usuario y define el proceso necesario para guardar y recuperar de un repositorio. (Puebla, 2006)

1.8.2 Apache Ant.

Es una herramienta gratuita que se utiliza para automatizar tareas mecánicas y repetitivas, normalmente en fase de compilación y construcción. Se basa en archivos de configuración XML y clases Java para la realización de las distintas tareas, es independiente al sistema operativo.

1.8.3 SVN (Subversion).

El uso de esta herramienta constituye una política de la entidad Procyon. Es utilizada para el control de versiones del proyecto. Es una aplicación del lado del servidor de código abierto para el control de versiones. Maneja ficheros y directorios y los cambios introducidos en ellos al paso del tiempo. Esto permite recuperar versiones antiguas de los datos o examinar la historia de cómo cambiaron sus datos. En este sentido, muchos piensan de un sistema de control de versiones como una especie de "máquina del tiempo".

Subversion puede funcionar a través de redes, lo que le permite ser utilizado por personas de los distintos equipos. Se trata de un sistema general que puede ser usado para administrar cualquier conjunto de ficheros, que pueden ser código fuente u otros. (Collins-Sussman, 2007)

1.9 CONCLUSIONES.

En este capítulo se abordaron las principales tecnologías empleadas en la construcción del sistema, se expusieron diferentes SVA, se abordaron diferentes comparaciones o justificaciones del uso de la plataforma y las herramientas de desarrollo a utilizar para construir el software.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 INTRODUCCIÓN.

En el presente capítulo, se hace un análisis del sistema que se pretende desarrollar, donde se presentan sus características, se describen los procesos que serán objetos de automatización y se propone el sistema a implementar donde se especifican los módulos que lo componen.

2.2 OBJETIVOS ESTRATÉGICOS DE PROCYON.

Procyon dedica sus esfuerzos a producir software para la telefonía celular y los diferentes servicios que esta soporta, como son las plataformas de contenidos y SVA. Tiene como objetivo principal insertarse en el mercado mundial y elevar su prestigio en el mismo, por lo que no descuida la actualización tecnológica y los caminos de las tendencias a producir para este mercado. El personal se mantiene en constante superación, lo que permite mejorar las técnicas de desarrollo para obtener un producto final con una calidad superior.

2.3 OBJETO DE AUTOMATIZACIÓN.

Los procesos que serán objeto de automatización son los involucrados en la gestión de la información en el SVA que brinda información sobre la Serie Nacional de Béisbol, estos son:

Recepción de las solicitudes de servicio: proceso en el cual se reciben los mensajes que están solicitando petición de información sobre la Serie Nacional de Béisbol al SVA.

Gestión de las Solicitudes: proceso en el cual se busca y se envía la información adecuada sobre la Serie Nacional de Béisbol.

Actualización de la información: proceso que se actualiza la información de último momento sobre la Serie Nacional de Béisbol.

2.4 PROPUESTA DEL SISTEMA.

En este trabajo se propone la creación de un SVA que se integrará a la plataforma Blueeye, el cual debe ofrecer a sus clientes información actualizada sobre la Serie Nacional de Béisbol, a través de un sitio WAP o mediante el envío de un SMS.

La aplicación está compuesta por tres módulos, a continuación se explica cada uno de estos:

- **AlertBaseball-Core:** Es el encargado de recibir todas las peticiones realizadas al sistema y gestionar las mismas. Se encarga del acceso a la base de datos y formula la respuesta que será enviada, así como determina por que vía hacerlo.
- **AlertBaseball-WAP:** Es el encargado de desplegar en un sitio WAP la información que se le desea presentar al cliente.
- **AlertBaseball-WEB:** Se encarga de la administración del sistema, aquí se gestiona la información que se le brinda a los clientes.

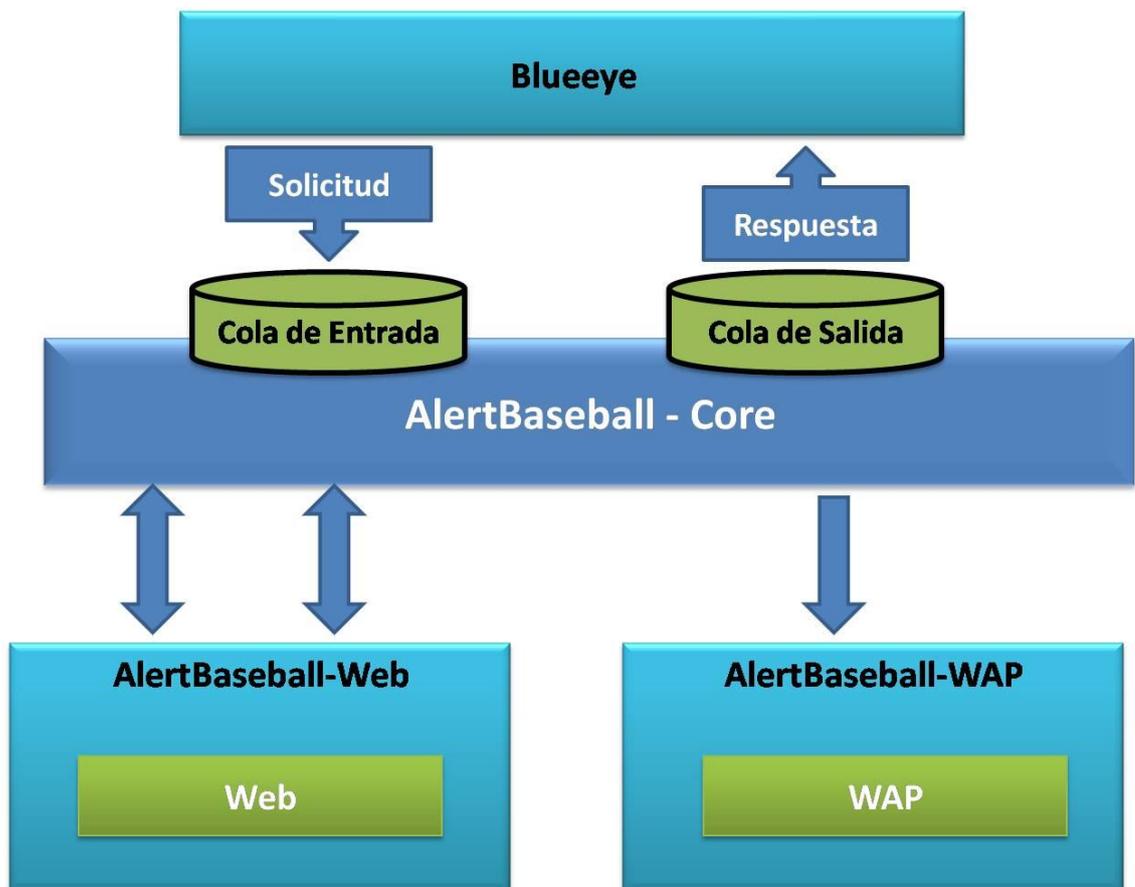


Figura 6 Descripción del sistema

2.4.1 Flujo de procesos del sistema propuesto.

Cuando la plataforma Blueeye recibe un SMS de petición de servicio, que además contiene la palabra reservada correspondiente a nuestro SVA, le envía la petición al sistema. Aquí se procesa la petición de acuerdo con los parámetros enviados por el usuario. El sistema inicialmente toma el nombre del equipo del que se solicita el juego correspondiente, si existe el mismo, se tramita la información requerida y se conforma la respuesta para ser enviada. El sistema toma la palabra clave que representa el modo de respuesta, en caso de que la solicitud de servicio exija una respuesta por medio de un sitio WAP, el sistema genera un código para conformar una URL (del inglés *Uniform Resource Locators*) única y determina el tiempo de expiración del sitio WAP. La URL es enviada al cliente, cuando este acepta el mensaje accede a un Servlet que se encuentra a la escucha y envía la información que corresponde a la URL que se muestra posteriormente en el navegador del dispositivo móvil. En caso de que la solicitud de respuesta es mediante un SMS, se conforma un resumen que se incluye en el SMS de respuesta y se envía. Cuando algún elemento de la solicitud no cumple el estándar establecido o si la información solicitada no puede ser resuelta, el sistema envía un SMS de respuesta con el texto "No results".

Cuando se desea gestionar la información del sistema, el administrador hace peticiones al sitio de administración de la información de la aplicación, se autentica y procede a realizar las operaciones necesarias para mantener actualizada la información que ofrece el servicio.

2.5 ANÁLISIS COMPARATIVO DE OTRAS SOLUCIONES EXISTENTES CON LA PROPUESTA.

Durante la fase inicial de desarrollo del sistema, se hizo un análisis de los diferentes servicios de envío de información a través de solicitudes vía SMS con el objetivo de seleccionar las características ventajosas para la creación del sistema. Se hizo un análisis profundo en dos proveedores de servicios de gran prestigio, la compañía TEMPOS 21 y La Tercera. Estas compañías resaltan características diferentes en cuanto al envío de la información. Por parte de TEMPOS 21, este envía a sus clientes las respuestas de servicio a través de SMS WAP Push, lo que puede provocar que exista imposibilidad de recibir el servicio para aquellos clientes que no soporten en sus teléfonos el protocolo de respuesta. La Tercera ofrece su servicio de información mediante el intercambio de SMS, esto puede resultar muy compatible, pero también puede no alcanzar las expectativas de muchos clientes que tengan posibilidades tecnológicas en sus dispositivos móviles y desean obtener una respuesta más elegante que contenga imágenes o animaciones. En el sistema que se pretende desarrollar, se gestiona la posibilidad de que los clientes sean libres de seleccionar el tipo de respuesta para el servicio,

decidiendo por su gusto o por las limitaciones tecnológicas que pueda presentar. De aquí también el servicio se ve enriquecido por un número mayor de clientes de los que pudiera tener de no dar esta posibilidad que mencionamos.

Muchos servicios de envío de información vía SMS constituyen un puente transaccional entre el cliente y un proveedor de la información, en muchas ocasiones es información con objetivos comerciales o propagandísticos. Esto constituye una limitante y dependencia del servicio a las diferentes empresas que tengan el interés de usar el mismo. El sistema que se pretende implementar presenta una administración Web que posibilitará gestionar la información que se le brindará a los clientes, lo que posibilitará al servicio mostrar información a sus intereses.

2.6 CONCLUSIONES.

En este capítulo se hizo una descripción del sistema, se describieron los flujos de procesos del campo de acción, fueron detallados los módulos que componen el sistema así como la comunicación que se establece entre ellos y se presentaron los flujos del sistema propuestos con sus descripciones.

CAPÍTULO 3: EXPLORACIÓN Y PLANIFICACIÓN

3.1 INTRODUCCIÓN.

En este capítulo se crea una visión refinada de qué debe hacer el sistema que se pretende desarrollar a través de las técnicas intrínsecas de la metodología de desarrollo seleccionada. También se hacen diferentes estimaciones que garantizan una correcta planificación del software y se generan los artefactos involucrados en estas fases

3.2 EXPLORACIÓN.

En la exploración se hace un descubrimiento de la visión y la misión del sistema que se va a desarrollar. Aquí los clientes y desarrolladores trabajan juntos en la elaboración de una declaración de lo que se desea a través de las historias de usuario. Clientes y desarrolladores trabajan juntos en esta fase en el estudio de opciones técnicas, la definición de requisitos y las posibilidades de la arquitectura presentando un prototipo. (Baird, 2003)

3.2.1 Historias de usuario.

Las historias de los usuarios son los principales artefactos para el desarrollo de equipos de proyectos XP. Una Historia de Usuario presenta un alto nivel de definición de un requisito que contiene suficiente información para que los desarrolladores puedan producir una estimación razonable de los esfuerzos para ponerlo en práctica. Una buena forma de pensar sobre una Historia de Usuario es que se trata de un recordatorio para mantener una conversación con su cliente (en XP partes interesadas en el proyecto se denominan clientes). Las historias de usuario son pequeñas, mucho más pequeñas que los casos de uso. En XP una Historia de Usuario debe poder ser implementada por dos personas en una sola iteración. (W.Ambler, 2004)

Tabla 1 Historia de Usuario Gestionar Solicitud de Servicio.

Historia de Usuario	
Número: 1	Nombre: Gestionar Solicitud De Servicio
Usuario: Blueeye	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 3	Iteración Asignada: 1
<p>Descripción: La plataforma Blueeye le envía el texto del SMS de solicitud de servicio al sistema, el sistema recepciona la petición en su cola de entrada, gestiona el mensaje en dependencia de la petición, genera un SMS con la información solicitada o un SMS WAP Push que contiene la dirección del sitio WAP donde se encuentra la información y lo envía a Blueeye para que sea enviado posteriormente al cliente.</p>	
<p>Observación: El SMS de petición lleva una estructura estricta con palabras clave definidas en el sistema.</p>	

Tabla 2 Historia de Usuario Gestionar Estadio.

Historia de Usuario	
Número: 2	Nombre: Gestionar Estadio
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Puntos Estimados: 2	Iteración Asignada: 2
<p>Descripción: El administrador inserta, modifica y elimina la información de un estadio en la base de datos.</p>	
<p>Observación: Sólo se podrán eliminar los estadios en los que no se haya jugado.</p>	

Tabla 3 Historia de Usuario Gestionar Equipo.

Historia de Usuario	
Número: 3	Nombre: Gestionar equipo
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Puntos Estimados: 1	Iteración Asignada: 2
Descripción: El administrador inserta, modifica y elimina la información de un equipo en la base de datos.	
Observación: Sólo se podrán eliminar equipos que no hayan efectuado ningún juego.	

Tabla 4 Historia de Usuario Gestionar Juego.

Historia de Usuario	
Número: 4	Nombre: Gestionar juego
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Puntos Estimados: 3.5	Iteración Asignada: 3
Descripción: El administrador crea, modifica y finaliza los juegos.	
Observación: No se podrán seleccionar equipos que se encuentren jugando ni estadios que se encuentren en uso.	

Tabla 5 Historia de Usuario Gestionar Respuesta WAP.

Historia de Usuario	
Número: 5	Nombre: Gestionar respuesta WAP
Usuario: Cliente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Iteración Asignada: 1
Descripción: Se crea en la base de datos un registro con el resumen de la información a mostrar en el sitio WAP. El sistema gestiona una URL única para cada usuario.	
Observación: Las URL tienen un tiempo de expiración que es determinado por el sistema.	

Tabla 6 Historia de Usuario Administrar Usuario

Historia de Usuario	
Número: 6	Nombre: Administrar Usuario
Usuario: Administrador	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Baja
Puntos Estimados: 0.5	Iteración Asignada: 3
Descripción: Se modifica la contraseña del administrador.	
Observación: Al menos debe existir un usuario.	

3.3 PLANIFICACIÓN.

En esta fase se crea un plan de lanzamiento que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se utiliza como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto. Se hacen pequeños lanzamientos con frecuencia. El desarrollo se divide en iteraciones, cada una de las cuales comienza con un plan de iteración para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.

3.3.1 Estimación de esfuerzo por historias de usuario.

A continuación se asigna una estimación del esfuerzo que se necesita para implementar cada una de las historias de usuario que fueron identificadas, esto se puede apreciar en la tabla que a continuación se muestra:

Tabla 7 Estimación de Esfuerzo por historias de usuario.

Historia de Usuario	Puntos estimados
Gestionar Solicitud De Servicio	3
Gestionar Estadio	2
Gestionar Equipo	1
Gestionar Juego	3.5
Mostrar Sitio WAP	1
Administrar Usuario	0.5

3.3.2 Plan de Iteraciones.

El plan de iteraciones es un elemento clave que agiliza el proceso de desarrollo. Las iteraciones deberán tener un tiempo de duración aproximadamente de 1-3 semanas y este tiempo de planificación permanece constante. Esta característica proporciona una previsibilidad para el cliente, gerentes, desarrolladores y probadores de la misma manera. El uso de iteraciones cortas también tiene en cuenta los cambios en un sistema de requisitos, a menos que una modificación represente un cambio de paradigma, siempre es programado para una iteración más tarde sobre la base de su prioridad desde el punto de vista del cliente. En la planificación, al tener iteraciones más cortas, el cambio, es a lo sumo, a un par de semanas de ser implementado (Mayford).

Para la realización de este sistema se definen 3 iteraciones las cuales se detallan a continuación.

3.3.2.1 Primera Iteración.

En esta iteración se implementarán las historias de usuario que tienen mayor prioridad, donde se obtendrá un modelo que representa la arquitectura del sistema. Esto representa una línea base para el desarrollo de las demás iteraciones. En esta iteración se desarrollan funcionalidades sobre **AlertBaseball-Core** y **AlertBaseball-WAP**.

3.3.2.2 Segunda Iteración.

En esta iteración se implementan varias funcionalidades sobre el módulo **AlertBaseball-WEB**, se desarrolla la gestión de la información de los equipos y los estadios. Esto garantiza la persistencia de la información de los equipos y estadio, que es necesaria para la siguiente iteración.

3.3.2.3 Tercera Iteración.

En esta iteración se completan las funcionalidades sobre **AlertBaseball-WEB**, se gestionan los juegos. Para esto se implementará la gestión de la información de los mismos, así como la gestión de las tablas de puntuación. También se implementa la administración de usuarios.

3.4 PLAN DE DURACIÓN DE LAS ITERACIONES.

A continuación en la tabla 8 se muestra el plan de duración de las iteraciones, en este plan queda detallado en que iteración serán implementadas las historias de usuario y tiempo de duración de cada iteración para cada equipo de desarrollo, en este caso existe uno solo.

Tabla 8 Plan de Duración de las Iteraciones.

Iteración	Orden de las Historias de Usuario a implementar	Duración Total de la Iteración
Iteración 1	Gestionar Solicitud De Servicio Mostrar Sitio WAP	4 Semanas
Iteración 2	Gestionar Equipo Gestionar Estadio	3 Semanas
Iteración 3	Gestionar Juego Administrar usuario	4 Semanas

3.5 PLAN DE ENTREGA.

El plan de entrega muestra las fechas reales de entrega de los módulos en sus diferentes versiones.

Tabla 9 Plan de Entrega.

Módulo	2da semana de Abril	1ra semana de Mayo	1ra semana de Junio
AlertBaseball-Core	0.1	0.2	0.3
AlertBaseball-WAP	0.1	0.1	0.2
AlertBaseball-WEB		0.1	0.2

3.6 CONCLUSIONES.

En este capítulo se abordó la fase de exploración y dentro de esta se hizo una introducción a las historias de usuario las cuales juegan un papel fundamental. A través de estas se hizo una descripción de la necesidad funcional del software. Se describió la fase de planificación, donde se especificaron cada una de las iteraciones y sus objetivos fundamentales y se creó además el plan de entrega de las diferentes versiones por las que transitará el sistema.

CAPÍTULO 4: DISEÑO, IMPLEMENTACIÓN Y PRUEBA

4.1 INTRODUCCIÓN

En este capítulo se detallan las fases de diseño, implementación y prueba. Aquí se describen cada una de las iteraciones realizadas, donde quedan especificadas las tareas de programación que se desarrollan en cada una de estas. También se expondrán las diferentes pruebas de aceptación que fueron realizadas al sistema.

4.2 DISEÑO

El diseño debe brindar sólo las funcionalidades que reflejen el interés de implementación hasta un momento determinado del proyecto. Debe mostrar una actitud de constante extensión y perfeccionamiento. Se debe ejecutar grandes decisiones lo más tarde como sea posible en el proceso de desarrollo. Al aplazar el costo de tomar las decisiones se tiene la mayor oportunidad posible de tener razón. (James Newkirk, 2002)

El diseño es realizado por los propios desarrolladores. En ocasiones se reúnen aquellos con más experiencia o incluso se involucra al cliente para diseñar las partes más complejas. En estas reuniones se emplean un tipo de tarjetas denominadas CRC (Clases-Responsabilidades-Colaboración) cuyo objetivo es facilitar la comunicación y documentar los resultados. Para cada clase identificada se rellenará una tarjeta de este tipo y se especificará su finalidad así como otras clases con las que interactúe. Las tarjetas CRC son una buena forma de cambiar de la programación estructurada a una filosofía orientada a objetos. (Zarzuela, 2002)

Los diagramas creados durante el desarrollo del sistema de información sobre la Serie Nacional de Béisbol se encuentran en el Anexo 1.

4.2.1 Patrones de Diseño.

Los patrones del diseño tratan los problemas del diseño que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Por lo tanto, los patrones de diseño son soluciones exitosas a problemas comunes, constituyen una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. Existen muchas formas de implementar patrones de diseño. Los detalles de las implementaciones son llamadas estrategias. (Ciberaula, 2006)

4.2.1.1 Modelo Vista Controlador.

El patrón de diseño Modelo Vista Controlador (del inglés Model-View-Controles o MVC) describe una forma, muy utilizada en la Web, de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. (ProgramaciónWeb.Net, 2008)

Elementos del patrón:

Modelo: El modelo es la estructura de datos que representa el contenido de la aplicación.

Vista: Muestra la información del modelo al usuario y al necesitar poco o nada de código, puede ser desarrollada por un equipo de diseñadores independiente de los programadores.

Controlador: Gestiona las entradas del usuario, recoge las órdenes y datos que el usuario envía desde la vista, las traduce en operaciones del modelo de negocio y, en base a los resultados, muestra los resultados en uno u otro formulario de la vista.

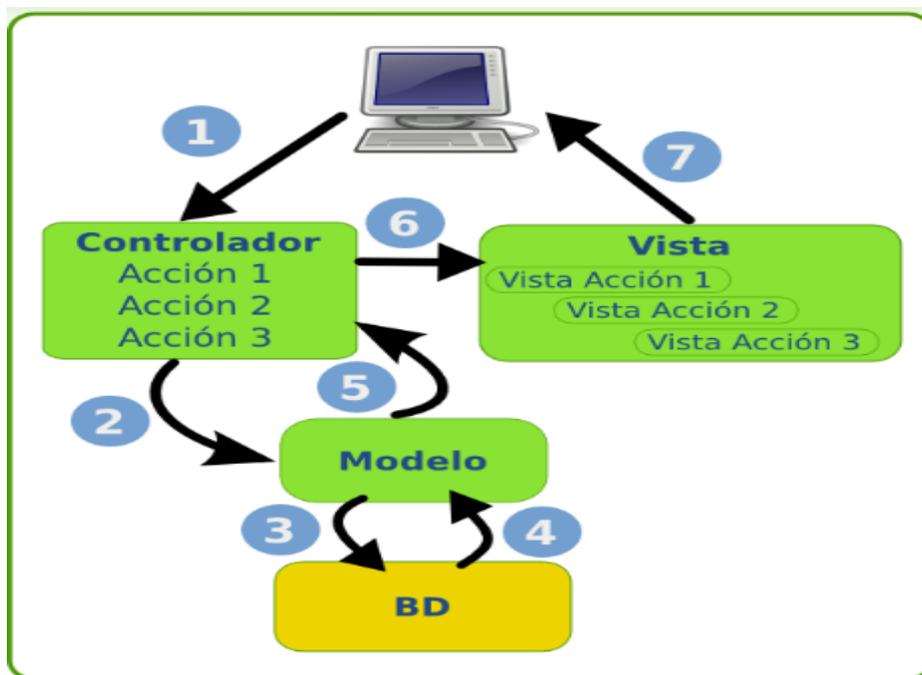


Figura 7 Modelo Vista Controlador.

4.2.1.2 Fábrica Abstracta.

Fábrica abstracta (del inglés *Abstract Factory*), permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando.

Estructuras típicas del patrón fábrica abstracta:

- La definición de interfaces para la familia de productos genéricos.
- Implementación de las interfaces de los productos para cada una de las distintas familias concretas.
- La definición de los métodos de creación de los productos genéricos en la interfaz de la fábrica.
- Implementación de una fábrica para cada una de las familias concretas.

4.2.1.3 Adaptador.

Se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pudiera utilizar la primera, haga uso de ella a través de la segunda. Convierte la interfaz de una clase en otra interfaz que el cliente espera. Adaptador permite a las clases trabajar juntas, de otra manera no podrían hacerlo debido a sus interfaces incompatibles.

4.2.1.4 Iterador.

Define una interfaz que declara los métodos necesarios para acceder secuencialmente a un grupo de objetos de una colección. Algunos de los métodos que podemos definir en la interfaz Iterador son:

- Primero ().
- Siguiente ().
- Hay_Mas ().
- Elemento_Actual ().

4.3 IMPLEMENTACIÓN.

En esta fase se procede a implementar las historias de usuario designadas en las diferentes iteraciones, cada Historia de Usuario se descompone en tareas de programación que son asignadas a un equipo de programación responsable de la realización de la misma. (Beck, 1999)

4.3.1 Primera Iteración.

En esta iteración se implementan las historias de usuario de mayor prioridad del sistema, se cimienta la base de la arquitectura y se obtiene un prototipo funcional del sistema.

Tabla 10 Historias de usuario desarrolladas en la primera iteración.

Historia de Usuario	Estimación	Real
Gestionar Solicitud De Servicio	3	3
Gestionar Respuesta WAP	1	1
TOTAL	4	4

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

Historia de Usuario: **Gestionar Solicitud De Servicio.**

Tabla 11 Tarea 1 de la Historia de Usuario Gestionar Solicitud De Servicio.

Tarea	
Número Tarea: 1	Número de Historia de Usuario: 1
Nombre Tarea: Recepcionar Solicitud	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Fecha de Inicio: 03/03/2008	Fecha de Fin: 06/03/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se implementa la interfaz EntryPoint definida en Blueeye que es la encargada capturar la solicitud y enviarla a la clase encargada de su procesamiento.	

Tabla 12 Tarea 2 de la Historia de Usuario Gestionar Solicitud De Servicio.

Tarea	
Número Tarea: 2	Número de Historia de Usuario: 1
Nombre Tarea: Procesar Solicitud	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.5
Fecha de Inicio: 07/03/2008	Fecha de Fin: 19/03/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
<p>Descripción: Se toma el mensaje de entrada con el equipo solicitado, se hace una consulta para ver el último juego. Con el resultado de esta consulta se llena una tabla resumen en la base de datos. Si la solicitud de respuesta es mediante un SMS, se conforma el mensaje con los datos guardados en la tabla resumen, si la solicitud de respuesta es mediante un sitio WAP, se elabora la respuesta mediante un SMS WAP Push.</p>	

Tabla 13 Tarea 3 de la Historia de Usuario Gestionar Solicitud De Servicio.

Tarea	
Número Tarea: 3	Número de Historia de Usuario: 1
Nombre Tarea: Elaborar Respuesta WAP Push	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 20/03/2008	Fecha de Fin: 24/03/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
<p>Descripción: Se toma el Id del campo a visualizar de la tabla que contiene los resúmenes. Se genera una clave única mediante la cual se va a identificar el sitio WAP a visualizar. Se le adiciona esta clave única a la URL para que en cada consulta que se haga se de como respuesta un sitio WAP diferente. Se establece la fecha de publicacion del sitio WAP y la fecha de expiración de dicho sitio. Se guardan los datos de la respuesta WAP en la base de datos en cuestión.</p>	

Tabla 14 Tarea 4 de la Historia de Usuario Gestionar Solicitud De Servicio.

Tarea	
Número Tarea: 4	Número de Historia de Usuario: 1
Nombre Tarea: Enviar Mensaje de Respuesta a Blueeye	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha de Inicio: 25/03/2008	Fecha de Fin: 27/03/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se toma la ruta de envío del mensaje(HTTP, OpenSMS, SMPP, etc.), se toma la dirección de quien solicitó el mensaje y se establece como destinatario para asegurar que le llegue el mensaje a la misma persona que lo solicitó. En caso de que la respuesta sea un SMS se incluye en el texto del SMS la respuesta, si la respuesta es solicitada por SMS WAP Push se incluye la URL del sitio WAP.	

Historia de Usuario: **Mostrar Sitio WAP**

Tabla 15 Tarea 1 de la Historia de Usuario Mostrar Sitio WAP.

Tarea	
Número Tarea: 1	Número de Historia de Usuario: 6
Nombre Tarea: Mostrar Sitio WAP	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de Inicio: 28/03/2008	Fecha de Fin: 04/04/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Mediante la URL se accede a un Servlet del cual se extrae el Id de la respuesta WAP previamente guardada en la base de datos, se obtiene el resumen del juego guardado en la base de datos, posteriormente se guardan los datos en el objeto HTTPrequest del sitio WAP y finalmente se muestran los datos en el sitio WAP.	

4.3.2 Segunda Iteración.

Durante el desarrollo de esta iteración se implementan funcionalidades sobre el módulo **AlertBaseball-WEB**. Esto posibilita la gestión de la información de los equipos y los estadios, garantizando la persistencia de la información de los equipos y estadio que es necesaria para la implementación de funcionalidades posteriores.

Tabla 16 Historias de usuario desarrolladas en la segunda iteración.

Historia de Usuario	Estimación	Real
Gestionar Estadio	2	2
Gestionar Equipo	2	2
TOTAL	4	4

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

Historia de Usuario: **Gestionar Estadio**

Tabla 17 Tarea 1 de la Historia de Usuario Gestionar Estadio.

Tarea	
Número Tarea: 1	Número de Historia de Usuario: 2
Nombre Tarea: Crear e Implementar Interfaz IDAOSTadium	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.7
Fecha de Inicio: 07/04/2008	Fecha de Fin: 10/04/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se crea la interfaz IDAOSTadium que permite realizar las operaciones de persistencia de los datos.	

Tabla 18 Tarea 2 de la Historia de Usuario Gestionar Estadio.

Tarea	
Número Tarea: 2	Número de Historia de Usuario: 2
Nombre Tarea: Crear e Implementa el Bean de Entidad CStadium	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Fecha de Inicio: 11/04/2008	Fecha de Fin: 14/04/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Crear e implementar el Bean de Entidad CStadium, clase que permite guardar el objeto estadio.	

Tabla 19 Tarea 3 de la Historia de Usuario Gestionar Estadio.

Tarea	
Número Tarea: 3	Número de Historia de Usuario: 2
Nombre Tarea: Crear Interfaces Web para Gestionar Estadio	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.7
Fecha de Inicio: 15/04/2008	Fecha de Fin: 18/04/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se Crea una interfaz Web para crear un estadio, también se crea una interfaz Web que permita modificar los estadios. Se crea una interfaz Web para eliminar estadio. Se implementa un Servlet para acceder de forma dinámica desde el servidor Web a los datos en el servidor de base de datos.	

Historia de Usuario: **Gestionar Equipo**

Tabla 20 Tarea 1 de la Historia de Usuario Gestionar Equipo.

Tarea	
Número Tarea: 1	Número de Historia de Usuario: 3
Nombre Tarea: Crear e Implementar Interfaz IDAOTeam	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.7
Fecha de Inicio: 21/04/2008	Fecha de Fin: 24/04/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se crea la interfaz IDAOTeam que permite realizar las operaciones de persistencia de los datos.	

Tabla 21 Tarea 2 de la Historia de Usuario Gestionar Equipo.

Tarea	
Número Tarea: 2	Número de Historia de Usuario: 3
Nombre Tarea: Crear e Implementa el Bean de Entidad de CTeam	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Fecha de Inicio: 25/04/2008	Fecha de Fin: 28/04/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Crear e implementar el Bean de Entidad CTeam, clase que permite guardar el objeto Equipo.	

Tabla 22 Tarea 3 de la Historia de Usuario Gestionar Equipo.

Tarea	
Número Tarea: 3	Número de Historia de Usuario: 3
Nombre Tarea: Crear Interfaces Web para Gestionar Equipo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.7
Fecha de Inicio: 29/04/2008	Fecha de Fin: 02/05/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se Crea una interfaz Web para crear nuevo equipo, también se crea una interfaz Web que permita modificar los equipos. Se crea una interfaz Web para eliminar equipo. Se implementa un Servlet para acceder de forma dinámica desde el servidor Web a los datos en el servidor de base de datos.	

4.3.3 Tercera Iteración.

Aquí completamos las funcionalidades sobre **AlertBaseball-WEB**, donde se implementa la gestión de los juegos, así como las puntuaciones de los mismos a medida que va transcurriendo.

Tabla 23 Historias de usuario desarrolladas en la tercera iteración.

Historia de Usuario	Estimación	Real
Gestionar Juego	3.5	3.5
Administrar usuario	0.5	0.5
TOTAL	4	4

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

Historia de Usuario: **Gestionar Juego**

Tabla 24 Tarea 1 de la Historia de Usuario Gestionar Juego.

Tarea	
Número Tarea: 1	Número de Historia de Usuario: 4
Nombre Tarea: Crear e Implementar interfaz IDAOGame.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.9
Fecha de Inicio: 05/05/2008	Fecha de Fin: 09/05/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se crea e implementa la interfaz IDAOGame, esto permite la persistencia de los datos en la base de datos.	

Tabla 25 Tarea 2 de la Historia de Usuario Gestionar Juego.

Tarea	
Número Tarea: 2	Número de Historia de Usuario: 4
Nombre Tarea: Crear el Bean de Entidad CGame.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha de Inicio: 12/05/2008	Fecha de Fin: 13/05/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se crea el Bean de Entidad CGame para guardar los datos del objeto juego.	

Tabla 26 Tarea 3 de la Historia de Usuario Gestionar Juego.

Tarea	
Número Tarea: 3	Número de Historia de Usuario: 4
Nombre Tarea: Crear e Implementar la interfaz IDAOTeamScore.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.8
Fecha de Inicio: 14/05/2008	Fecha de Fin: 19/05/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se crea e implementa la interfaz IDAOTeamScore, esto permite la persistencia de las puntuaciones de los equipos en la base de datos.	

Tabla 27 Tarea 4 de la Historia de Usuario Gestionar Juego.

Tarea	
Número Tarea: 4	Número de Historia de Usuario: 4
Nombre Tarea: Crear el Bean de Entidad CTeamScore.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha de Inicio: 20/05/2008	Fecha de Fin: 21/05/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se crea el Bean de Entidad CTeamScore para guardar los datos del objeto puntuación de equipo.	

Tabla 28 Tarea 5 de la Historia de Usuario Gestionar Juego.

Tarea	
Número Tarea: 5	Número de Historia de Usuario: 4
Nombre Tarea: Crear interfaces Web y Servlet.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha de Inicio: 22/05/2008	Fecha de Fin: 27/05/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se crea la interfaz Web "CrateGame.jsp", se crea la interfaz Web "UpdateGame.jsp", se crea un Servlet para garantizar el acceso de forma dinámica entre el servidor Web y la base de datos.	

Historia de Usuario: **Administrar Usuario.**

Tabla 29 Tarea 1 de la Historia de Usuario Administrar Usuario.

Tarea	
Número Tarea: 1	Número de Historia de Usuario: 6
Nombre Tarea: Gestionar usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 28/05/2008	Fecha de Fin: 01/06/2008
Programador Responsable: Marcos Cruz Cué – Jaisniel Pérez Evora	
Descripción: Se modifica la contraseña del administrador..	

4.4 FASE DE PRUEBA.

“Soy bueno engañándome en creer que lo que escribí es lo que quise decir. También soy bueno en creer que lo que quise decir es lo que debería haber significado. Por lo tanto, no confío en nada que escribo hasta que tenga pruebas de ello” (Beck, 1999)

Las pruebas se dividen en dos grupos: pruebas unitarias, encargadas de verificar el código y las pruebas de aceptación, responsables de evaluar el cumplimiento de las funcionalidades requeridas. Las pruebas son un recurso y a la vez una responsabilidad. Escribir sólo una prueba y hacerla correr no declara haber terminado. Se deben escribir todas las pruebas que se pueda imaginar de forma tal que compruebe todas las posibles fallas. (Beck, 1999)

4.4.1 Pruebas de Aceptación.

Las pruebas de aceptación se crean a partir de las historias de usuario. El cliente define las pruebas para verificar si una Historia de Usuario ha sido correctamente implementada. Una Historia de Usuario puede tener una o varias pruebas de aceptación. El cliente es responsable de verificar el pasaje de las pruebas de aceptación y priorizar la corrección de las pruebas fallidas.

Las pruebas de aceptación son pruebas tipo caja negra a nivel del sistema: cada prueba de aceptación corresponde a un resultado producido por el sistema, deben crearse pruebas de aceptación en cada iteración. Si no hay pruebas de aceptación nuevas no se ha hecho nada nuevo. Las pruebas de aceptación deben ser automáticas, correrse frecuentemente, publicarse sus resultados y programarse su corrección para la próxima iteración. Una Historia de Usuario no está completa hasta no haber pasado todas sus pruebas de aceptación. (Barbone, 2004)

4.5 CONCLUSIONES.

En este capítulo abordamos las fases de diseño, implementación y prueba. Se hizo una breve descripción del sistema y sus características esenciales. Se detallaron cada una de las iteraciones en la fase de implementación y se explicaron las pruebas para esta metodología en especial, como las pruebas de aceptación que tienen una vital importancia para medir la calidad del producto final.

CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

5.1 INTRODUCCIÓN

En el presente capítulo se procede con el estudio de factibilidad, el cual tiene como objetivo conocer la viabilidad de implementar el proyecto, donde se estima la planificación, tamaño y esfuerzo que necesita el mismo para su creación.

5.2 CARACTERÍSTICAS DEL PROYECTO

Tabla 30 Entradas Externas (EI)

Nombre de la Entrada Externa	Cantida de Ficheros	Cantidad de Elementos de Datos	Clasificación (simple,media y compleja)
Adicionar un equipo	1	4	simple
Actualizar un equipo	1	4	simple
Eliminar un equipo	1	1	simple
Adicionar un estadio	1	1	simple
Actualizar un estadio	1	1	simple
Eliminar un estadio	1	1	simple
Iniciar juego	2	12	medio
Actualizar juego	2	9	medio
Modificar contraseña	1	2	simple
Total		35	

Tabla 31 Salidas Externas(SE)

Nombre de la Salida Externa	Cantida de Ficheros	Cantidad de Elementos de Datos	Clasificación (simple,media y compleja)
Total		0	

Tabla 32 Consultas Externas (EQ)

Nombre de la Consulta Externa	Cantida de Ficheros	Cantidad de Elementos de Datos	Clasificación (simple,media y compleja)
Solicitar Servicio	2	3	simple
Total		3	

Tabla 323 Peticiones

Nombre de la Petición	Cantida de Ficheros	Cantidad de Elementos de Datos	Clasificación (simple,media y compleja)
Total		0	

Tabla 334 Ficheros internos (LIF)

Nombre del Fichero Interno	Cantida de Ficheros	Cantidad de Elementos de Datos	Clasificación (simple,media y compleja)
tbgame	1	8	simple
tbgameresume	1	18	simple
tbestadium	1	2	simple
tbteam	1	5	simple
tbteamscore	1	6	simple
tbaseballsecurity	1	3	simple
tbWAPresponse	1	5	simple
Total		47	

Table 345 Interfaces Externas

Nombre de la Interfaz Externa	Cantida de Ficheros	Cantidad de Elementos de Datos	Clasificación (simple,media y compleja)
Total		0	

Tabla 356 Puntos de función desajustados.

Elementos	Simple		Medio		Complejo		Subtotal
	No.	Peso	No.	Peso	No.	Peso	
Entradas externas(EI)	14	3	21	4	0	6	126
Salidas externas(EO)	0	4	0	5	0	7	0
Consulta externas(EQ)	3	3	0	4	0	6	9
Ficheros internos(LIF)	47	7	0	10	0	15	329
Interfaces internas(EIF)	0	5	0	7	0	10	0
Total							455

5.2.1 Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

El modelo COCOMO II usa Puntos Función 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 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. (Adriana Gómez, 1999)

Tabla 367 Características

Características	Valor
Puntos de función desajustados	455
Lenguaje(JAVA)	53
Instrucciones fuentes por punto de función	24115
Instrucciones fuentes	24.1 KSLOC

5.3 ESTIMACIÓN

Tabla 378 Factores de Escala.

Nombre	Valor	Justificación
PREC	1.24	Existen varios sistemas con igual similitud en muchos países, pero no en Cuba.
FLEX	1.01	La flexibilidad de desarrollo del producto es muy alta.
RESL	1.41	Existen muy pocos riegos.
TEAM	1.10	El Equipo presenta una alta cohesión.
PMAT	1.56	Se tienen conocimientos de como se desarrollan estos sistemas.
Total(SF)	6.32	

Tabla 389 Multiplicadores de Esfuerzo.

Nombre	Valor	Justificación
RCPX	0.83	El sistema presenta complejidad media.
RUSE	1.07	Existe gran reutilización de código en el producto.
FCIL	1.00	Se utilizan herramientas para el modelado y se utiliza un IDE, existe un soporte multilugar débil.
PERS	0.83	El personal tiene alta capacitación.
PREX	0.87	Vasta experiencia de desarrollo con la plataforma y lenguaje.
PDIF	1.00	La plataforma es estable y se hace un uso normal de la memoria y el almacenamiento.
SCED	1.14	No existen problemas con el cronograma de desarrollo, es asequible.
Total(EM)	0.73	

Cálculos

A(Constante)=2.45

B(Factor escala)=0.91 * 0.1 * SF = 0.91 * 0.1 * 6.32 = 0.58

C = 3.67

$$D = 0.24$$

Esfuerzo nominal (PM nominal)

$$PM \text{ nominal} = A * (SIZE) ^ B$$

$$PM \text{ nominal} = 2.45 * (24.1) ^ 0.58 = 2.45 * 6.33 = 15.5$$

Esfuerzo (PM)

$$PM = PM \text{ nominal} * \pi EM$$

$$PM = 15.5 * 0.73 = 11.3$$

Tiempo Calendario del Desarrollo (TDEV)

$$TDEV = C * PM (\text{esfuerzo}) ^ F$$

$$F = D + 0.2 * 0.01 * \sum SF = 0.24 + 0.2 * 0.01 * 6.32 = 0.24 + 0.01$$

$$F = 0.25$$

$$TDEV = 3.67 * 11.3 ^ 0.25 = 3.67 * 1.83$$

$$TDEV \approx 6.7$$

Costo Humano

$$CH = PM/TDEV = 11.3/6.7 \approx 1.69$$

$$C = CH * Salario * PM = 1.69 * 100 * 11.3$$

$$C = \$1910$$

Resultados

Table 40 Resultados.

Cálculo de:	Valor
Esfuerzo	11.3 hombres/mes
Tiempo de desarrollo	8 meses
Cantidad de hombres	2 hombres
Salario medio	\$ 100
Costo	\$ 1910

5.4 BENEFICIOS TANGIBLES E INTANGIBLES

Este sistema de envío de información sobre la Serie Nacional de Béisbol es un SVA que será integrado a la plataforma Blueeye para componer un producto con fines comerciales. Este SVA constituye una importante fuente de información de la pasión cubana (el béisbol), así como contribuye a aumentar el valor de la plataforma para su posterior comercialización en el plano nacional e internacional.

El sistema constituye una base para la creación de los SVA que gestionan información por solicitud del cliente en la entidad de Procyon, también aumenta el prestigio de la misma tanto en el ámbito nacional como internacional.

5.5 ANÁLISIS DEL COSTO

Todo producto comercial tiene como objetivo final alcanzar el éxito financiero, para esto debe existir una relación favorable entre costo/beneficio. La implementación de nuestro sistema no conllevó a la realización de grandes gastos debido a que todas las herramientas utilizadas son libres de licencias de uso y comercialización, sólo influyen en el costo del sistema el salarios de los desarrolladores.

5.6 CONCLUSIONES

En este capítulo se hizo un estudio de la factibilidad del sistema propuesto, se realizaron las estimaciones pertinentes de planificación, tamaño y esfuerzo necesario para su implementación y se determinó que es viable llevar a cabo el desarrollo de mismo a través de la relación costo-beneficio que se establece en su culminación y puesta en práctica en la plataforma Blueeye.

CONCLUSIONES GENERALES

Con la realización de este trabajo se profundizó en el conocimiento de las tecnologías inalámbricas y los servicios que se prestan a través de estas, haciendo énfasis en los SVA. Se hizo un estudio de los mismos a escala internacional para la creación de un sistema enriquecido por la comparación de varios similares existentes en la actualidad. La metodología utilizada para el proceso de desarrollo del software fue XP, que ajusta sus principales prácticas de desarrollo a las condiciones y se utiliza la plataforma JAVA EE, donde todas las herramientas utilizadas fueron libres de costo por licencia. Se comprobó la viabilidad del proyecto y se procede a su puesta en marcha. Se cumplieron satisfactoriamente con los objetivos trazados y las expectativas del cliente.

A continuación se exponen los resultados obtenidos:

- Se implementó un SVA de envío de información sobre la Serie Nacional de Béisbol para ser incorporado a la plataforma Blueeye.
- Se implementó un sitio de administración Web para gestionar la información que es brindada por el servicio.
- Se implementó la gestión del sitio WAP, necesaria para presentar al cliente la información solicitada.

RECOMENDACIONES

- Utilizar la información acumulada en la base de datos para generar reportes y estadísticas.
- Reutilizar el servicio para implementar otros de características similares y enriquecer entonces la plataforma Blueeye.
- Poner en práctica el sistema para que sea ejecutado en condiciones reales.

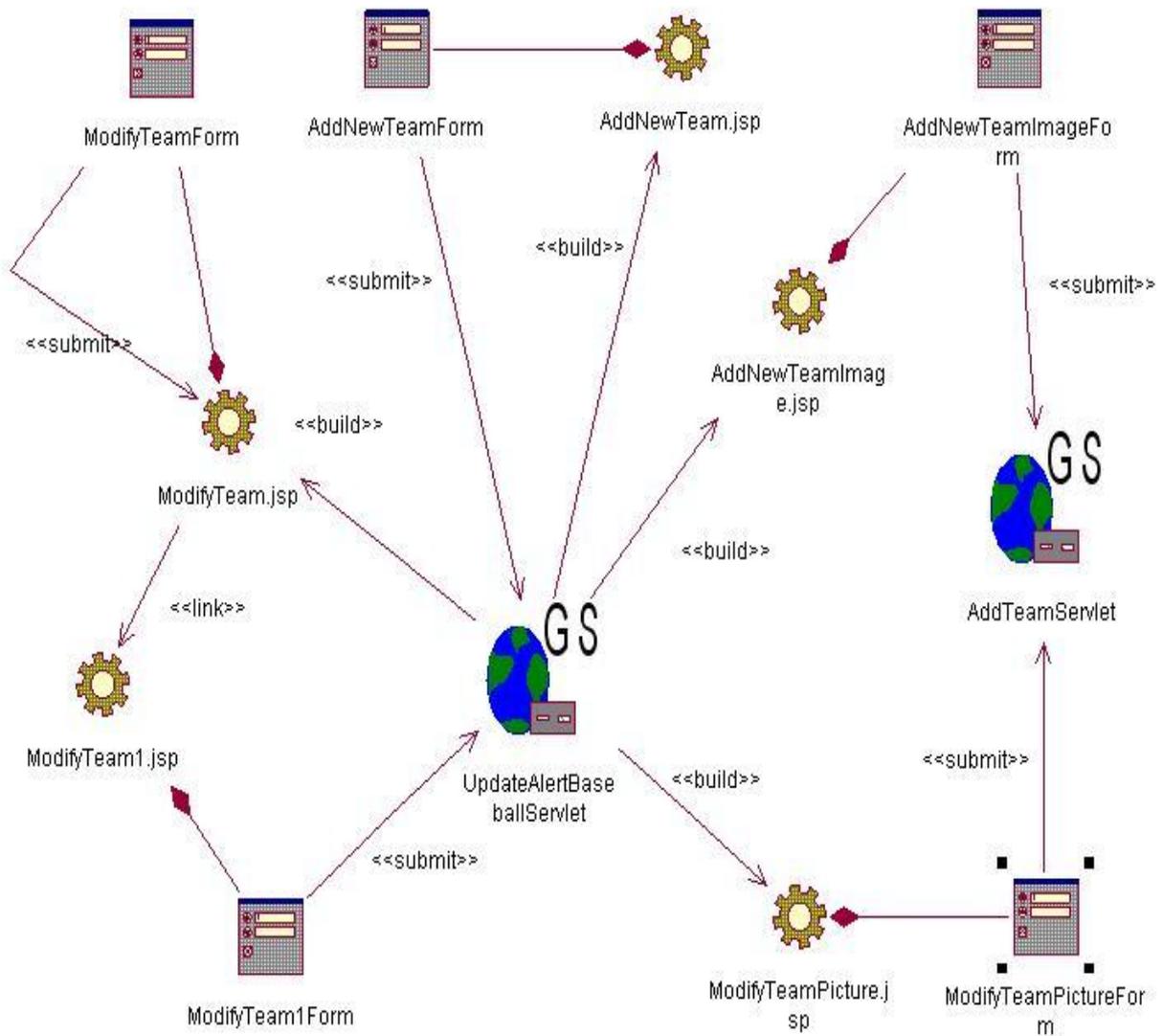
BIBLIOGRAFÍA

- Baird, Stewart. 2003.** *Sams Teach Yourself Extreme Programming in 24 Hours.* 2003.
- Beck, Kent. 1999.** *Extreme programming explained.* s.l. : Addison-Wesley, 1999.
— . **2002.** *Una explicación de la programación extrema: aceptar el cambio.* s.l. : Addison-Wesley Iberoamericana Espanya, S.A, 2002.
- Castillo, Xavier. 2006.** *Aplicaciones Móviles y su impacto en las cuentas de resultados.* 2006.
- Ciberaula. 2006.** <http://java.ciberaula.com>. [Online] 06 05, 2006. [Cited: 06 05, 2008.]
http://java.ciberaula.com/articulo/diseno_patrones_j2ee/.
- Collins-Sussman, Ben. 2007.** *Version Control with Subversion.* 2007.
- Corporation, Borland Software. 2003.** *Desarrollo de aplicaciones.* 2003.
- Daniel Meyer #97-29843, Efrain Pinto #97-30010, Friedrich Gonzalez #97-29608, Miguel Ojea #97-29925.** [Online] [Cited: Diciembre 05, 2007.]
<http://www ldc.usb.ve/~figueira/cursos/redes2/em01/EXPO-em01/WAP/#capas>.
entumovil.net. [Online] [Cited: Diciembre 05, 2007.] <http://entumovil.net/que-es-wap-push.php>.
- Gilfillan, Lan. 2003.** *MySQL.* 2003.
- Hall, Marty. 08.** www.programacion.net. [Online] 01 9, 08. [Cited: 01 9, 08.]
http://www.programacion.net/java/tutorial/servlets_jsp/1/.
- Hanson, Jeff.** [Online] [Cited: Diciembre 05, 2007.]
http://www.programacion.com/java/articulo/jap_persis_ejb/#0_intro.
- Adriana Gómez, María del C.López,. 1999.** *Cocoma II.* 1999.
- Hispano, MySQL. 2008.** MySQL Hispano. [Online] 01 10, 2008. [Cited: 01 10, 2008.]
<http://www.mysql-hispano.org>.
- Ing. Scándolo, Carlos Iván.** *Desarrollo Wap: BBSMóvil.* Córdoba,Argentina : s.n.
- James Newkirk, Robert C. Martin. 2002.** *La Programación Extrema en la práctica.* Iberoamericana Espanya, S.A. : Addison-Wesley, 2002.
- Mayford.** Tecnologías Mayford. [Online] [Cited: 04 11, 08.]
http://www.mayford.ca/xp/iteration_plan.html.
- Microsystem, Sun. 2008.** <http://dev.mysql.com>. [Online] 06 01, 2008. [Cited: 06 01, 2008.]
<http://dev.mysql.com/doc/refman/5.0/es/storage-engines.html>.

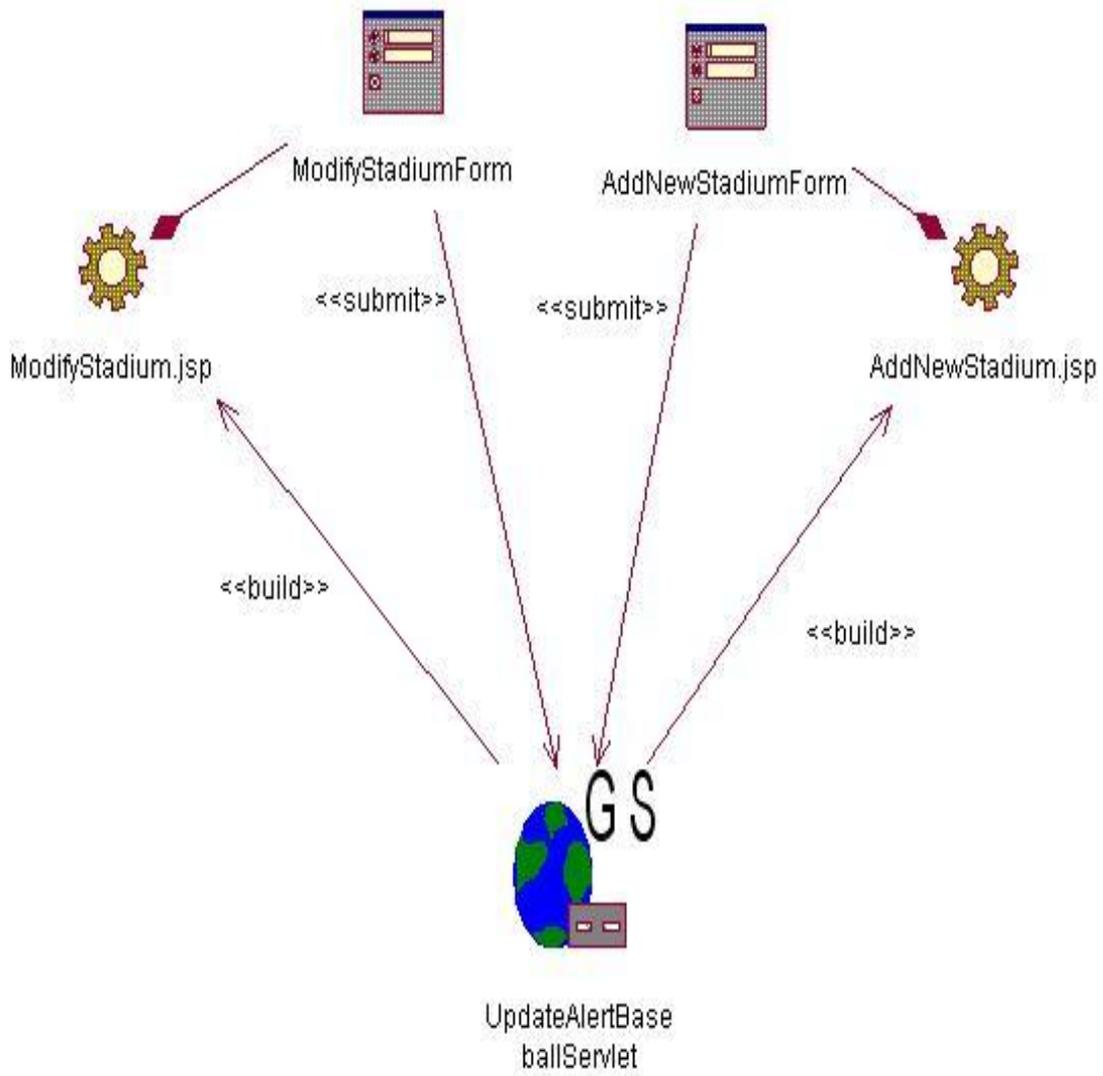
- Molpeceres, Alberto. 2002.** www.ibm.com. [Online] 12 15, 2002. [Cited: 05 15, 2008.]
<http://www.willydev.net/InsiteCreation/v1.0/Descargas/articulos/general/cualxpfdrup.pdf>.
- Passani, Luca. 08.** <http://wurfl.sourceforge.net>. [Online] enero 8, 08. [Cited: enero 8, 2008.]
<http://wurfl.sourceforge.net/java/wall.php>.
- Patel, Rima, Brose, Gerald y Silverman, Micah. 2006.** *Mastering Enterprise JavaBeans 3.0*. Indianapolis : Wiley, 2006. 0-321-49029-0.
- ProgramaciónWeb.Net. 2008.** <http://www.programacionweb.net>. [Online] 06 07, 2008. [Cited: 06 07, 2008.] <http://www.programacionweb.net/articulos/articulo/?num=505>.
- Puebla, Iván García. 2006.** [Online] 07 18, 2006. [Cited: 01 18, 2008.]
http://www.gui.uva.es/~laertes/mambo/index2.php?option=com_content&do_pdf=1&id=56.
- Red Hat, Inc. 2007.** www.redhat.es/jboss. [Online] 2007. [Cited: 01 20, 2008.]
- Sun.** [Online] [Cited: Diciembre 05, 2007.] <http://www.programacion.com/java/tutorial/jaxb/1/>.
- W.Ambler, Scott. 2004.** *The Object Primer*. 2004.
- Zarzuela, Gregorio Robles Martínez y Jorge Ferrer. 2002.** *Gregorio Robles Martínez y Jorge Ferrer Zarzuela*. 2002.

ANEXOS

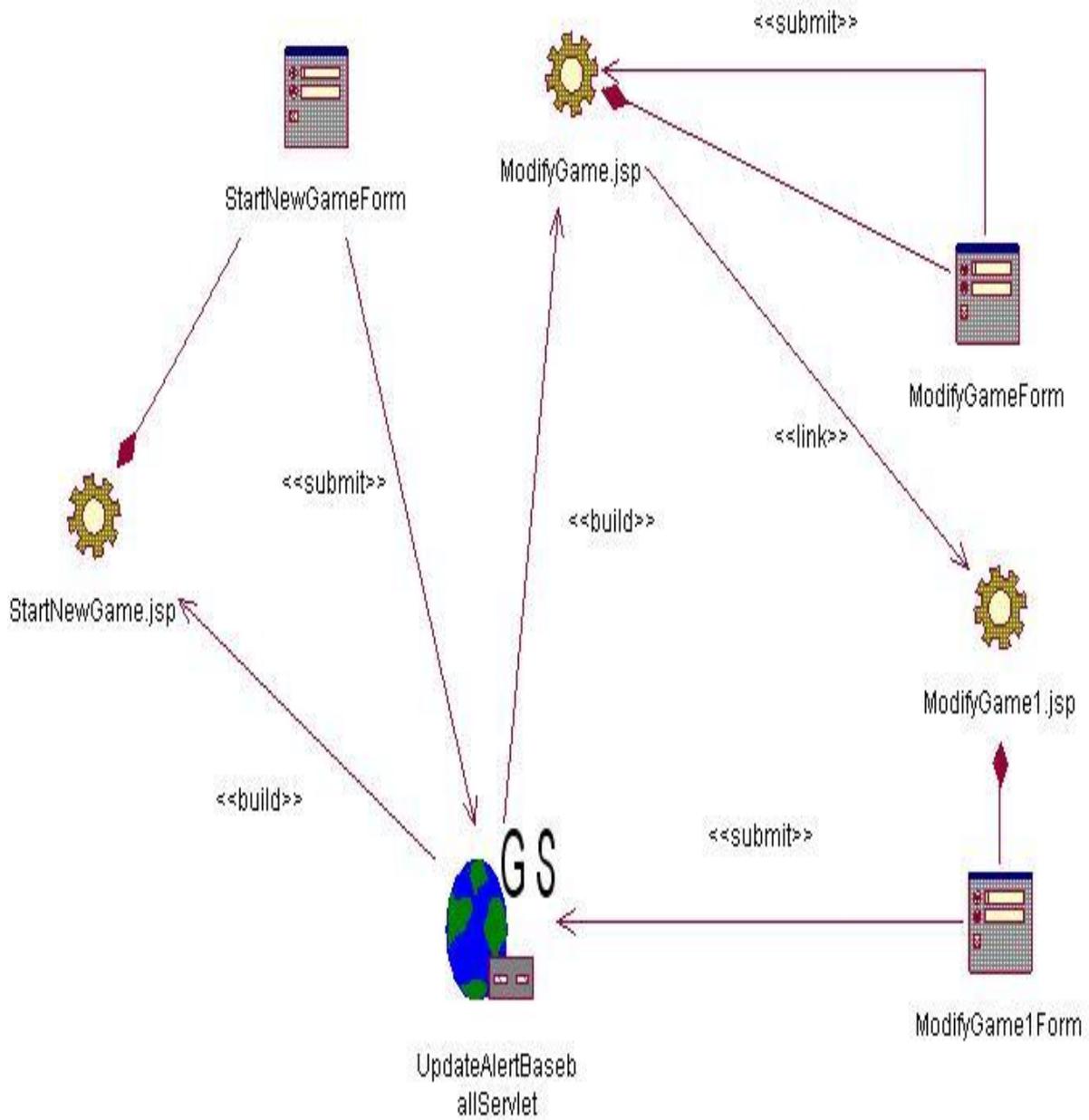
Anexo 1. Diagrama de Clases Del Subsistema.



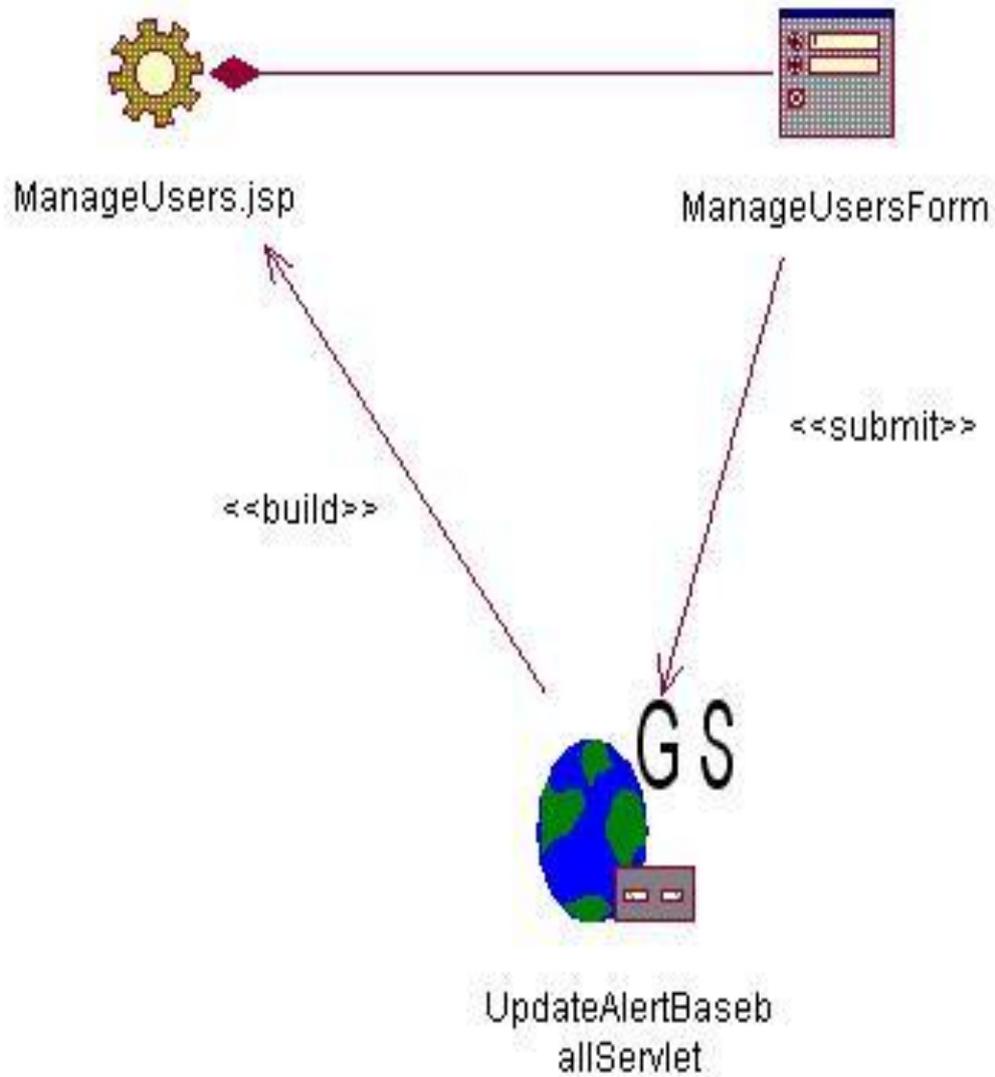
Anexo 1. 1 Diagrama de clases. Historia de Usuario Gestionar Equipo.



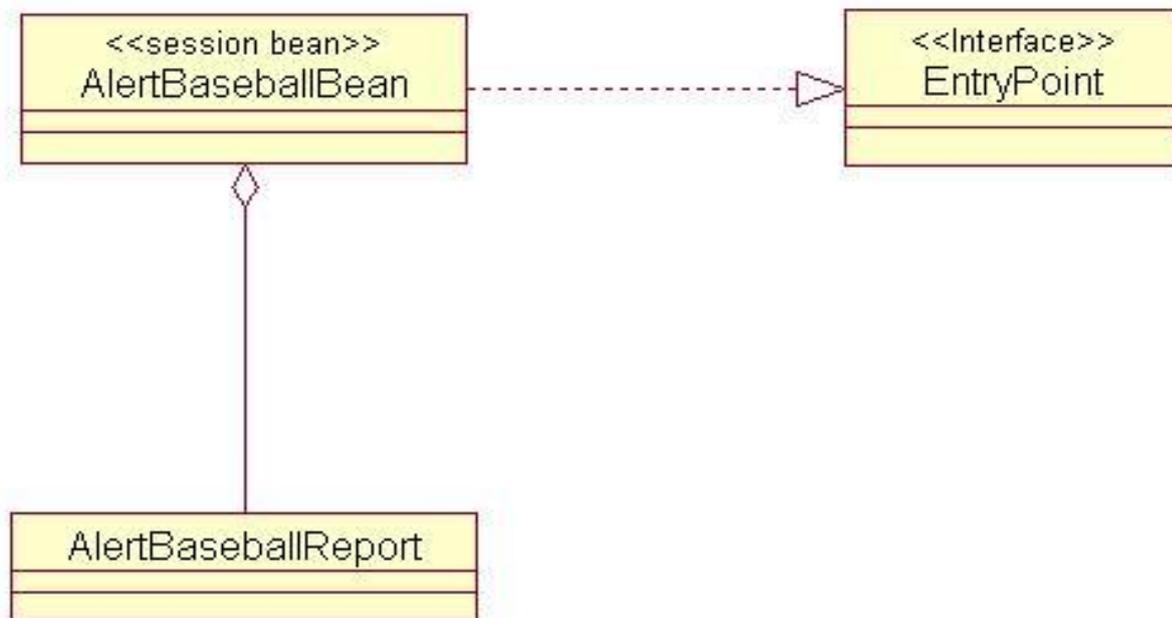
Anexo 1. 2 Diagrama de clases. Historia de Usuario Gestionar Estadio.



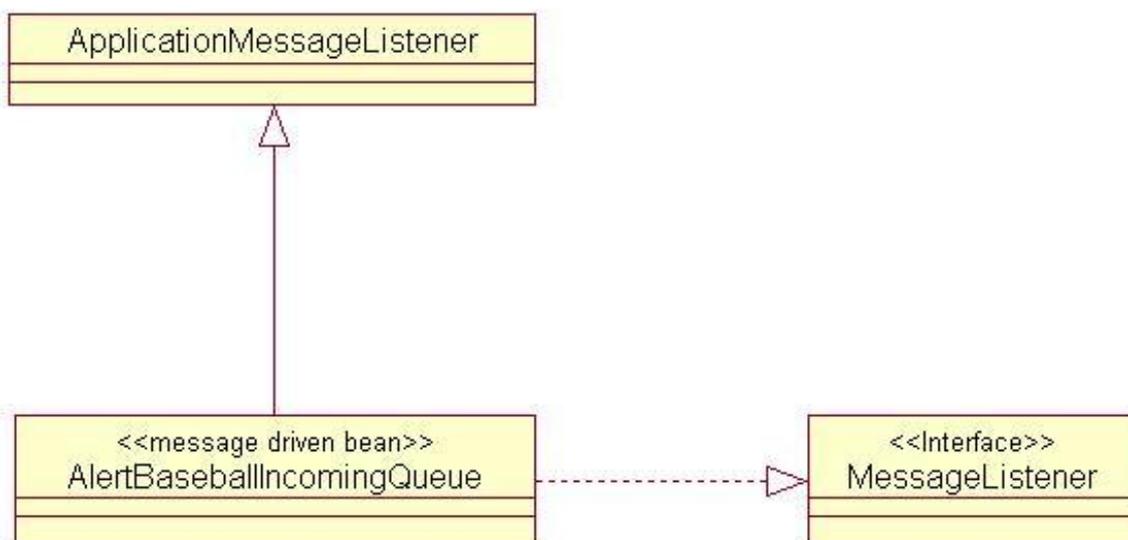
Anexo 1. 3 Diagrama de clases. Historia de Usuario Gestionar Juego.



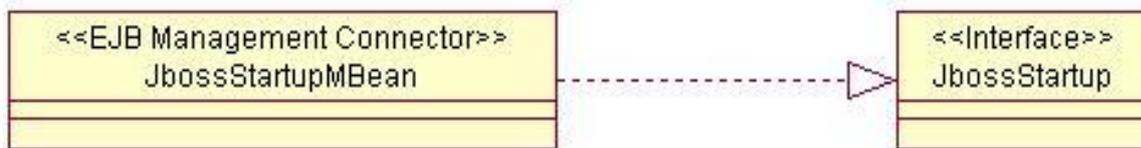
Anexo 1. 4 Diagrama de clases. Historia de Usuario Gestionar Usuario.



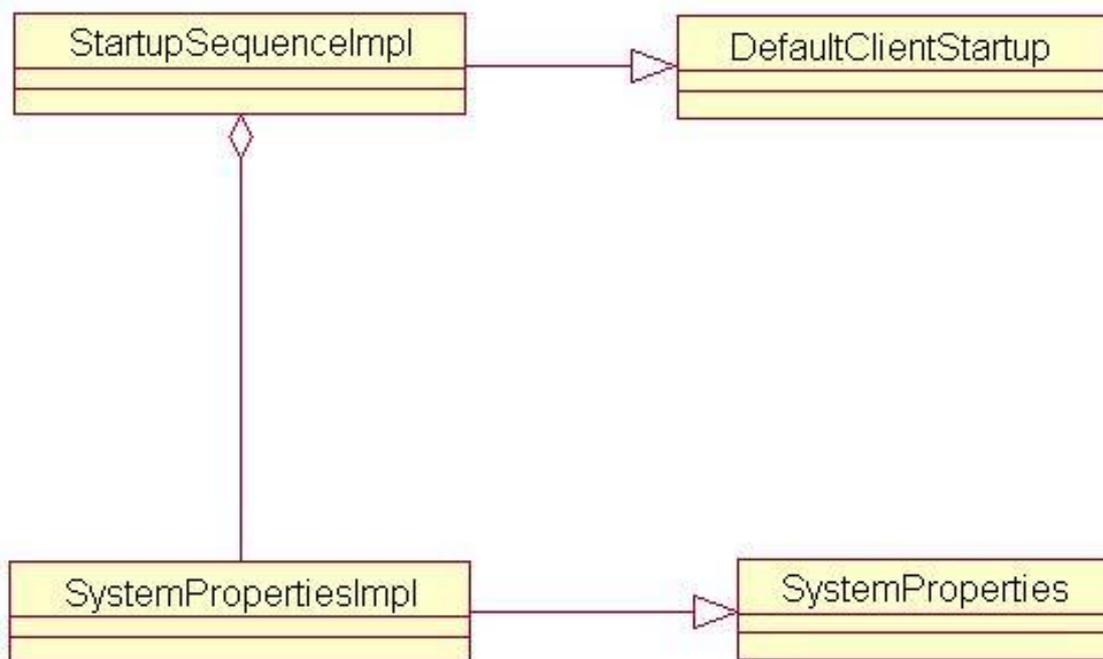
Anexo 1. 5 Diagrama de clases. Paquete sms_mms.ejb.



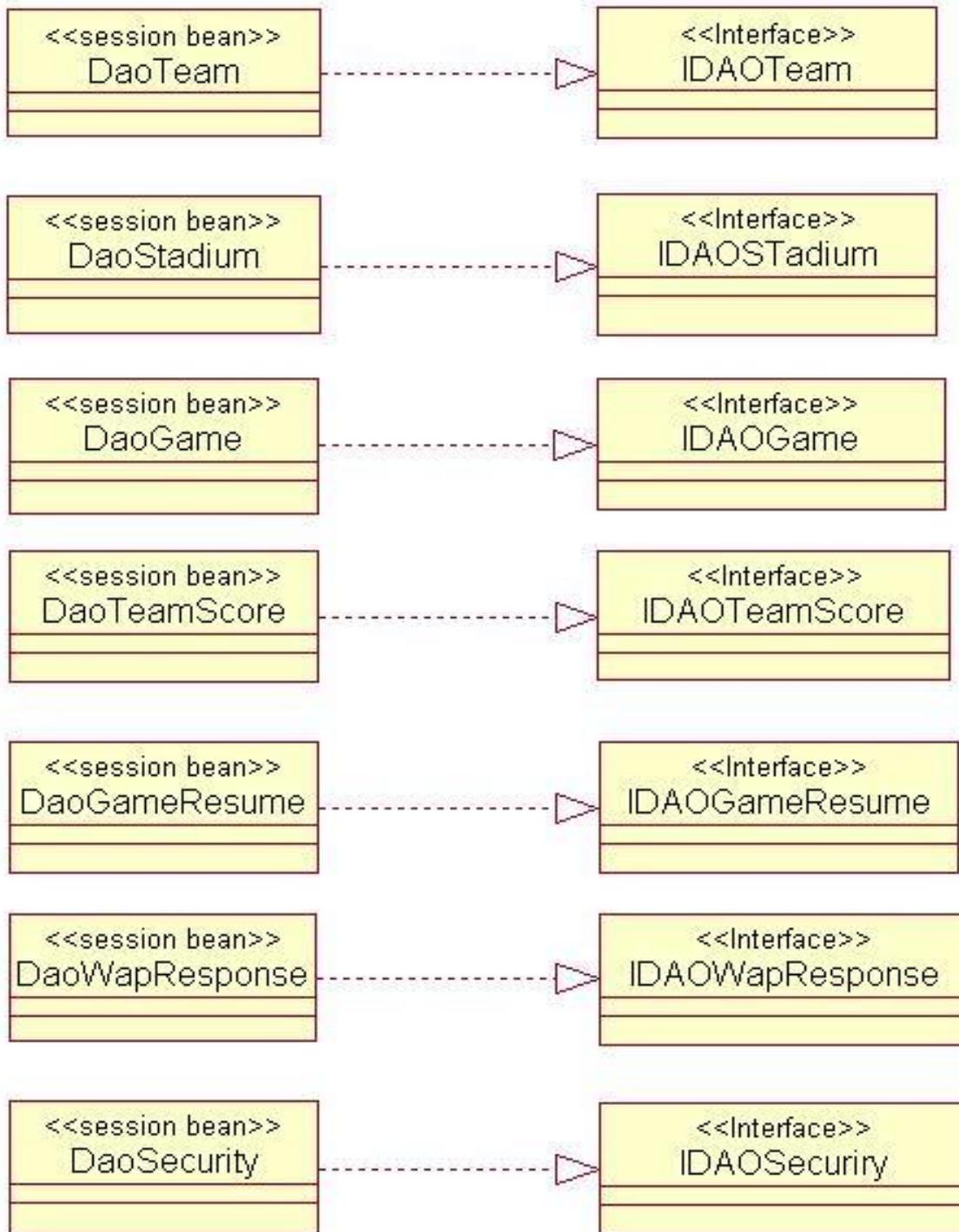
Anexo 1. 6 Diagrama de clases. Paquete sms_mms.io.



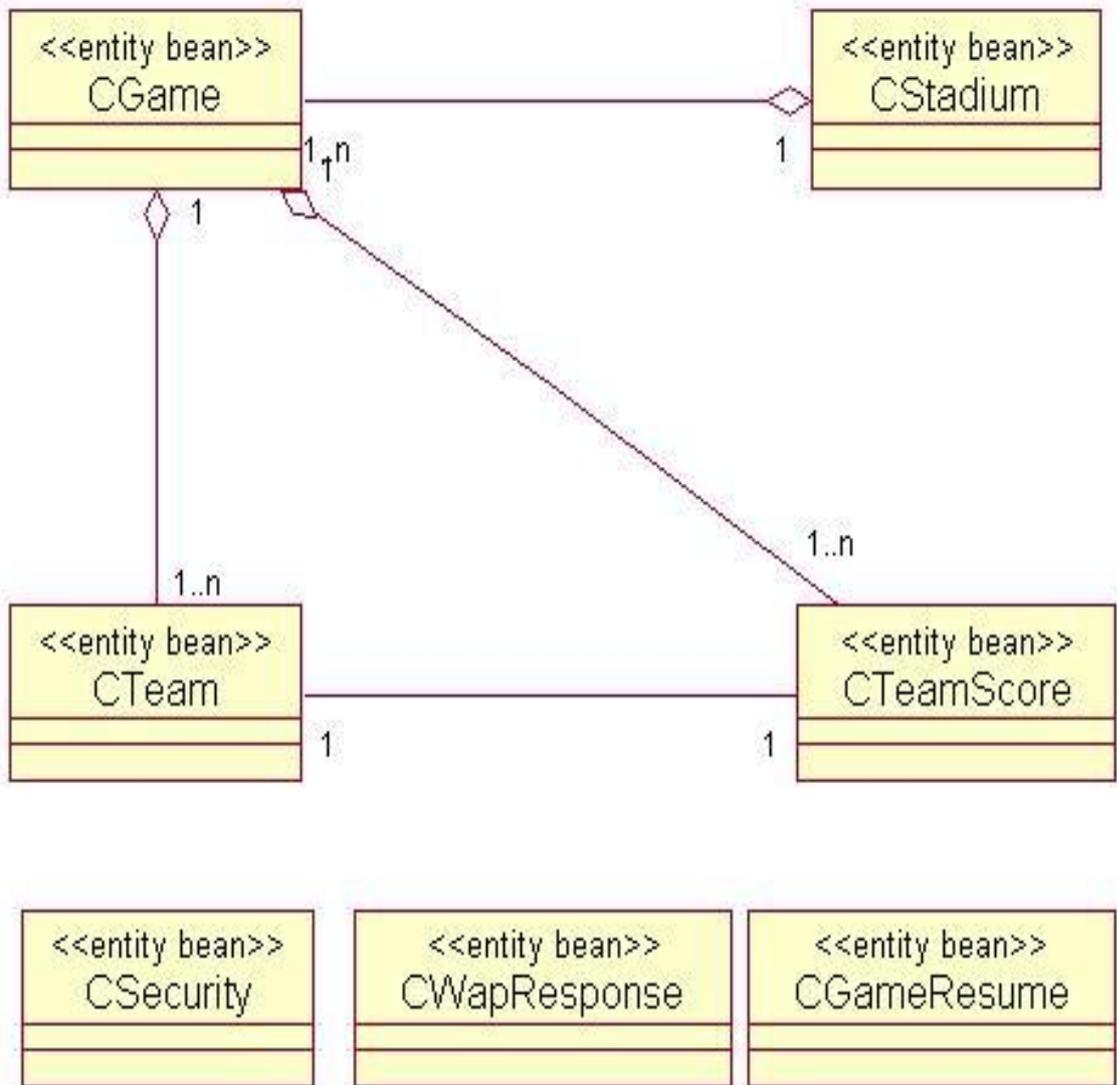
Anexo 1. 7 Diagrama de clases. Paquete sms_mms.startup.jboss.



Anexo 1. 8 Diagrama de clases. Paquete sms_mms.startup.



Anexo 1. 9 Diagrama de clases. Modelo de acceso a datos.



Anexo 1. 10 Diagrama de clases. Modelo de persistencia de datos.

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: Solicitar servicio con respuesta SMS.	
Descripción: Prueba para verificar la funcionalidad de responder una solicitud vía SMS.	
Condiciones de Ejecución: <ul style="list-style-type: none">• El servidor debe estar iniciado y ejecutándose.• Blueeye debe estar desplegado en el servidor y en escucha.• Se enviará un SMS con la palabra clave sbb y la abreviatura del equipo.	
Entrada/Pasos de Ejecución: Se intenta enviar una respuesta SMS al teléfono que realizó la solicitud.	
Resultado Esperados: El teléfono recibe una respuesta vía SMS.	
Evaluación de la Prueba: Prueba Satisfactoria	

Anexo 2. 11 Gestionar Solicitud de Servicio con respuesta SMS

Caso de Prueba de Aceptación	
Código: HU1_P2	Historia de Usuario: 1 – Gestionar Solicitud de Servicio.
Nombre: Solicitar servicio con respuesta SMS WAP Push.	
Descripción: Prueba para verificar la funcionalidad de responder una solicitud vía SMS WAP Push.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • El servidor debe estar iniciado y ejecutándose. • Blueeye debe estar desplegado en el servidor y en escucha. • Se enviará un SMS con la palabra clave bb y la abreviatura del Equipo. 	
Entrada/Pasos de Ejecución: Se intenta enviar una respuesta SMS WAP Push al teléfono que realizó la solicitud.	
Resultado Esperados: El teléfono recibe una respuesta vía SMS WAP Push.	
Evaluación de la Prueba: Prueba Satisfactoria	

Anexo 2. 12 Gestionar solicitud de servicio con respuesta SMS WAP Push

Caso de Prueba de Aceptación	
Código: HU1_P3	Historia de Usuario: 1 – Gestionar Solicitud de Servicio.
Nombre: Solicitar un juego.	
Descripción: Prueba para verificar que se envíe el último juego efectuado del equipo solicitado.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • El servidor debe estar iniciado y ejecutándose. • Blueeye debe estar desplegado en el servidor y en escucha. • Se enviará un SMS con la palabra clave sbb y la abreviatura de un equipo específico. 	
Entrada/Pasos de ejecución: Se intenta enviar una respuesta SMS al teléfono que realizó la solicitud con el último juego realizado por el equipo especificado.	
Resultado Esperados: El teléfono recibe una respuesta vía SMS que contiene un resumen del último juego del equipo solicitado. .	
Evaluación de la Prueba: Prueba Satisfactoria.	

Anexo 2. 13 Solicitar un Juego.

Pruebas de aceptación de la Historia de Usuario **Gestionar Estadio**.

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2 – Gestionar Estadio.
Nombre: Insertar un estadio.	
Descripción: Prueba para verificar que se inserte un estadio en la base de datos.	
Condiciones de Ejecución: <ul style="list-style-type: none">• No debe existir registrado el equipo que se adiciona.	
Entrada/Pasos de ejecución: Se intenta registrar el estadio en la base de datos	
Resultado Esperados: Se registra correctamente el estadio en la base de datos	
Evaluación de la Prueba: Prueba Satisfactoria.	

Anexo 2. 14 Insertar estadio.

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2 – Gestionar Estadio.
Nombre: Modificar un estadio.	
Descripción: Prueba para verificar que se modifique un estadio en la base de datos	
Condiciones de Ejecución:	
Entrada/Pasos de ejecución: Se intenta modificar un estadio en la base de datos	
Resultado Esperados: Se modifica correctamente el estadio en la base de datos	
Evaluación de la Prueba: Prueba Satisfactoria.	

Anexo 2. 15 Modificar estadio.

Pruebas de aceptación de la Historia de Usuario **Gestionar Equipo**.

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 2 – Gestionar Equipo.
Nombre: Insertar un equipo.	
Descripción: Prueba para verificar que se inserte un equipo en la base de datos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • No debe existir registrado el equipo que se adiciona. • La imagen que se adiciona debe tener una resolución menor de 32 x 32 pixeles, de lo contrario no se adjutará imagen para el equipo. 	
Entrada/Pasos de Ejecución: Se intenta registrar el equipo en la base de datos	
Resultado Esperados: Se registra correctamente el equipo en la base de datos	
Evaluación de la Prueba: Prueba Satisfactoria.	

Anexo 2. 16 Insertar equipo.

Caso de Prueba de Aceptación	
Código: HU3_P2	Historia de Usuario: 2 – Gestionar Equipo.
Nombre: Modificar un equipo.	
Descripción: Prueba para verificar que se modifique un equipo en la base de datos	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La imagen que se adiciona debe tener una resolución menor o igual que 32 x 32 pixeles, de lo contrario no modificar la imagen. 	
Entrada/Pasos de ejecución: Se intenta modificar un equipo en la base de datos	
Resultado Esperados: Se modifica correctamente el equipo en la base de datos	
Evaluación de la Prueba: Prueba Satisfactoria.	

Anexo 2. 17 Modificar un equipo.

Pruebas de aceptación de la Historia de Usuario **Gestionar Juego**.

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 2 – Gestionar Juego.
Nombre: Crear un juego.	
Descripción: Prueba para verificar que se inserte un juego en la base de datos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Deben existir al menos 2 equipos que no se encuentren jugando. • Debe existir al menos un estadio donde no se este efectuando ningún juego. 	
Entrada/Pasos de ejecución: Se intenta registrar el juego en la base de datos.	
Resultado Esperados: Se registra correctamente el juego en la base de datos	
Evaluación de la Prueba: Prueba Satisfactoria.	

Anexo 2. 6 Crear un Juego.

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: 2 – Gestionar Juego.
Nombre: Modificar juego.	
Descripción: Prueba para verificar que se modifique un juego en la base de datos	
Condiciones de Ejecución:	
Entrada/Pasos de Ejecución: Se intenta modificar un juego en la base de datos	
Resultado Esperados: Se modifica correctamente el juego en la base de datos	
Evaluación de la Prueba: Prueba Satisfactoria.	

Anexo 2. 7 Modificar un juego.

Pruebas de aceptación de la Historia de Usuario **Gestionar Respuesta WAP.**

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 2 – Gestionar Respuesta WAP.
Nombre: Mostrar sitio WAP.	
Descripción: Prueba para verificar que se muestren correctamente los datos del sitio WAP.	
Condiciones de Ejecución: <ul style="list-style-type: none">• El sitio debe estar desplegado.	
Entrada/Pasos de ejecución: Se intenta acceder al sitio WAP.	
Resultado Esperados: El sitio se muestra correctamente.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Anexo 2. 8 Mostrar sitio WAP.

GLOSARIO

SVA (Servicio de Valor Agregado): Son servicios que utilizan como soporte servicios básicos, telemáticos, de difusión o cualquier combinación de estos, prestados a través de una red de telecomunicaciones, que proporcione un beneficio adicional, independientemente de la tecnología.

SMS (Servicio de Mensajes Cortos): (del inglés *Short Messaging Service*) Los mensajes pueden tener hasta 160 caracteres alfanumérico.

SMS WAP Push: SMS que contiene una URL y redirecciona automáticamente a un sitio WAP.

GSM: Sistema estándar para comunicación utilizando teléfonos móviles que incorporan tecnología digital.

WBMP: Imágenes monocromáticas creadas especialmente para teléfonos de bajas prestaciones (teléfonos de pantalla monocromáticos).

MMS (Servicio de Mensajería Multimedia): (del inglés *Multimedia Messaging Service*) Es usado para la transferencia de archivos de música, imágenes y video.

PDA (Asistente Personal Digital): (del inglés *Personal Digital Assistant*) Es un dispositivo de pequeño tamaño que combina un ordenador, teléfono/fax, internet y conexiones de red.

HTML (Lenguaje de Marcado de Hipertexto): (del inglés *HyperText Markup Language*) Es el lenguaje de marcado predominante para la construcción de páginas Web.

JavaScript: Es un lenguaje de programación interpretado, es decir, no requiere compilación. Utilizado principalmente en páginas Web.

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

JSP: Es una tecnología Java que permite general contenido dinámico para Web.

CGI: Es una tecnología del Internet que permite a un cliente solicitar datos de un programa ejecutado en un servidor Web.

Servlet: Son objetos que corren dentro del contexto de un contenedor de Servlet.

Portlet: Son componentes modulares de interfaz de usuario gestionadas y visualizadas en un portal Web.

JVM (Máquina Virtual de Java): (del inglés *Java Virtual Machine*) Es un programa ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial el cual es generado por el compilador del lenguaje Java.

JBOSS: Es un servidor de aplicaciones de código abierto implementado en Java puramente.

Plug-ins: Es una aplicación informática que interactúa con otra aplicación para soportar alguna función o utilidad específica.

Apache Tomcat: Funciona como un contenedor de Servlets, implementa la especificaciones de los Servlet y JSP.

URL (Localizador Uniforme de Recurso): (del inglés *Uniform Resource Locator*) 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.