

Universidad de las Ciencias Informáticas

Facultad 2



Título: Sistema de Servicio de Información Cultural para Celulares.

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

Autor(es):

Yanira Hernández Acosta.

Yudenia De La Rosa Simón.

Tutor(es):

Ing. Iris Margarita Reina Heredia.

Consultante:

Ing. Lex Karel Zayas Hernández.

Ciudad Habana, Junio de 2010

DECLARACIÓN DE AUTORÍA

Declaramos ser autoras de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo para su beneficio.

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

Autoras:

Tutor:

Yudenia De La Rosa Simón

Yanira Hernández Acosta

Ing. Iris Margarita Reina

DATOS DE CONTACTO

Nombre: Iris Margarita Reina Heredia

Correo: imreina@uci.cu

Título: Ing. en Ciencias Informáticas.

Centro de Estudios: Universidad de Ciencias Informáticas. Ciudad de la Habana. Cuba.

Graduado: Julio del 2008.

Honores: Título de Oro.

Experiencia profesional:

Comenzó en su etapa de estudiante, donde trabajó como analista del proyecto SIGESC durante 2 años. Al obtener el título de Ingeniera en Ciencias Informáticas se desempeñó como Líder del grupo de Gestión y Planificación de proyectos del centro de Telemática de la facultad 2, lugar donde se ha desempeñado además como profesora de pregrado. Prestó además sus conocimientos de análisis en el proyecto TELTRONIC durante 1 año.

AGRADECIMIENTOS

A mi familia en especial a mis abuelos (Julio y Elena) y a mi madre, quienes me infundieron la ética y el rigor que guían mi transitar por la vida.

A mis tíos Carmelina, Juan, July, José y Rafael, por confiar en mí.

A Osmari, mi mejor amiga, mi hermana, mi consejera, la persona más correcta y sensata del mundo.

A todos mis amigos durante esto 5 años, con los cuales he compartido y vivido momentos inolvidables, los cuales nunca se me olvidaran.

A Yanira mi amiga y compañera de tesis, por estar siempre presente en todos los momentos.

A mi tutora Iris que siempre nos estuvo apoyando en todo momento y por habernos enseñado tanto.

A todos mis amigos que me apoyaron y ayudaron a la realización de este trabajo.

Yudenia de la Rosa Simón.

A mis padres por ser lo más importante que tengo en la vida, por haberme hecho la mujer que hoy soy, a ellos les debo tanto, gracias por su amor su empeño y sacrificio diario.

A mi hermanito Yanel, al que quiero tanto, le doy gracias a la vida por haberme dado un hermano como tú, espero no decepcionarte nunca, porque he tratado de ser un ejemplo para ti.

A mi segunda madre, Rosa, gracias por quererme como una hija y haberme dado tu cariño incondicionalmente

A mis todos mis tíos, a los cuales quiero muchísimo, gracias por haber confiado en mí.

A todas mis amistades durante estos 5 años, la gente con la cual he compartido y vivido momentos inolvidables, los cuales llevaré siempre en mi corazón

A mis tutora por haberme guiado a lo largo de este tiempo, por su comprensión y dedicación continua

A todas las personas que directa o indirectamente, contribuyeron a hacer realidad mi sueño.

A todos ellos Gracias!!!!!!

Yanira Hernández Acosta.

DEDICATORIA

A mi madre por ser una mujer espectacular y mi ejemplo a seguir, por darme vida, amor, confianza, sueños y esperanza.

A mi abuela Elena que ha sido como una segunda madre para mí, brindándome mucho amor a lo largo de toda mi vida.

A mi familia por estar siempre apoyándome.

A todos mis amigos y amigas que estuvieron ahí para darme ánimo.

Yudenia de la Rosa Simón.

Le dedico este trabajo y todo mi esfuerzo a mis padres a los cuales les debo tanto. Por complacerme en todas las cosas y creer en mí cuando yo perdía las esperanzas. A mi hermano por ser mi amigo, mi motivo para seguir adelante y trazarle un camino honroso y digno. A todos mis tíos en especial a mis tíos... José, por estar apoyándome en mi carrera. A Rosa por ser mi segunda madre y a Mary. En especial a mis amistades y mis amigas del alma.

Yanira Hernández Acosta.

RESUMEN

La tecnología de teléfonos móviles ha tomado un gran auge que va en crecimiento a través de los años, por lo que la demanda de contenido en este ámbito ha provocado que la administración de estos se haya convertido en una rama importante del desarrollo mundial. El avance en la telefonía móvil ha permitido que los teléfonos no solo se utilicen para comunicarse mediante llamadas telefónicas, o mensajes de texto sino que ha propiciado al usuario una gran variedad de servicios de valor agregado para un mayor uso de la misma, al cual también se le suma las conexiones a internet, los juegos y muchas más aplicaciones. La empresa ETECSA se encuentra inmersa en la tarea de entrar a un mercado cada vez mayor en las telecomunicaciones y servicio de valor agregado para los celulares, acompañado a esto se encuentra el número creciente de usuarios de estos teléfonos celulares. Es por esto que se hace necesario el desarrollo de un sistema de Servicio de Mensajes Cortos que permita a los usuarios conocer informaciones culturales. Por lo que el presente trabajo está dirigido principalmente a la creación de un sistema que permita el envío de mensajes para los usuarios de telefonía móvil que deseen conocer información sobre eventos de cultura que estarán ocurriendo.

PALABRAS CLAVE

- Celulares.
- Dispositivos móviles.
- ETECSA.
- Servicio de valor agregado.
- SMS.
- Servicio de información cultural.

INDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Introducción	4
1.2 Estado del arte	4
1.2.1 Telefonía Celular en Cuba.....	5
1.2.2 Servicios más utilizados en la actualidad.....	5
1.3 Servicios de valor agregado	7
1.3.1 Tipo de servicios de valor agregado.....	7
1.3.2 Plataforma en las que se desarrollan los servicios de valor agregado.....	8
1.4 Plataforma de Desarrollo	10
1.4.1 Actores principales del mercado móvil.....	12
1.4.2 JSR 212.....	13
1.4.3 Framework utilizados.....	14
1.5 Gestor de Base de Datos	17
1.5.1 Gestor de Base de Datos PostGreSQL.....	17
1.6 Metodología de desarrollo	17
1.6.1 Características esenciales.....	18
1.7 Herramientas de desarrollo utilizadas	19
1.7.1 Entorno de desarrollo.....	20
1.7.2 Entornos de desarrollo Eclipse y NetBeans.....	20
1.7.4 Visual Paradigm.....	21
1.8 Conclusiones	22
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	23
2.1 Introducción	23
2.2 Problema y situación problemática	23
2.3 Objeto de Automatización	23
2.4 Propuesta del Sistema	23
2.5 Modelo De Dominio	24
2.5.1 Conceptos del Modelo de Dominio.....	24

2.6 Requerimientos del Sistema.	25
2.6.1 Requerimientos Funcionales.	25
2.6.2 Requerimientos No Funcionales.	26
2.7 Modelo de Casos de Uso del Sistema.	26
2.7.1 Definición de los actores del sistema a automatizar.	27
2.7.2 Listado de Casos de Uso del sistema.	27
2.7.3 Diagrama de Casos de Uso del sistema (CUS).	28
2.7.4 Descripción de los Casos de Uso del sistema.	28
2.8 Conclusiones.	37
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	38
3.1 Introducción.	38
3.2 Modelo de Análisis.	38
3.2.1 Diagramas de Clases de Análisis.	39
3.3 Modelo de Diseño.	40
3.3.1 Diagramas de Clases del Diseño.	41
3.3.2 Diagramas de Secuencia	43
3.3.3 Modelo lógico de datos.	44
3.3.4 Modelo físico de datos (modelo de datos).	45
3.4 Estilos Arquitectónicos.	45
3.5 Patrones de Diseño.	46
3.5.1 Tipo de Patrones Utilizados.	46
3.6 Conclusiones.	47
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	48
4.1 Introducción.	48
4.2 Modelo de Implementación.	48
4.2.1 Diagrama de Componentes.	48
4.2.2 Diagrama de Despliegue.	51
4.3 Modelo de Pruebas.	51
4.3.1 Métodos de Prueba.	52
4.3.2 Diseño de Casos de Prueba.	52
4.4 Conclusiones.	56

CAPÍTULO 5: FACTIBILIDAD DEL SISTEMA	57
5.1 Introducción	57
5.2 Método de estimación Puntos por Casos de Uso	57
5.2.1 Cálculo de Puntos de Casos de Uso sin ajustar.....	58
5.2.2 Cálculo de Puntos de Casos de Uso ajustados.....	59
5.2.3 Cálculo del Esfuerzo.....	62
5.2.4 Distribución del Esfuerzo entre las diferentes actividades del sistema.....	62
5.3 Beneficios tangibles e intangibles	71
5.4 Análisis de costos y beneficios	71
5.5 Conclusiones	64
CONCLUSIONES	73
RECOMENDACIONES	74
BIBLIOGRAFÍA	75
REFERENCIAS BIBLIOGRÁFICAS	78

INTRODUCCIÓN

Las telecomunicaciones móviles en especial la telefonía celular, están revolucionando la industria de las telecomunicaciones. Desde su irrupción en el ámbito comercial a principios de los ochenta, la industria celular en el mundo ha venido creciendo en forma acelerada, con especial dinamismo en la década de los noventa y primeros años del siglo XXI, período este en el cual su crecimiento ha sido espectacular.

Los teléfonos celulares se han convertido en una herramienta primordial para las personas comunes y de negocios.

El gran avance de la tecnología ha hecho que los celulares brinden una serie de funciones que años atrás se veían algo inalcanzables, como los juegos, reproducción de archivos MP3 y MP4, la conexión mediante bluetooth e infrarrojos, conectividad USB, envío y recepción de SMS (Short Messages Standard), EMS (Enhanced Messages Services) y MMS (Multimedia Messages Services), además de opciones de agenda electrónica y televisión digital.

El crecimiento acelerado del uso de la telefonía celular, 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 e Internet.

El servicio de la telefonía celular se inaugura oficialmente en Cuba el 24 de Febrero de 1993, por la Empresa Teléfonos Celulares de Cuba, S.A. (CUBACEL). En los primeros años el sistema utilizado fue el denominado Advanced Mobile Phone System o Sistema Avanzado de Telefonía Móvil (AMPS). El GSM comenzó a ser introducido y explotado en Cuba en el año 2001 por la empresa estatal C_COM. Estas empresas están asociadas a ETECSA que es una organización cubana de capital mixto y tiene como objeto social prestar servicios públicos de telecomunicaciones, mediante la operación, instalación, explotación, comercialización y mantenimiento de redes públicas de telecomunicaciones en Cuba.

Esta empresa tiene una alta responsabilidad en el desarrollo socio-económico del país y en especial, en la informatización de la sociedad, garantizando una efectiva conectividad. Desde su creación ETECSA ha mantenido un avance sostenido con logros progresivos en el transcurso de sus últimos años de existencia. Actualmente se encuentra inmersa en la tarea de entrar a un mercado cada vez mayor en las telecomunicaciones, incrementando el número de servicios de valor agregado para sus clientes. Teniendo en cuenta lo servicios que brinda la Empresa de Telecomunicaciones de Cuba S.A

y su afán de mejorar su aceptación con los usuarios, se desea desarrollar un sistema que permita la suscripción de usuarios y envío de mensajes sobre los eventos culturales que están ocurriendo en el país.

Atendiendo a las posibilidades tecnológicas que existen en la universidad y la problemática de ETECSA antes planteada surge como **problema científico** ¿Cómo facilitar la suscripción y el envío de noticias sobre eventos culturales mediante SMS a los usuarios de Cubacel?

Para el desarrollo de la investigación se define como **objeto de estudio** La prestación de servicios de mensajería móvil. El **campo de acción** queda enmarcado en el servicio de información sobre los eventos de cultura para dispositivos móviles.

Con el fin de resolver el problema planteado se define como **objetivo general** desarrollar un sistema que permita la suscripción y el envío de mensajes a teléfonos celulares sobre información cultural.

Para darle cumplimiento al objetivo general se definieron las siguientes **tareas de investigación**:

- Realizar un estudio del estado del arte.
- Obtener una visión del negocio en cuestión, de manera que se puedan identificar de una forma rápida y confiable las funcionalidades con las que debe cumplir el sistema a desarrollar.
- Aplicar para el diseño y la construcción del software una arquitectura acorde a las características del sistema.
- Seleccionar y ajustar, según las características propias del equipo de desarrollo y de las actividades a realizar, una metodología de desarrollo de software que facilite el trabajo y garantice la calidad del proceso y del producto.
- Implementar las funcionalidades requeridas para llevar a cabo el desarrollo de la aplicación.
- Desarrollar pruebas al sistema obtenido.
- Seleccionar las tecnologías y herramientas necesarias para el desarrollo de un sistema de envío de mensajes para celulares.
- Aplicar al desarrollo del software patrones de diseño y estilos arquitectónicos de manera que se obtenga un sistema robusto, flexible y organizado.
- Realizar un estudio de la factibilidad de desarrollo del producto mediante el análisis de costo-beneficios.

Este trabajo de diploma está conformado por cinco capítulos, en los cuales se tratará todo el estudio realizado, así como la propuesta de solución.

Capítulo 1: Fundamentación Teórica

En este capítulo se describen cada una de las tecnologías, metodologías, lenguajes, conceptos y herramientas a usar para el desarrollo del sistema. Además, se expone el estado del arte de los sistemas de mensajería móvil.

Capítulo 2: Características del Sistema

En este capítulo se presentan las características generales del sistema, se procederá a realizar la captura de requisitos funcionales y no funcionales a los que se debe dar cumplimiento, se examinan los temas referentes a las necesidades de los clientes para obtener las bases y definir el objeto de automatización y la propuesta de solución que será desarrollada.

Capítulo 3: Análisis y Diseño del Sistema

Este capítulo se centra principalmente en el análisis y diseño del sistema, obteniendo como principal resultado los diagramas de clases del diseño y los diagramas de interacción. Se define además la arquitectura del sistema.

Capítulo 4: Implementación y Prueba

En este capítulo se hace referencia a las disciplinas de implementación y prueba. Se describe el modelo de implementación utilizado mediante los diagramas de componentes y de despliegue. Se describe el diseño de los casos de prueba y el resultado de las pruebas realizadas.

Capítulo 5: Estudio de la Factibilidad

En el capítulo 5 se hace una estimación del costo de realización del sistema, calculando una aproximación del tiempo de desarrollo y el análisis de la factibilidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

En este capítulo se describen los principales aspectos y conceptos de relevancia que constituyen objeto de análisis para el desarrollo de la investigación. Las tecnologías y metodología a utilizar, el lenguaje de programación a elegir y se expone el estado del arte referente a las tendencias y herramientas que se utilizan para la creación de servicios de valores agregados en la telefonía móvil.

1.2 Estado del arte.

En la actualidad hay muchas empresas que dedican su labor a la prestación de servicios de telefonía celular, que es un dispositivo inalámbrico electrónico que permite tener acceso a la red debido a las antenas repetidoras que la conforman, cada una de las cuales es una célula.

Su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar, sumado a esto hay una serie de SVA (Servicio de Valor Agregado) como 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 (Short Messages Services), de forma tal que estos reciban toda la información pertinente que solicitaron. A continuación se describen algunos de los servicios que se brindan a través del envío de SMS.

El servicio de mensajes cortos o SMS (Short Messages Services) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto, o más coloquialmente, textos o mensajitos) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. En un principio el SMS se pensó como una manera de avisar al usuario, por ejemplo, de llamadas perdidas o mensajes en el buzón de voz, sin embargo, pocos creyeron que el SMS sería usado como el medio para enviar mensajes de texto de un usuario móvil a otro.

El primer mensaje comercial SMS fue enviado por la red GSM de Vodafone el 3 de diciembre de 1992 en el Reino Unido a través de un ordenador el cual el texto del mensaje era “la navidad alegre”.

El crecimiento inicial era lento, con clientes en 1995 que rondan el promedio de sólo 0.4 mensajes por cliente GSM por mes. Un factor clave de la lentitud del SMS era que los operadores eran lentos para establecer el cobro de sistemas, sobre todo para suscriptores con pago por adelantado donde era difícil eliminar el fraude de facturación. Con el tiempo, el fraude de facturación fue eliminado por la facturación de interruptor y así hacia el final de 2000, el número medio de mensajes por usuario alcanzó una cifra de 35 mensajes.

Comercialmente el SMS es una industria masiva que 2006 merece más de 80 mil millones de dólares a escala mundial. SMS tiene un precio medio global de 11 centavos y mantiene un margen de beneficio del 90 %.

1.2.1 Telefonía Celular en Cuba.

Desde los inicios de la telefonía celular en Cuba se ha visto con un incremento limitado de su tecnología desde la aparición en 1993 de este recurso, los usuarios no disponían de toda la cantidad de servicios que son brindados por móviles. Los interesados en la telefonía celular en Cuba disponen ahora de un paquete de servicios agregados mediante mensajes de texto, una opción que en la isla parecía de “ciencia ficción” cuando apareció este servicio. Las comodidades y usos diversos que ya tienen en el mundo los teléfonos celulares no se habían expandido antes en el país por la carencia de una plataforma tecnológica que los garantizara, ya existen diferentes plataformas lo que no existen son los variados servicios. Ahora se puede consultar desde los móviles las carteleras de las salas de cine, la ubicación de las embajadas, el estado del tiempo y el curso de un huracán, entre otros datos. La opción de votar con mensajes de texto (SMS, por sus siglas en inglés) en los concursos convocados por el Festival Internacional de Cine de La Habana y la competencia “Lucas”, que elige al mejor video clip musical del año en la isla fue uno de los servicios que se brindaron recientemente.

El desarrollo de la telefonía celular o móvil ha sido impetuoso en los últimos años. Tanto los servicios que se prestan, las capacidades de comunicación, ancho de banda, estandarización de códigos, señales y servicios a nivel internacional, así como el desarrollo de los teléfonos utilizados, que han dado un importante salto en calidad y posibilidades.

1.2.2 Servicios más utilizados en la actualidad.

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 clave 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 re-direccionado al sitio WAP que contiene la información que fue solicitada (1).

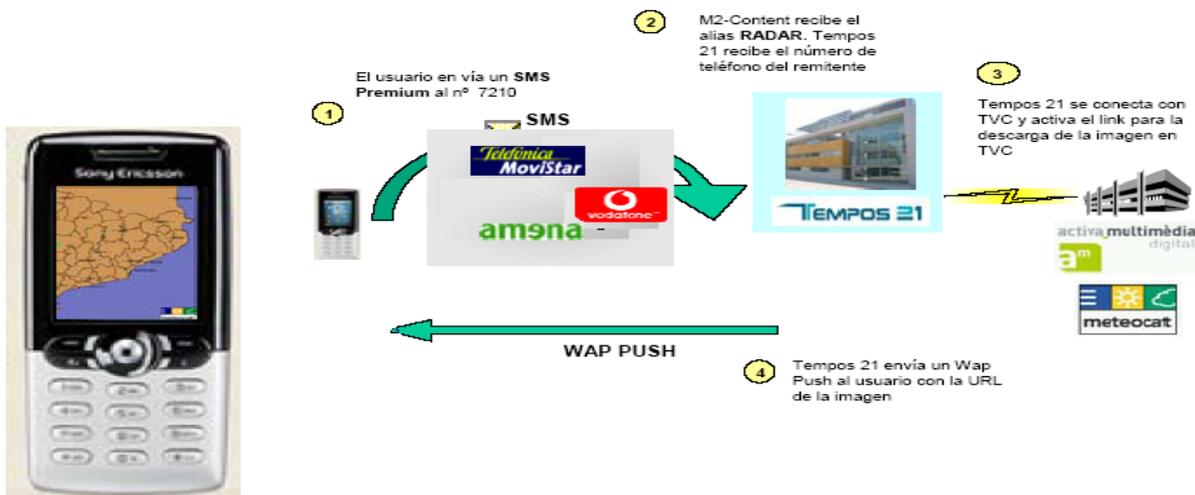


Figura 1 Dinámica desarrollada por Tempos 21 en el servicio del tiempo.

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, también se parece al servicio de deporte antes explicado, en la figura 3 se brinda una idea de cómo se comporta este servicio.



Figura 3 Servicio de información de Horóscopo.

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.

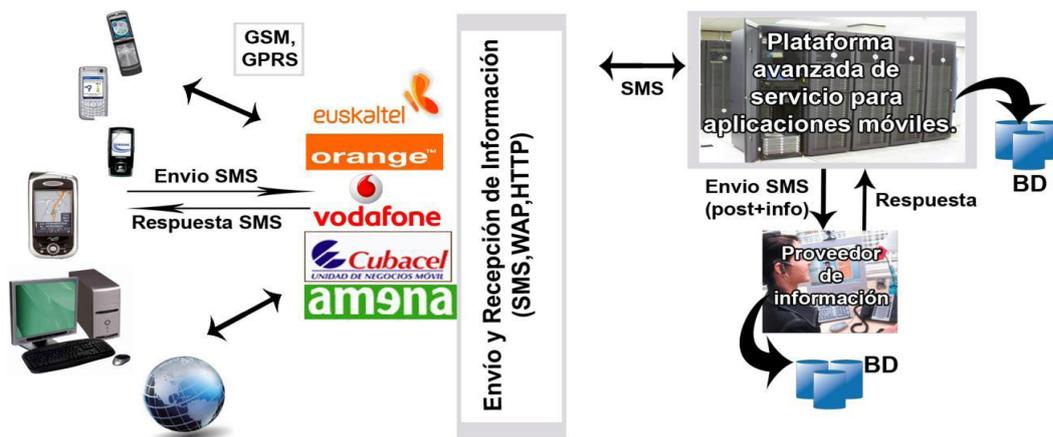


Figura 4: Esquema del Servicio de envío de información.

1.3 Servicios de valor agregado.

1.3.1 Tipo de servicios de valor agregado.

En la telefonía móvil los SVA aparecen cuando los servicios básicos de SMS, Servicio de Mensajes Multimedia y WAP- PUSH comienzan a utilizarse con el objetivo de obtener un servicio adicional. Hoy existe un amplio desarrollo 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 (2).

1.3.2 Plataforma en las que se desarrollan los servicios de valor agregado.

Con la introducción de los servicios de datos los operadores se han visto obligados a incorporar nuevas tecnologías que posibiliten esta nueva oferta de servicios. Las plataformas más utilizadas para los servicios de datos son:

Java 2 Micro Edition (J2ME): Es la plataforma de desarrollo de aplicaciones más utilizada en el mundo y fue creada originalmente para enfrentar los problemas derivados de su uso en dispositivos pequeños.

Esta tecnología está basada en tres elementos:

Configuración: provee un conjunto básico de bibliotecas y capacidades de la máquina virtual para un gran abanico de dispositivos. Define una mínima plataforma para una categoría horizontal de dispositivos, cada uno con requerimientos similares en cantidades de memoria total y poder de procesamiento. Una configuración establece y define el lenguaje Java y las características de la máquina virtual y bibliotecas de clases mínimas que un fabricante de dispositivos o un proveedor de contenido espera encontrar en todos los dispositivos de una misma categoría.

Perfil: un conjunto de APIs para soportar un conjunto más reducido de dispositivos. Conformar una capa por encima de la configuración, y por lo tanto la extiende, al mismo tiempo que atiende la demanda específica de un determinado mercado “vertical” o familia de dispositivos. El objetivo principal de un perfil es garantizar interoperabilidad con una determinada familia de dispositivos o dominio, definiendo un estándar Java para dicho mercado. Los perfiles incluyen bibliotecas de clases que son más específicas de dominio que las clases provistas en una configuración

Paquetes opcionales: conjunto de APIs de tecnología específica.

La plataforma Java ME ha sido a su vez dividida en dos configuraciones de base: una para pequeños dispositivos móviles y otra diseñada para dispositivos móviles con más capacidad, como *Smart Phones* y sintonizadores externos de televisión por cable/satélite.

Brew: Aun siendo tecnológicamente agnóstico, sólo ha sido mayormente lanzado por operadores CDMA. Además de habilitar la creación de aplicaciones, proporciona un ecosistema conveniente para los operadores y desarrolladores de contenido.

WAP: Tecnología anteriormente criticada, en la actualidad es un aspecto fundamental del negocio de datos. Los portales WAP son los que acogen a las aplicaciones para ser descargadas.

A su vez, se han desarrollado sistemas operativos diseñados para las necesidades de los dispositivos móviles, con la suficiente flexibilidad para migrarlos a medida que los terminales adquieren mayor poder de procesamiento y memoria, entre otros factores.

Windows Mobile: Los operadores creen que es bueno ofrecerlo por la facilidad de uso por parte de los usuarios ya familiarizados con el Windows para PC. Este SO es una versión de Windows adaptada para las necesidades de los dispositivos móviles (3).

1.4 Plataforma de Desarrollo.

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 por el proyecto Cubacel.

Actualmente se le define como una plataforma tecnológica más que como un lenguaje. El lenguaje propiamente dicho tiene mucho de la sintaxis derivada de C y C++, pero un modelo de objetos más simple y menos facilidades de bajo nivel.

Las características principales que nos ofrece Java respecto a cualquier otro lenguaje de programación, son (4):

SIMPLE:

Ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje. Elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el reciclador de memoria dinámica. No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un thread de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria. Reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- aritmética de punteros.
- no existen referencias.
- registros (struct).
- definición de tipos (typedef).
- macros (#define).
- necesidad de liberar memoria (free).

ORIENTADO A OBJETOS:

Incorpora funcionalidades inexistentes en C++ como por ejemplo, la resolución dinámica de métodos. Esta característica deriva del lenguaje Objetivo C. En C++ se suele trabajar con librerías dinámicas (DLLs) que obligan a recompilar la aplicación cuando se retocan las funciones que se encuentran en su interior. Este inconveniente es resuelto por Java mediante una interfaz específica llamada RTTI (*RunTime Type Identification*) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el RunTime que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.

DISTRIBUIDO:

Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como HTTP y FTP. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. La verdad es que Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.

ROBUSTO:

Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas.

Además, para asegurar el funcionamiento de la aplicación, realiza una verificación de los bytes-codes, que son el resultado de la compilación de un programa Java. Es un código de máquina virtual que es interpretado por el intérprete. No es el código máquina directamente entendible por el hardware, pero ya ha pasado todas las fases del compilador: análisis de instrucciones, orden de operadores, etc., y ya tiene generada la pila de ejecución de órdenes.

PORTABLE:

Más allá de la portabilidad básica por ser de arquitectura independiente, implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Además, construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, PC o Mac.

DINÁMICO:

Se beneficia todo lo posible de la tecnología orientada a objetos. No intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. También simplifica el uso de protocolos nuevos o actualizados. Si su sistema ejecuta una aplicación Java sobre la red y encuentra una pieza de la aplicación que no sabe manejar, tal como se ha explicado en párrafos anteriores, es capaz de traer automáticamente cualquiera de esas piezas que el sistema necesita para funcionar.

SEGURO:

Resultan extremadamente seguras, ya que no acceden a zonas delicadas de memoria o de sistema, con lo cual evitan la interacción de ciertos virus. No posee una semántica específica para modificar la pila de programa, la memoria libre o utilizar objetos y métodos de un programa sin los privilegios del kernel del sistema operativo. Además, para evitar modificaciones por parte de los crackers de la red, implementa un método ultra seguro de autenticación por clave pública. El Cargador de Clases puede verificar una firma digital antes de realizar una instancia de un objeto. Por tanto, ningún objeto se crea y almacena en memoria, sin que se validen los privilegios de acceso. Es decir, la seguridad se integra en el momento de compilación, con el nivel de detalle y de privilegio que sea necesario.

1.4.1 Actores principales del mercado móvil.

Operador móvil (carrier celular):

Habilitar nuevos servicios y enriquecer experiencias.

Manejar la ganancia promedio por usuario (ARPU), fortalecer la retención del usuario.

Profesional IT (infraestructura):

1. Seguridad, confiabilidad, administrabilidad.

2. Integración con los bienes activos actuales y futuros de IT.

Desarrollador (generación de software):

Plataforma consistente.

Herramientas productivas y que resulten familiares.

Usuario final:

1. Interfaz de usuario familiar

2. Selección de usuario.

La plataforma tecnológica Java contempla y soluciona las necesidades de los tres últimos actores: Profesional, Desarrollador y Usuario Final. Elemento que lleva a utilizar esta herramienta para el desarrollo del trabajo.

1.4.2 JSR 212

Es un API (del inglés *application programming interface*) del lado del servidor para servicios móviles. Por servicios móviles se entiende mensajería, servicios de ubicación geográfica, servicios de presencia y administración centralizada de dispositivos. Estos servicios se pueden acceder desde extremos móviles Java ME, pero también desde fijos Java SE/EE.

Responde a la necesidad de un acceso unificado desde el lado del cliente a dichos servicios, a la vez que aumenta el ingreso-promedio-por-usuario (*ARPU*) con servicios del lado del servidor.

Se hace referencia a servicios del lado del servidor ya que existen plataformas, entre ellas CDC (Connected Device Configuration), con infraestructura sencilla basada en protocolos livianos y eficientes, como sockets. JSR 212 se refiere a los servicios montados en el extremo fijo, maquinas permanentes e infraestructura convencional de red, que atienden clientes móviles. Al mismo tiempo, algunos fabricantes han reconocido la necesidad de dar respuesta competitiva a la plataforma .NET, entre ellos la corporación Nokia, y por lo tanto han llevado adelante este JCP (Java Community Process).

El objetivo principal de **JSR 212** es entonces definir un acceso unificado a los servicios, con dos metas en orden de prioridad:

1. Especificar un núcleo de funcionalidad común de API, que será utilizado por todos los servicios SAMS.
2. Especificar la API de mensajería SMS y MMS.

Para ello, se define un modelo o estándar que debe ser respetado por todos los JSR relacionados con acceso al servidor desde clientes móviles. En este sentido, SAMS es un contenedor para futuros JSR relacionados.

Algunas de las funcionalidades que ofrece este sistema son:

1. Alto nivel de abstracción, fácil de utilizar y aplicar.
2. La API proporciona una abstracción del protocolo subyacente ejecución.
3. La separación de la API y la implementación del protocolo.
4. Portabilidad.
5. Desfragmentación.
6. Modelo de programación uniforme.
7. La API no está a favor de un protocolo específico, el protocolo aplicación, los elementos de red o proveedor.
8. La API proporciona un entorno de programación estable a los desarrolladores de aplicaciones (5).

1.4.3 Framework utilizados

Spring: Es un framework contenedor liviano basado en la técnica Inversión de Control (IOC) y una implementación de desarrollo según el paradigma de Orientación a Aspectos (AOP).

Framework: porque define la forma de desarrollar aplicaciones J2EE, dando soporte y simplificando complejidad propia del software corporativo.

Inversión de Control (IOC): promueve el bajo acoplamiento a partir de la inyección de dependencias (DI) entre los objetos (relaciones).

Orientación a Aspectos (AOP): presenta una estructura simplificada para el desarrollo y utilización de aspectos (módulos múltiple objeto crosscutting).

Diseñado en módulos, con funcionalidades específicas y consistentes con otros módulos, facilita el desarrollo de funcionalidades específicas y hace que la curva de aprendizaje sea favorable para el desarrollador. Dentro de las ventajas que ofrece Spring, encontramos que facilita la manipulación de los objetos que se usan EJBs o no, reduce la proliferación de Singletons, elimina la necesidad de usar

distintos y variados tipos de ficheros de configuración, mejora la práctica de programación, permite el uso o no de EJBs, realizando el mismo tipo de funciones sin ellos.

A pesar de que Spring no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al considerársele una alternativa y sustituta del modelo de Enterprise JavaBean. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria (6).

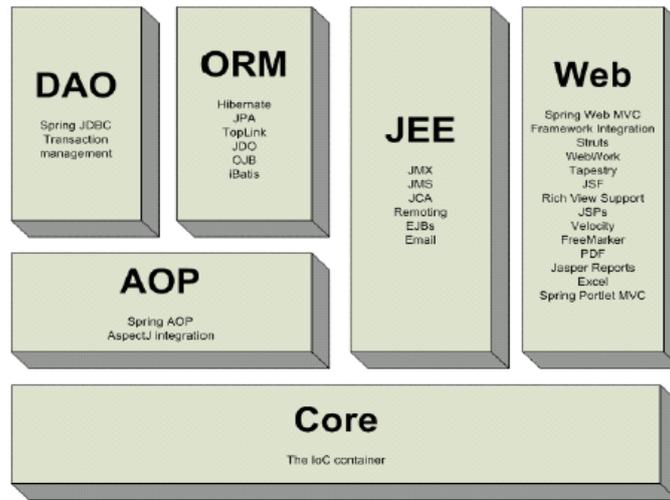


Figura 5 Arquitectura del Framework Spring.

DAO	accesos JDBC con manejo de transacciones (desde el módulo AOP).
ORM	integración con Hibernate, JDO, etc.
JEE	acceso e interacción con servicios enterprise.
Web	propone un contexto apropiado para el desarrollo de aplicaciones web e integración con otros frameworks (Struts, JSF, Tapestry, etc).

Tabla 1 Descripción de los elementos del framework.

Otro de los framework que se utilizará será Hibernate el cual presenta las siguientes características:

Hibernate: Hibernate es una herramienta para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación. Hibernate es una herramienta ORM completa que ha conseguido en un tiempo record una excelente reputación en la comunidad de desarrollo posicionándose claramente como el producto OpenSource líder en este campo gracias a sus prestaciones (7).

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Proporciona además un lenguaje para el manejo de consultas a la base de datos. Este lenguaje es similar a SQL y es utilizado para obtener objetos de la base de datos según las condiciones especificadas en el HQL (Hibernate Query Language). El uso de HQL permite usar un lenguaje intermedio que según la base de datos que se use y el dialecto que se especifique será traducido al SQL dependiente de cada base de datos de forma automática y transparente.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma.

Java Persistence API (JPA): proporciona un modelo de persistencia basado en POJO's para mapear bases de datos relacionales en Java. El Java Persistence API fue desarrollado por el grupo de expertos de EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones de escritorio; incluso fuera de la plataforma Java EE, por ejemplo, en aplicaciones Java SE.

En su definición, se han combinado ideas y conceptos de los principales frameworks de persistencia como Hibernate, Toplink y JDO, y de las versiones anteriores de EJB. Todos estos cuentan actualmente con una implementación JPA (7).

1.5 Gestor de Base de Datos.

1.5.1 Gestor de Base de Datos PostGreSQL.

Se utilizará PostGreSQL, este es un sistema de gestión basado en el proyecto POSTGRES. PostGreSQL es una derivación libre (OpenSource) de este proyecto, y utiliza el lenguaje SQL92/SQL99, este cuenta con características que lo distinguen por encima de los otros gestores como (8):

1. Implementación del estándar SQL92/SQL99.
2. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
3. Incorpora una estructura de datos array.
4. Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
5. Permite la declaración de funciones propias, así como la definición de disparadores.
6. Soporta el uso de índices, reglas y vistas.
7. Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
8. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
9. Compatibilidad entre ediciones

El proyecto Cubacel establece como política el uso de PostGreSQL 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.

1.6 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 óptimas o no dependiendo de las características del proyecto. Una de las metodologías de desarrollo de software más utilizadas en la actualidad es el RUP ya que posee características muy fuertes.

1.6.1 Características esenciales

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está **dirigido por los Casos de Uso**, está **centrado en la arquitectura**, y es **iterativo e incremental**.

Casos de Uso: no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo.

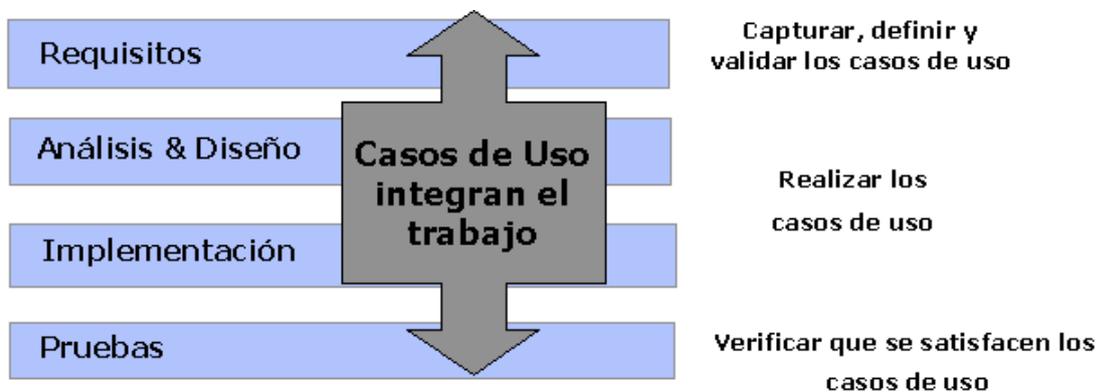


Figura 6 Los Casos de Uso integran el trabajo

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Proceso centrado en la arquitectura: La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además, la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma

software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema. Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para RUP, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura, el cual recibe este nombre porque lo forman las vistas lógicas, de implementación, de proceso y de despliegue, más la de Casos de Uso que es la que da cohesión a todas.

Proceso iterativo e incremental: El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto (9).

1.7 Herramientas de desarrollo utilizadas.

En esta parte del epígrafe se fundamentan las herramientas de desarrollo utilizadas para la realización del sistema. Estas herramientas fueron escogidas por las características que poseen y la ventaja que ofrecen por encima de las demás herramientas.

1.7.1 Entorno de desarrollo.

Un entorno de desarrollo integrado o IDE, es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

1.7.2 Entornos de desarrollo Eclipse y NetBeans.

Conceptos

NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software (10).

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.(11).

Características

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes para creación de proyectos, clases, tes, etc., y refactorización. Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversiones integración con Hibernate. También es un IDE universal: No solo es para java, también se puede usar con c, c++, cobol, D, Tex. Existen versiones para casi cualquier SO, Linux y Windows entre ellos.

El **IDE NetBeans** herramienta para programadores. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

Posee código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactorización.

Modularidad: Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

1.7.4 Visual Paradigm.

Poderosa herramienta CASE que al igual que el Rational Rose utiliza UML para el modelado, es la herramienta por excelencia para ser utilizada en múltiples plataformas. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual.

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Facilita licencias especiales para fines académicos

Entre sus características más significativas se puede resaltar que es multiplataforma, sus ediciones son compatibles y es de fácil uso para la creación de aplicaciones web (12).

1.8 Conclusiones.

En este capítulo se abordaron las principales tecnologías empleadas en la construcción del sistema a desarrollar, 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. Se realizó además un estudio del estado del arte.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción.

En este capítulo se presentan las características generales del sistema, se procede a realizar la captura de requisitos funcionales y no funcionales a los que se debe dar cumplimiento, se examinan los temas referentes a las necesidades de los clientes para obtener las bases y definir el objeto de automatización y la propuesta de solución que será desarrollada.

2.2 Problema y situación problemática.

Las comodidades y usos diversos que ya tienen en el mundo los teléfonos celulares no se habían expandido antes en el país por la carencia de una plataforma tecnológica que los garantizara. Por primera vez en Cuba se van a utilizar los teléfonos móviles con prestaciones que tienen que ver con los contenidos. La empresa de ETECSA se encuentra inmersa en la tarea de entrar a un mercado cada vez mayor en las telecomunicaciones y servicio de valor agregado para los celulares, acompañado a esto se encuentra el número creciente de usuarios de estos teléfonos celulares. Es por esto que se hace necesario el desarrollo de un sistema de Servicio de Mensajes Cortos que permita a los usuarios conocer informaciones culturales. Para esto es imprescindible implementar un sistema que permita, además de gestionar los eventos culturales que van a suceder, suscripción de usuarios tomando su número de teléfono para una comunicación del usuario con el sistema, obtener los datos y almacenarlos en una base de datos. El software debe estar disponible para todos los usuarios de telefonía móvil de ETECSA.

2.3 Objeto de Automatización

Serán objeto de automatización los procesos que intervienen en el servicio de envío de mensajes sobre información cultural a dispositivos móviles.

2.4 Propuesta del Sistema.

Se propone la creación de un sistema que añada un nuevo servicio de valor agregado (SVA) a la plataforma que se desarrolla actualmente en la entidad Cubacel. Se decidió realizar la implementación de un sistema que le permita a los usuarios de Cubacel conocer, mediante el envío de SMS, información actualizada sobre los eventos de cultura que están ocurriendo actualmente en el país, dentro de dichos eventos se encuentran Danza, Teatro, Música y Literatura.

2.5 Modelo De Dominio.

Como se ha venido explicando, en Cuba no existe un servicio de información cultural que brinde asistencia a los usuarios de teléfonos celulares, por lo que se propone el siguiente Modelo de Dominio donde se mostrarán los principales conceptos que ayudarán a obtener una visión de los conocimientos que se necesita para el desarrollo de la aplicación. Esto ayuda a los usuarios, clientes, desarrolladores e interesados en general, a utilizar un vocabulario común para poder entender el contexto en que se ubica el sistema, para hacer una captura correcta de los requisitos y poder construir una aplicación robusta que presente los beneficios que de ella se esperan.

La figura muestra los principales conceptos y sus relaciones, presentes en el modelo de dominio.

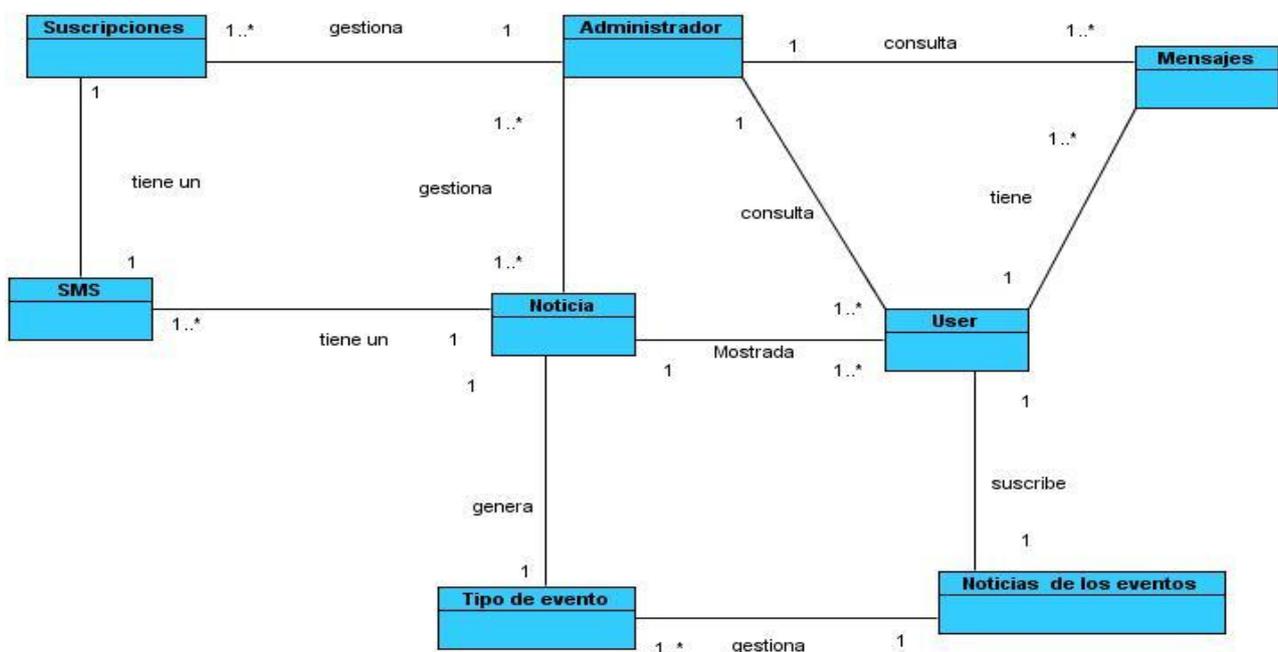


Figura 7 Modelo de Dominio.

2.5.1 Conceptos del Modelo de Dominio.

Descripción de los conceptos que fueron representados en el Modelo del Dominio, los cuales simbolizan objetos del mundo real que se encuentran dentro del dominio del negocio, con la necesidad de comprender las relaciones que se establecen entre las diferentes clases.

Administrador: Este rol gestiona todo lo referente a las informaciones culturales y las suscripciones, consulta además los mensajes enviados por los usuarios.

Eventos: Representa todos los eventos de cultura que acontecen y a los cuales los usuarios hacen las consultas.

Usuarios: Representa todos los usuarios que se inscriben en el servicio, y cada uno envía mensajes.

Suscripción de usuarios: Representa todas las suscripciones de los usuarios al servicio que se brinda de información cultural.

Mensajes: representa todos los mensajes que son creados para enviar a los usuarios como los enviados por los usuarios.

2.6 Requerimientos del Sistema.

Un requerimiento es una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. Los requisitos se pueden clasificar en: funcionales y no funcionales.

Requerimientos Funcionales: Son capacidades o condiciones que el sistema debe cumplir.

Requerimientos No Funcionales: Son propiedades o cualidades que el producto debe tener.

2.6.1 Requerimientos Funcionales.

R1: Registrar Noticias del Evento.

1.1 Adicionar noticias sobre los eventos al sistema.

R2: Suscribirse Usuarios.

2.1 Adicionar usuarios al sistema.

R3: Mostrar Usuarios.

3.1 Buscar todos los usuarios del sistema según el identificador.

R4: Mostrar mensajes.

4.1 Buscar todos los mensajes enviados por el usuario según un identificador.

R5: Mostrar noticias.

5.1 Mostrar las noticias según el tipo de evento.

R6: Enviar Noticia.

6.1 Enviar noticias a los usuarios sobre los eventos.

R7: Gestionar Suscripción.

7.1 Gestionar todo lo referente al mensaje que envía el usuario para suscribirse.

R8: Autenticar Usuario

8.1 Autenticar usuarios al sistema.

2.6.2 Requerimientos No Funcionales.

Software:

- Se necesita tener instalado el servidor web Apache Tomcat 6.0.
- Se necesita tener instalado en el cliente el flash player.
- La aplicación web debe ser compatible con varios navegadores, tales como el Mozilla Firefox y el Internet Explorer.

Hardware:

- Se necesita como requerimientos mínimos un servidor con procesador de 2.4 GHz Intel Pentium IV y con una memoria RAM de 1 GB.

Portabilidad:

- El sistema deberá ser portable para cualquier sistema operativo.

Usabilidad:

- La aplicación podrá ser manejada por cualquier usuario con conocimientos básicos del tema.

Soporte:

- Se desarrollará una documentación apropiada que describa todas las funcionalidades del sistema.

Seguridad:

- Permitir que el usuario se autentique en el sistema.
- El sistema debe proteger la integridad de la información y los contenidos. No permitir el acceso no autorizado al sistema.

2.7 Modelo de Casos de Uso del Sistema.

A continuación se presenta el modelo de casos de uso del sistema que comprende los actores del sistema, casos de uso que estos inicializan, el diagrama de casos de uso del sistema y una descripción detallada de los casos de uso.

2.7.1 Definición de los actores del sistema a automatizar.

Actores	Justificación
Administrador	Es la persona que gestiona todo lo relacionado con la información que se maneja.
Usuario	Es el móvil que interactúa con el sistema.

Tabla 2 Definición de Actores

2.7.2 Listado de Casos de Uso del sistema.

Los casos de uso son artefactos narrativos que describen el comportamiento del sistema desde el punto de vista del usuario. Establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. A continuación se muestran los casos de usos del sistema teniendo en cuenta los requisitos funcionales antes identificados.

CUS 1. Mostrar Noticias.

CUS 2. Registrar Noticia.

CUS 3. Mostrar Usuario.

CUS 4. Mostrar Mensajes.

CUS 5. Gestionar Suscripción.

CUS 6. Enviar noticia.

CUS 7. Suscribirse Usuario.

CUS 8: Autenticar Usuario.

2.7.3 Diagrama de Casos de Uso del sistema (CUS).

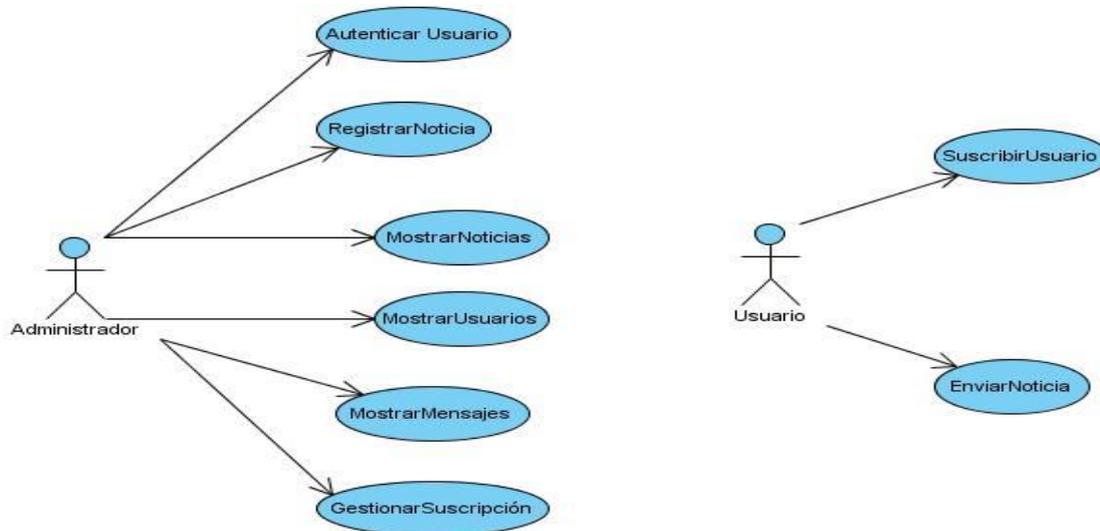


Figura 8 Diagrama de Casos de Uso del Sistema.

2.7.4 Descripción de los Casos de Uso del sistema.

CUS 1.Mostrar Noticias.

Caso de Uso:	Mostrar Noticias.
Actores:	Administrador.
Resumen:	Muestra las noticias existentes en el sistema según el tipo de evento.
Referencias	R6,R7
Precondiciones:	No Aplica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción listar noticia. 3. El administrador selecciona el tipo de evento del cual quiere ver las noticias almacenadas.	2. El sistema le muestra un interfaz con los tipos de eventos que tiene el sistema para que este seleccione un tipo de evento. 4. El sistema busca en la base de datos todas las noticias relacionadas con el evento. 5. Muestra una interfaz con todas las noticias relacionadas al tipo de evento.

Flujos Alternos2a “Evento no seleccionado”	
Acción del Actor	Respuesta del Sistema
2a1 El usuario da aceptar sin seleccionar el evento	2a2 El sistema muestra una ventana diciendo que seleccione un tipo de evento.
Flujos Alternos3a “Evento sin noticias almacenadas”	
	3a1 Muestra un mensaje al usuario informando de que el tipo de evento seleccionado no presenta noticias almacenadas.
Poscondiciones	No Aplica
Prioridad	Baja

Tabla 3 Descripción del CUS_MostrarNoticia.

CUS 2.Registrar Noticia.

Caso de Uso:	Registrar Noticia
Actores:	Administrador
Resumen:	El administrador registra las noticias en el sistema según el tipo de evento al cual pertenece.
Referencias	R1.
Precondiciones:	No Aplica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El administrador selecciona la opción “Registrar Noticia”.</p> <p>3. El administrador selecciona el tipo de evento al cual va a adicionar la noticia.</p> <p>4. El administrador introduce los siguientes datos de la noticia:</p> <ul style="list-style-type: none"> • Texto de la noticia. 	2. El sistema muestra una interfaz con los tipos de eventos.

2. El Administrador selecciona la opción "Buscar".	5. El sistema busca en la base de datos todos los usuarios que cumplan con los criterios introducidos. 6. El sistema muestra en la interfaz "Buscar Usuarios" los resultados obtenidos. 7. El sistema habilita la opción "Mostrar Mensajes de Usuario".
Flujos Alternos4a " Criterio de búsqueda no seleccionado"	
Acción del Actor	Respuesta del Sistema
4a1: El usuario selecciona la opción buscar sin seleccionar un intervalo de fecha.	4a2: El sistema muestra una ventana diciendo que selecciona algún criterio de búsqueda.
Poscondiciones	No Aplica
Prioridad	Media

Tabla 5 Descripción del CUS_Mostrar Usuarios.

CUS 4.Mostrar Mensajes.

Caso de Uso:	Mostrar Mensajes.
Actores:	Administrador
Resumen:	Se muestran los mensajes enviados por un usuario determinado.
Referencias	R5
Precondiciones:	Que se haya ejecutado antes el CU Mostar Usuarios.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción Mostrar Mensajes de la interfaz "Buscar Usuarios".	2. El sistema obtiene de la BD todos los mensajes relacionados con el número de teléfono del usuario seleccionado. 3. Muestra la interfaz "Mensajes de Usuario" con los resultados obtenidos.
Flujos Alternos 4a"Usuario no seleccionado"	

Acción del Actor		Respuesta del Sistema
		4a1: El sistema envía un mensaje de error "Debe seleccionar algún usuario".
Poscondiciones	No Aplica	
Prioridad	Media.	

Tabla 6 Descripción del CUS_MostrarMensajes.

CUS 5.Gestionar Suscripción.

Caso de Uso:	Gestionar Suscripción.	
Actores:	Administrador	
Resumen:	Se registra, modifica o elimina la forma de los usuarios suscribirse a un evento determinado.	
Referencias	R4	
Precondiciones:	No Aplica	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
<p>1. El Administrador selecciona una de las siguientes opciones de la interfaz Principal.</p> <ul style="list-style-type: none"> • Registrar Forma de Suscripción a Evento. • Modificar Forma de Suscripción a Evento. • Eliminar Forma de Suscripción a Evento 		<p>2. Si selecciona la opción:</p> <ul style="list-style-type: none"> • Registrar Forma de Suscripción a Evento, ir a la sección "Registrar Suscripción". • Modificar Forma de Suscripción a Evento, ir a la sección "Modificar Suscripción". • Eliminar Forma de Suscripción a Evento, ir a la sección "Eliminar Suscripción".
Sección "Registrar Suscripción"		
Acción del Actor		Respuesta del Sistema
<p>4. El Administrador introduce los datos solicitados.</p> <p>5. El Administrador selecciona la opción "Aceptar".</p>		<p>3. El sistema muestra la interfaz "Registrar Suscripción" donde solicita los siguientes datos de la suscripción:</p> <ul style="list-style-type: none"> • Tipo de evento al cual está asociada. • SMS para suscribirse. • Tiempo de suscripción.

	<p>6. El sistema verifica que los campos no estén vacíos.</p> <p>7. El sistema registra los datos en la BD.</p>
Flujo Alterno 6a "Campos vacíos"	
Acción del Actor	Respuesta del Sistema
	<p>6a1. El sistema muestra al usuario un mensaje de error indicando que todos los campos deben estar llenos.</p> <p>Ir a la acción 4.</p>
Sección "Modificar Suscripción"	
Acción del Actor	Respuesta del Sistema
<p>4. El Administrador selecciona el tipo de evento del cual quiere modificar la forma de suscribirse.</p> <p>7. El Administrador modifica los datos deseados.</p> <p>8. El Administrador selecciona la opción "Modificar".</p>	<p>3. El sistema muestra la interfaz "Buscar Suscripción" con los tipos de eventos.</p> <p>5. El sistema obtiene de la BD los datos de la suscripción del evento seleccionado.</p> <p>6. El sistema muestra los resultados obtenidos en la interfaz "Registrar Suscripción".</p> <p>9. El sistema verifica que los campos no estén vacíos.</p> <p>10. El sistema modifica los datos de la suscripción a partir del tipo de evento.</p>
Flujo Alterno 8a "Campos vacíos"	
Acción del Actor	Respuesta del Sistema
	<p>8a1. El sistema muestra al usuario un mensaje de error indicando que todos los campos deben estar llenos.</p> <p>Ir a la acción 6.</p>
Sección "Eliminar Suscripción"	

Acción del Actor		Respuesta del Sistema
<p>4. El Administrador selecciona el tipo de evento del cual quiere eliminar la forma de suscribirse.</p> <p>5. El Administrador selecciona la opción "Eliminar".</p>		<p>3. El sistema muestra la interfaz "Buscar Suscripción" con los tipos de eventos.</p> <p>6. El sistema elimina de la BD la suscripción del evento seleccionado.</p>
Poscondiciones	Se registra, modifica o elimina una instancia de suscripción.	
Prioridad	Crítica.	

Tabla 7 Descripción del CUS_ GestionarSuscripción.

CUS 6. Suscribir Usuario.

Caso de Uso:	Suscribir Usuario	
Actores:	Usuario.	
Resumen:	El usuario se suscribe al sistema para recibir el servicio de información cultural.	
Referencias	R2	
Precondiciones:	No Aplica	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El Usuario envía un mensaje.		<p>2. El sistema recibe el mensaje enviado.</p> <p>3. Obtiene de la BD la suscripción de evento asociada al tipo de mensaje enviado.</p> <p>4. El sistema almacena el número de teléfono del usuario.</p> <p>5. El sistema suscribe el usuario al evento usando el número del usuario y el tipo de evento.</p> <p>6. El sistema almacena el mensaje enviado por el usuario.</p> <p>7. El sistema envía al usuario las noticias relacionadas con el tipo de evento al cual se suscribió.</p>
Flujos Alternos3a "El usuario envía un mensaje incorrecto"		

Acción del Actor		Respuesta del Sistema
		3a1: El sistema almacena el número de teléfono. 3a2: El sistema almacena el mensaje del usuario.
Poscondiciones	Se registran en la BD instancias de Suscripción de usuario, Usuario, y Mensaje.	
Prioridad	Crítica.	

Tabla 8 Descripción del CUS_SuscribirUsuario.

CUS7. Enviar Noticia.

Caso de Uso:	Enviar Noticia.	
Actores:	Usuario	
Resumen:	Se le envía al usuario las noticias solicitadas.	
Referencias	R6	
Precondiciones:	No Aplica	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El Usuario envía un mensaje.		2. El sistema recibe el mensaje enviado. 3. Obtiene de la BD la noticia asociada al tipo de mensaje enviado. 4. El sistema almacena el número de teléfono del usuario. 5. El sistema almacena el mensaje enviado por el usuario. 6. El sistema envía al usuario la noticia obtenida a partir del mensaje recibido.
Flujos Alternos1a "El usuario envía la palabra incorrecta"		
Acción del Actor		Respuesta del Sistema
		1a1: El sistema almacena el mensaje incorrecto del usuario por si él hace luego algún redamo.
Poscondiciones	No Aplica.	
Prioridad	Crítica.	

Tabla 9 Descripción del CUS_EnviarNoticia.

2.8 Conclusiones

En este capítulo se realizó una descripción de la solución propuesta, se representó el modelo de dominio con los principales objetos del sistema, así como los requisitos funcionales y no funcionales. Se realizó el diagrama de casos de uso del sistema, así como la descripción detallada de estos.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

El objetivo principal de este capítulo es realizar la propuesta de diseño del sistema, obteniendo como resultado los artefactos más importantes para modelar su funcionamiento. Se utilizan patrones de diseño que aportan soluciones concretas a problemas específicos, para lograr un diseño que cumpla con las características que debe tener el sistema.

3.2 Modelo de Análisis.

El modelo de análisis ofrece un poder expresivo y una formalización para describir los aspectos del sistema, proporciona una estructura centrada en el mantenimiento, específicamente la flexibilidad ante los cambios y la reutilización. El modelo de análisis puede considerarse una primera aproximación al modelo de diseño aunque es un modelo por sí mismo. Es importante también hacer notar que en el modelo de análisis se hacen abstracciones para evitar resolver algunos problemas que es mejor posponer al diseño y a la implementación.

Este modelo contiene clases y sus objetos organizados en paquetes que colaboran. Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Se inserta además la realización de casos de uso del análisis que describe cómo se lleva a cabo y se ejecuta un caso de uso determinado en término de las clases del análisis y de sus objetos en interacción. Contiene además los paquetes de análisis que organizan los artefactos del análisis en piezas manejables así como la descripción de la arquitectura (vista del modelo de análisis) que muestra los artefactos significativos para la arquitectura del sistema propuesto.

Para modelar el sistema se realizó un diagrama de clases de análisis por cada caso de uso presente en el dominio del problema. A continuación se muestra una representación de los diagramas de clases del análisis correspondiente a los casos de usos más importantes para el sistema. Los diagramas de los casos de uso restantes se encuentran en los anexos.

3.2.1 Diagramas de Clases de Análisis.

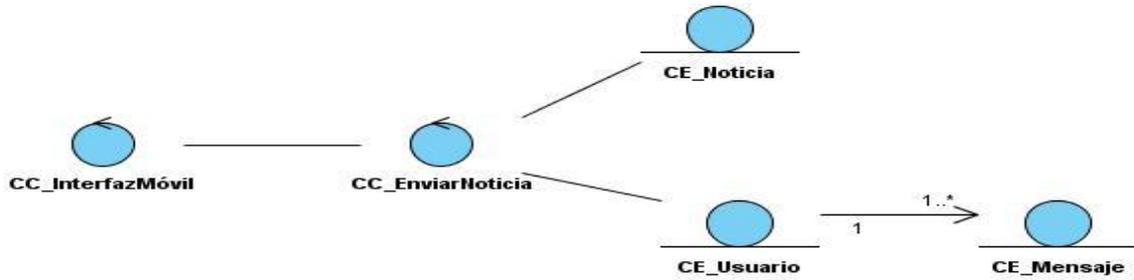


Figura 9 Diagrama de Clases de Análisis del Caso de Uso Enviar Noticias.

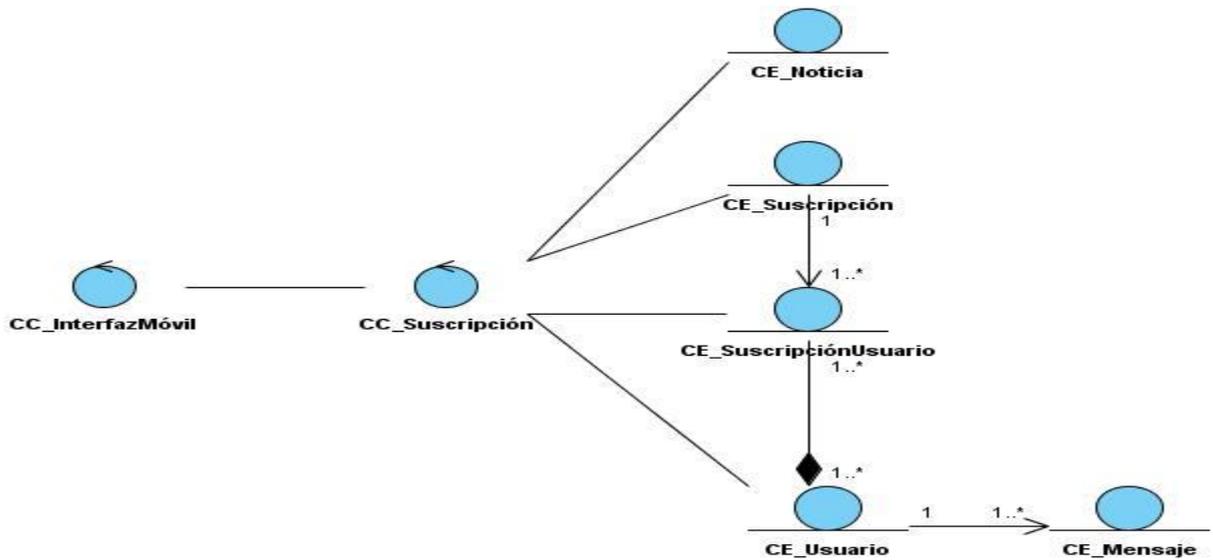


Figura 10 Diagrama de Clases de Análisis del Caso de Uso Suscribir Usuarios.

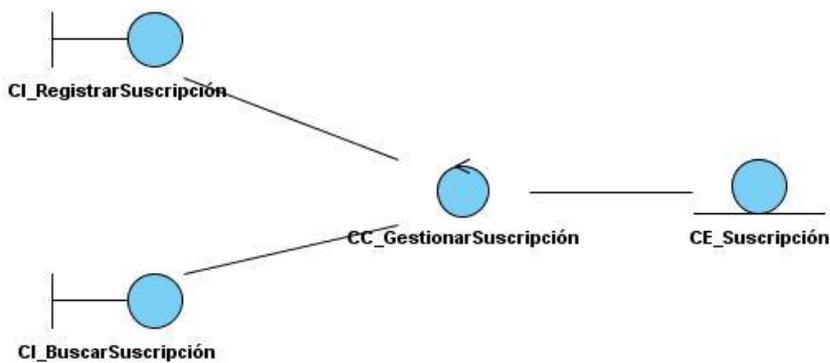


Figura 11 Diagrama de Clases de Análisis del Caso de Uso Gestionar Suscripción.

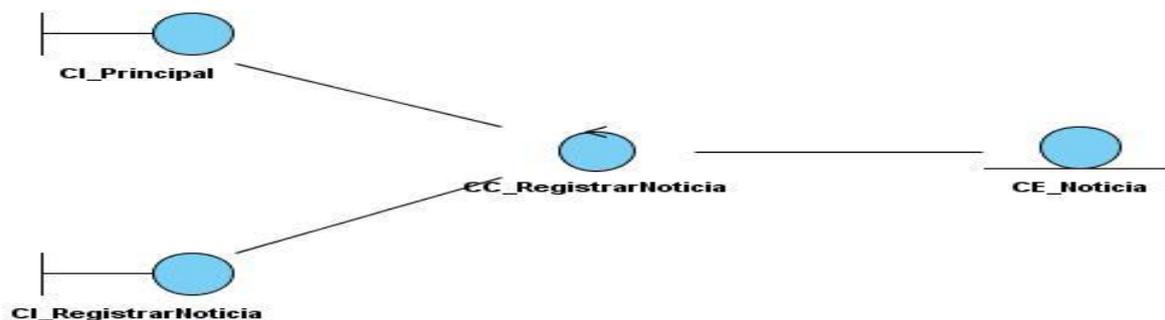


Figura 12 Diagrama de Clases de Análisis del Caso de Uso Registrar Noticia.

3.3 Modelo de Diseño.

El modelo de diseño, es un modelo de objetos que describe la realización física de los casos de uso, centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además el modelo de diseño sirve de abstracción de la implementación del sistema y es de ese modo utilizado como una entrada fundamental de las actividades de implementación. Los casos de uso son realizados por las clases de diseño y sus objetos.

El modelo de diseño está compuesto por los artefactos clases del diseño, realización caso de uso-diseño, diagramas de interacción (Colaboración y/o Secuencia), diagramas de clases, subsistemas del diseño, interfaz, descripción de la arquitectura (vista del modelo de diseño), modelo de despliegue, descripción de la arquitectura (vista del modelo de despliegue).

A continuación se muestra la realización de los Casos de Uso arquitectónicamente significativos los demás se adjuntan en los anexos.

3.3.1 Diagramas de Clases del Diseño.

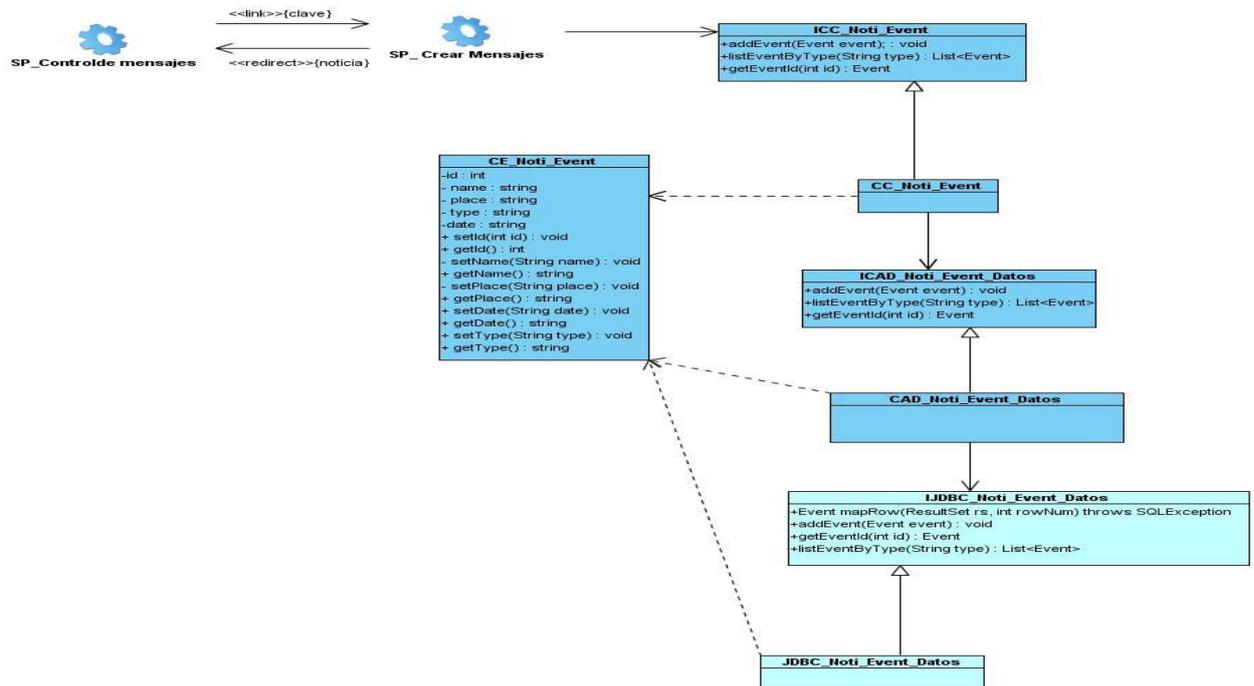


Figura 13 Diagrama de Clases de Diseño del Caso de Uso Enviar Noticia.

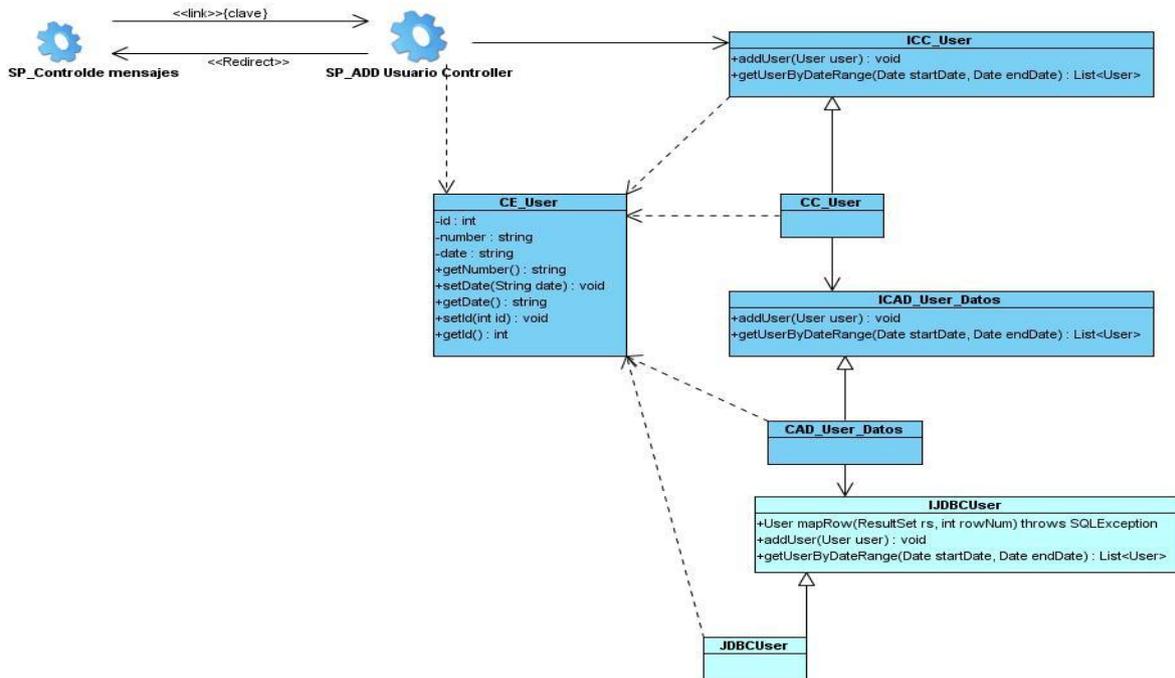


Figura 14 Diagrama de Clases de Diseño del Caso de Uso Suscribir Usuario.

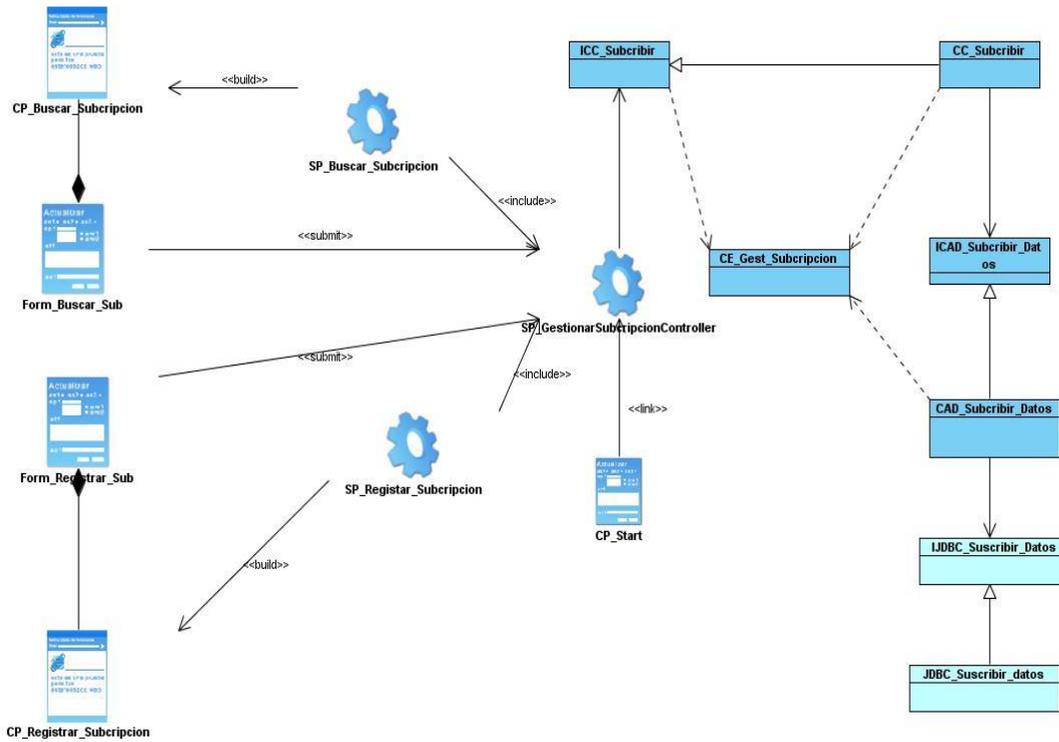


Figura 15 Diagrama de Clases de Diseño del Caso de Uso Gestionar Suscripción.

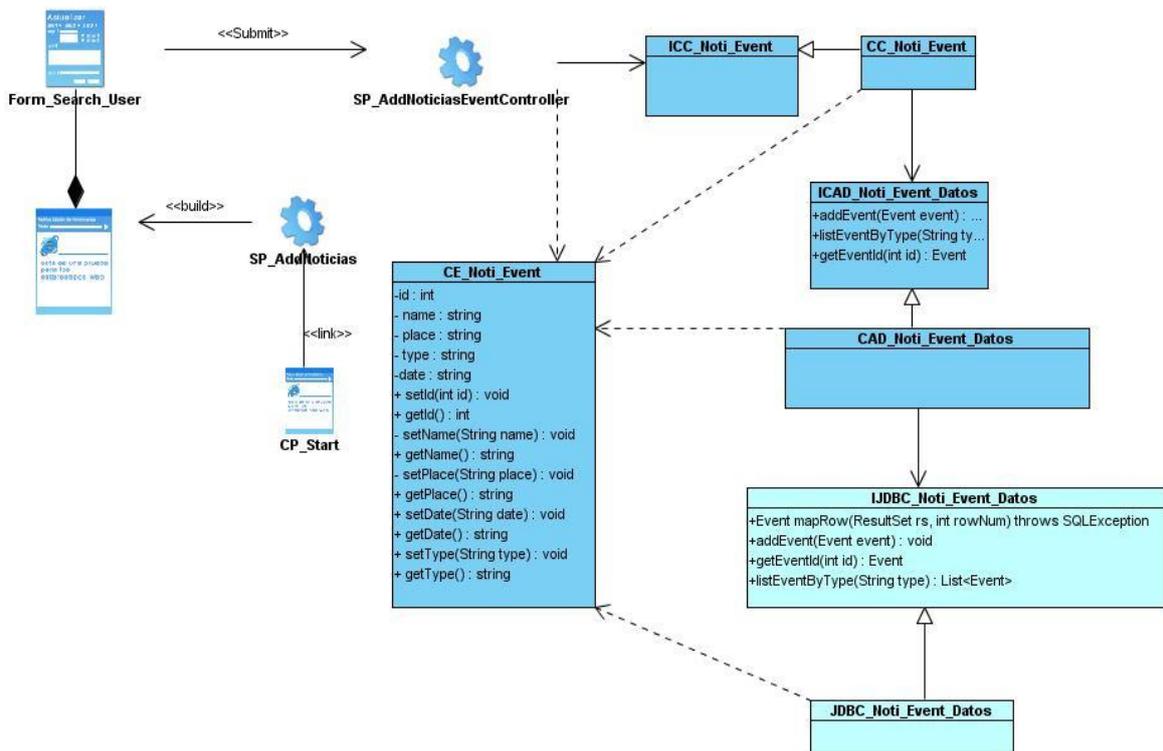


Figura 16 Diagrama de Clases de Diseño del Caso de Uso Registrar Noticia.

3.3.2 Diagramas de Secuencia

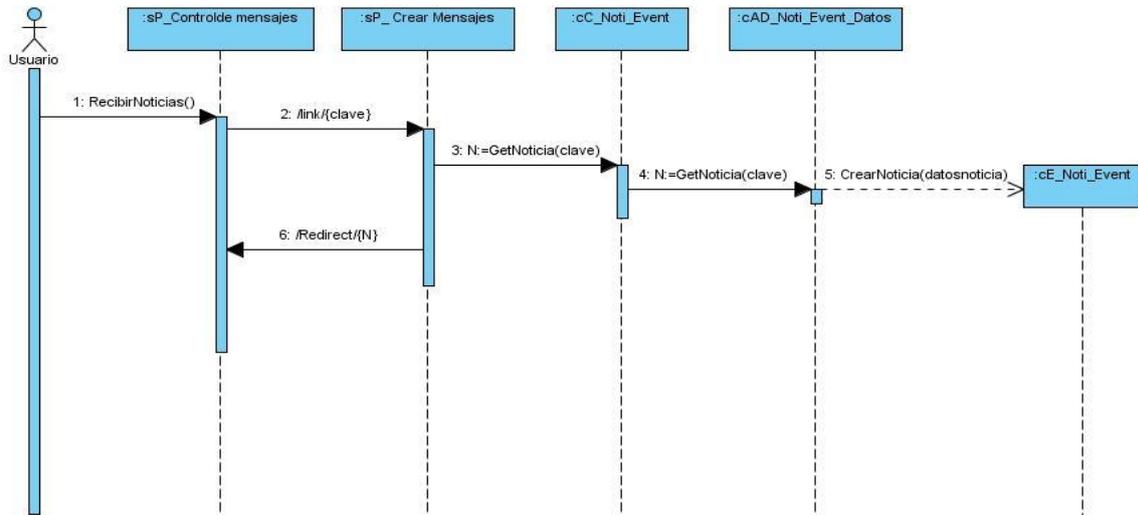


Figura 17 Diagrama de Secuencia del Caso de Uso Enviar Noticia.

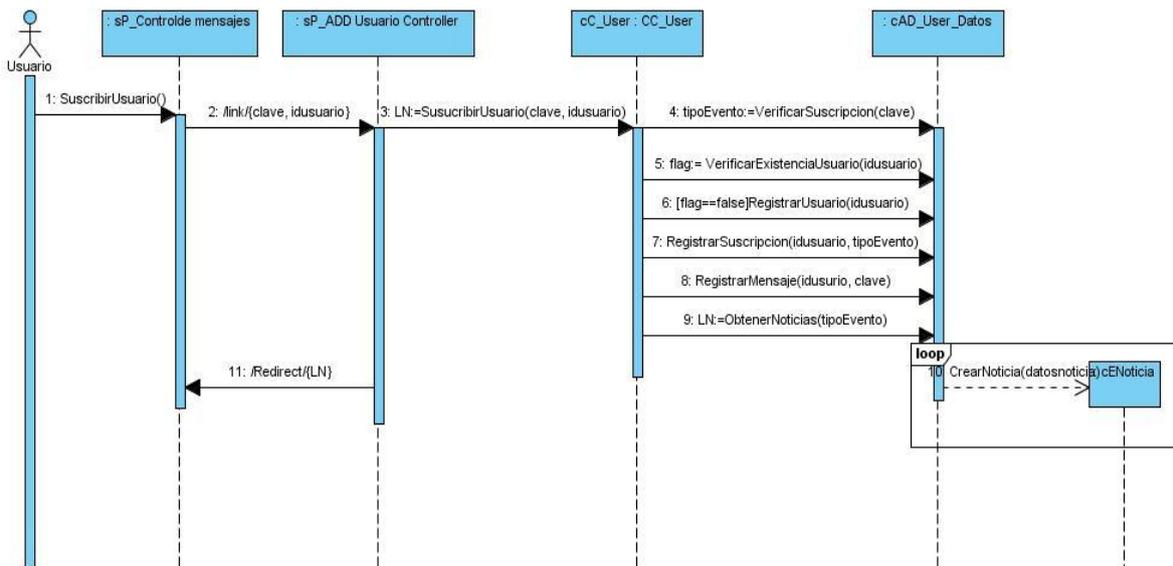


Figura 18 Diagrama de Secuencia del Caso de Uso Suscribir Usuarios.

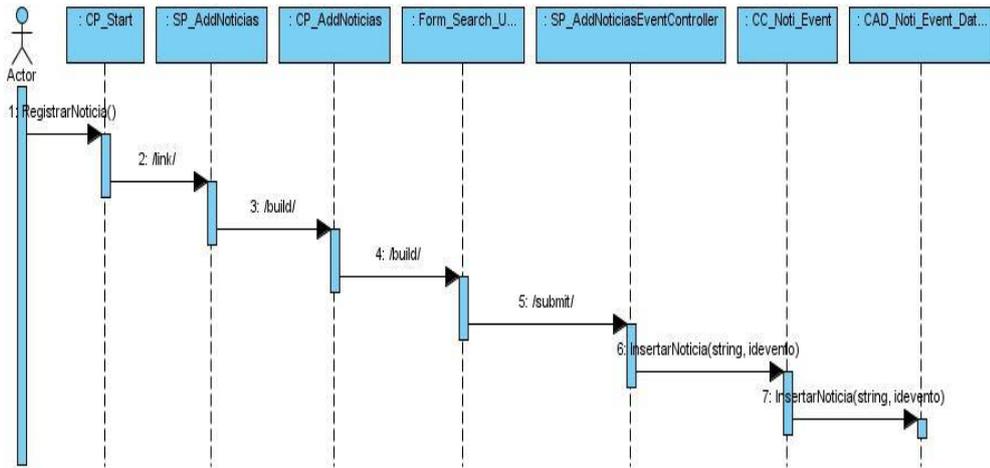


Figura 20 Diagrama de Secuencia del Caso de Uso Registrar Noticia.

3.3.3 Modelo lógico de datos.

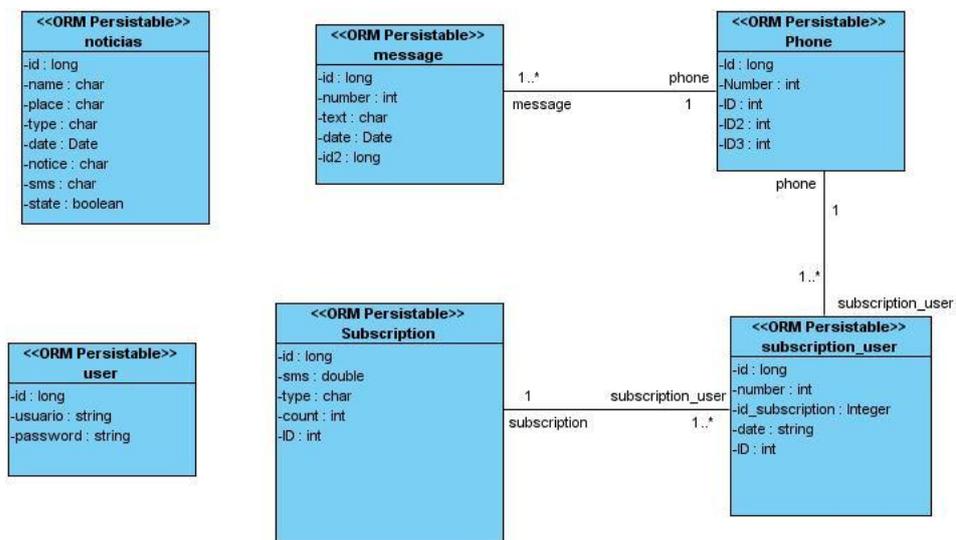


Figura 21 Diagrama de Clases Persistentes.

3.3.4 Modelo físico de datos (modelo de datos).

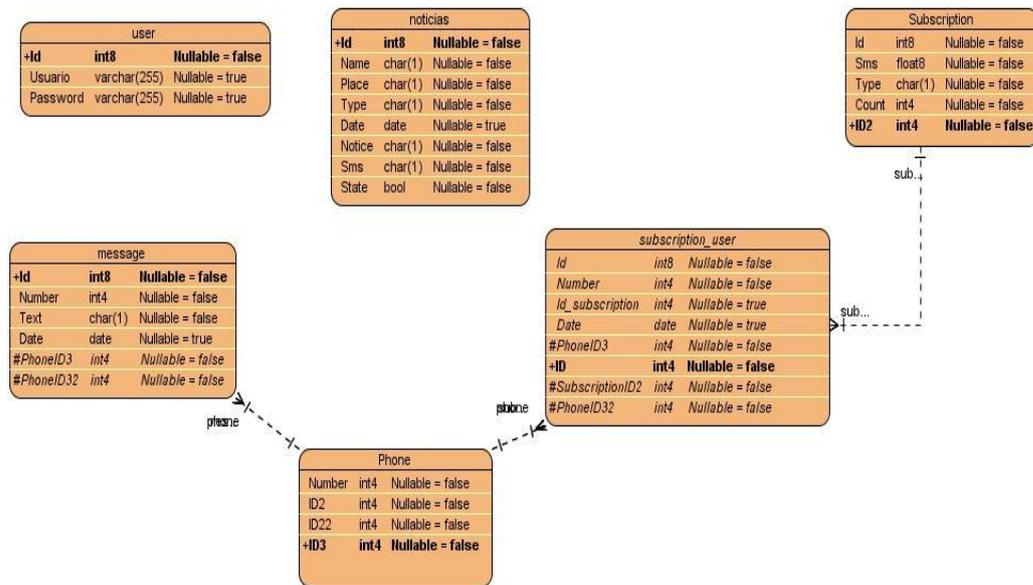


Figura 22 Diagrama Entidad-Relación.

3.4 Estilos Arquitectónicos.

Para la construcción de la aplicación se aplicaron dos estilos arquitectónicos el MVC y el Modelo 3 Capas, los cuales explican cómo está conformado el sistema arquitectónicamente. El patrón 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. El Modelo componente encargado del acceso a datos, la Vista define la interfaz de usuario y el Controlador responde a eventos y modifica la vista y el modelo.



Figura 23 Modelo Vista Controlador.

La arquitectura 3 capas separa las capas por funcionalidades independientes, una capa que servirá para el acceso a los datos, una capa para centralizar la lógica de negocio y por

último una interfaz gráfica que facilite al usuario el uso del sistema. A continuación, mostramos como está compuesta la arquitectura 3 Capas en el sistema.



Figura 24 Modelo 3 Capas del Sistema.

3.5 Patrones de Diseño.

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Un sistema bien estructurado está lleno de patrones.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular, además identifica Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades. Entre los patrones de diseño se encuentra el grupo de los GRASP (patrones generales de software para asignar responsabilidades), estos patrones se aplican al preparar los diagramas de interacción, cuando se asignan las responsabilidades o durante ambas actividades.

3.5.1 Tipo de Patrones Utilizados.

Patrones GRASP utilizados en el diseño del sistema.

Controlador: Consiste en asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente un manejador artificial de todos los

eventos del sistema de un caso de uso, generalmente denominados “Manejador” (controlador de casos de uso).

En el diseño propuesto se representa por cada caso de uso una clase controladora de las vistas para manejar todas las peticiones de los usuarios y una clase controladora del modelo para manejar todos los eventos del sistema.

Experto: Consiste en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Con la utilización de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento.

En el diseño propuesto se dividen las clases por capa, dándole a cada una de estas la responsabilidad de manejar un determinado tipo de información según la capa que representen.

Creador: Consiste en asignarle a la clase B la responsabilidad de crear una instancia de clase A en caso de que B contenga los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A). B es un creador de los objetos A.

Las clases propuestas de Acceso a los Datos contienen la información necesaria para crear las instancias de las clases del dominio, por lo que son expertas creando objetos de las clases entidades.

3.6 Conclusiones

En este capítulo se expusieron los principales artefactos del análisis y diseño del sistema propuesto como solución a la problemática planteada, principalmente los diagramas de clases e interacción del análisis y el diseño. Además, se describieron los patrones de diseño utilizados.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción.

En este capítulo se hace referencia a las disciplinas de implementación y prueba. Se describe el modelo de implementación utilizado y se muestran los diagramas de componente y de despliegue. Se realiza el modelo de prueba a través de los casos de prueba aplicados en el desarrollo del sistema propuesto. Además, se expone la distribución del sistema en nodos y los protocolos de comunicación entre cada uno de ellos. Se explica la trazabilidad de las clases del diseño a componentes de implementación mediante los diagramas de componentes del modelo de implementación.

4.2 Modelo de Implementación.

El resultado principal de la implementación es el modelo de implementación, el cual describe, al igual que los elementos del modelo de diseño, cómo las clases se implementan en términos de componentes, ficheros de código fuente, ejecutable, scripts, librerías, APIs, dentro del flujo de implementación que propone RUP. El modelo de implementación está conformado por los diagramas de despliegue y componentes, éstos describen los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará el sistema. El modelo de implementación es la entrada principal de las etapas de prueba que siguen a la implementación.

4.2.1 Diagrama de Componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre esos elementos. También se utilizan para mostrar las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros que son los resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados.

Además, es un grafo de componentes unidos a través de relaciones de dependencia que pueden ser de compilación o de ejecución. A continuación se muestran los diagramas de componentes más significativos generados durante el modelado del sistema los demás se encuentran en los Anexos.

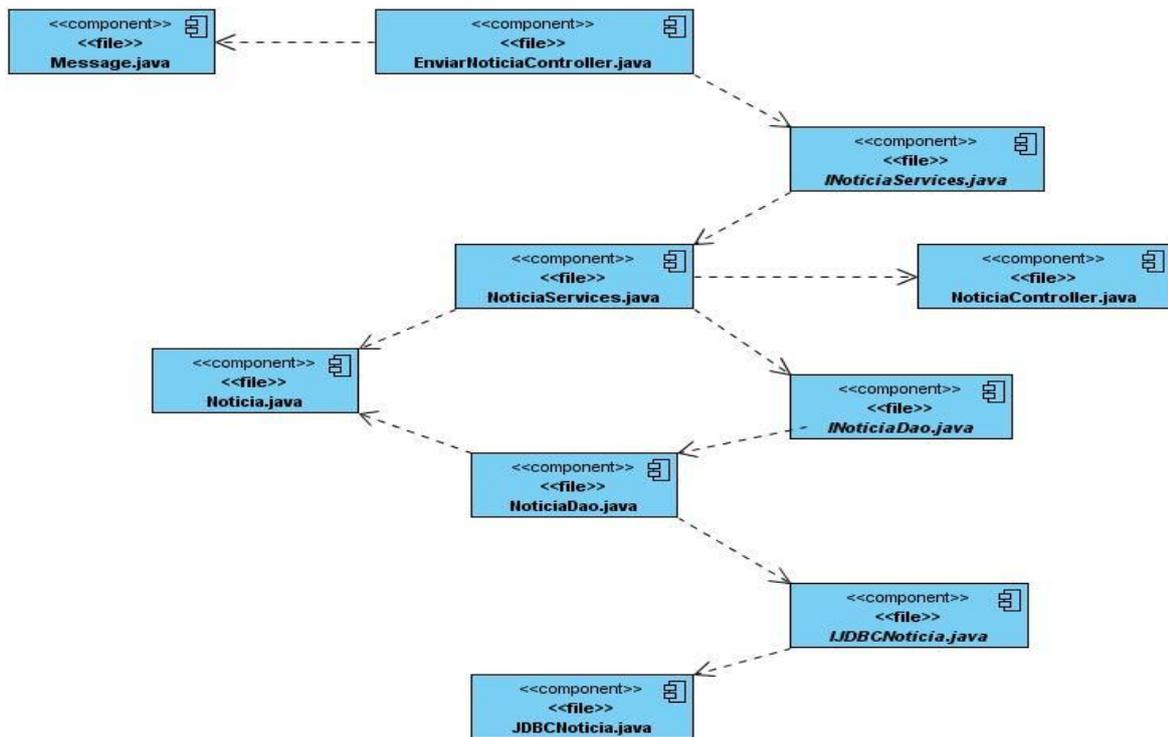


Figura 25 Diagrama de Componentes del CU_EnviarNoticia.

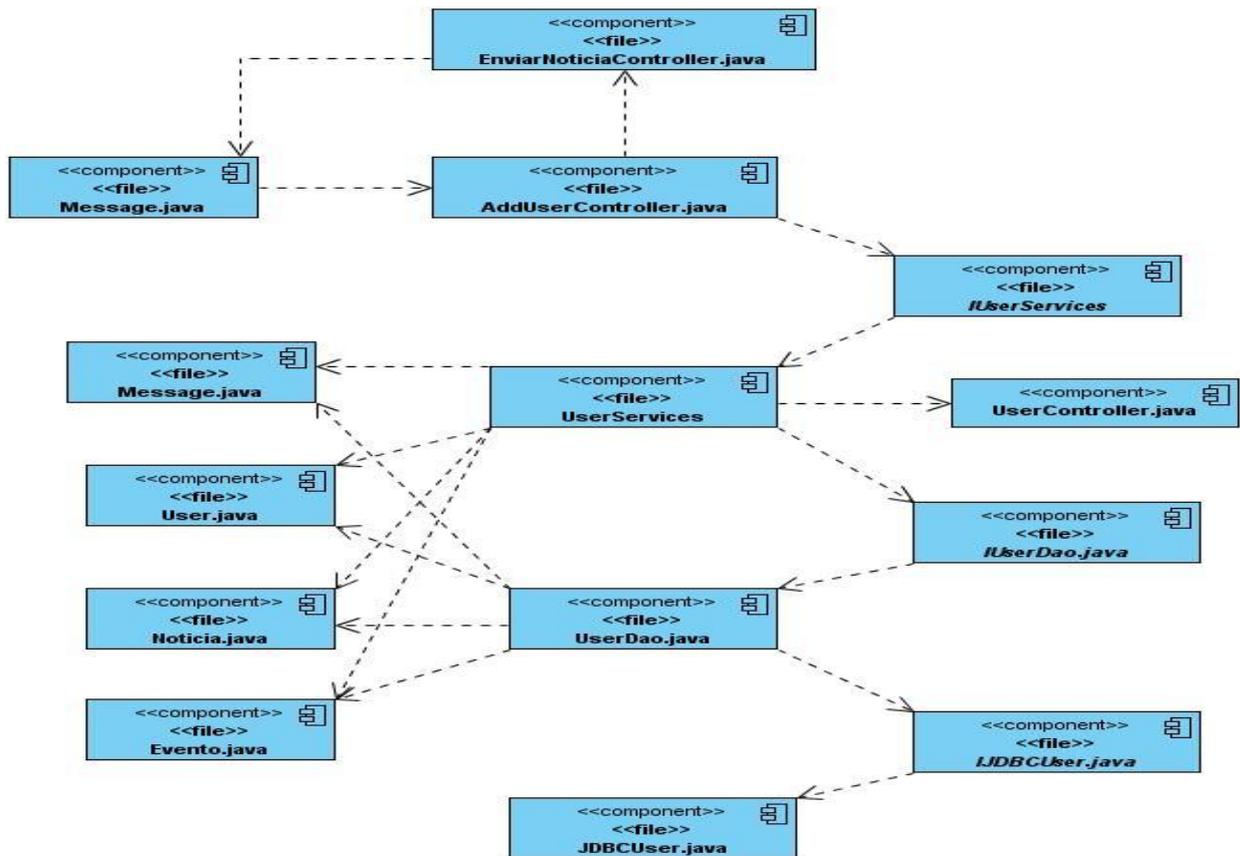


Figura 26 Diagrama de Componentes del CU_SuscribirUsuario.

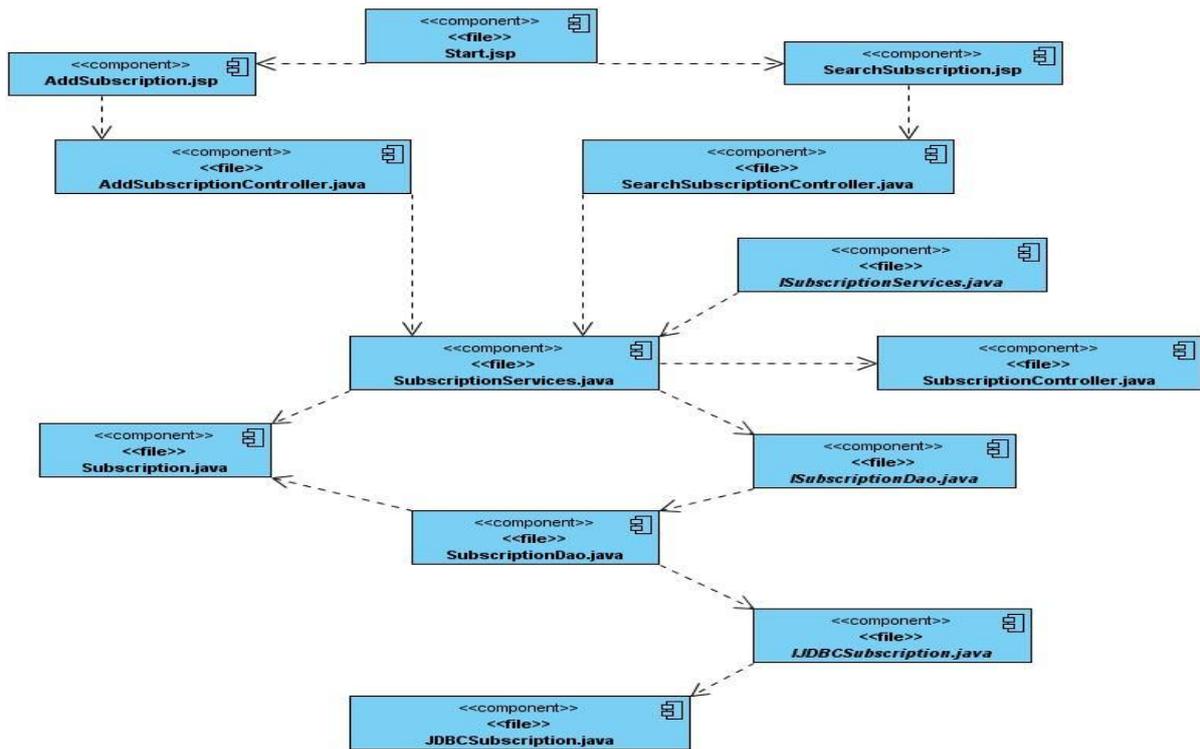


Figura 27 Diagrama de Componentes del CU_GestionarSuscripción.

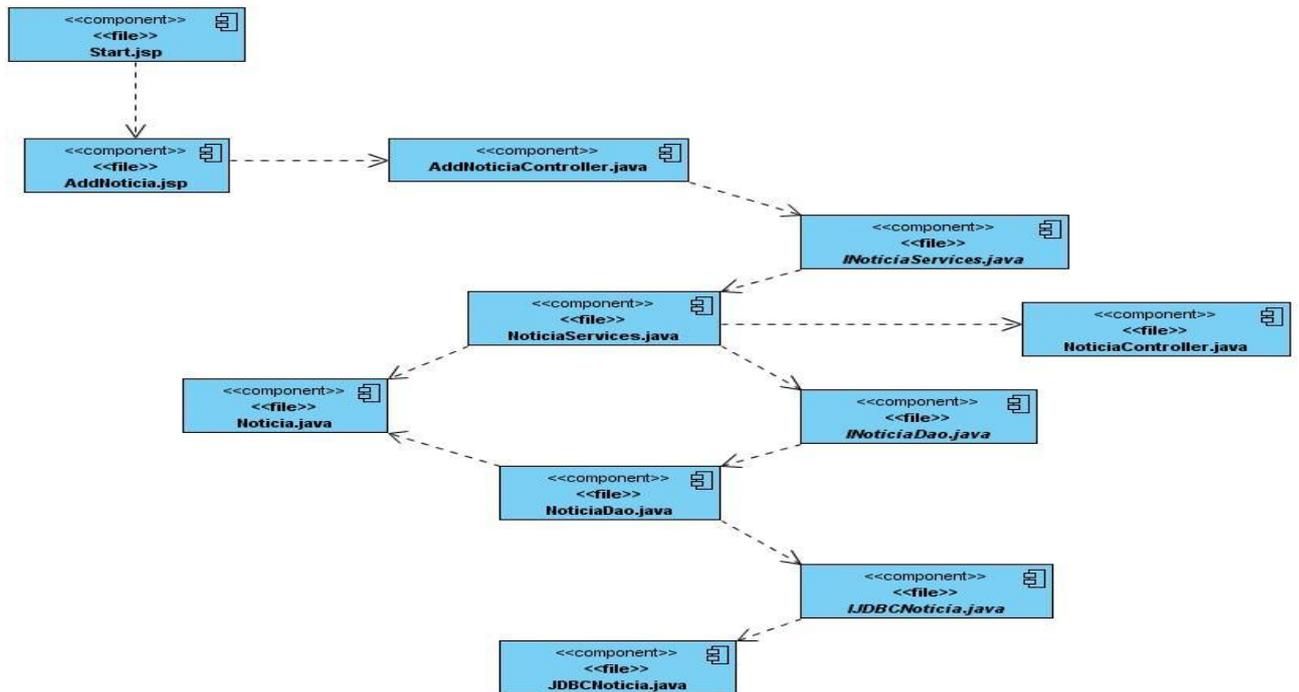


Figura 28 Diagrama de Componentes del CU_RegistrarNoticia.

4.2.2 Diagrama de Despliegue.

El diagrama de despliegue representa la distribución física del sistema. Es inicializado en el flujo de trabajo de análisis y diseño y perfeccionado en el de implementación. Describe la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software, y describe además la topología del sistema que no es más que la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Los nodos son objetos físicos que se dividen en dos tipos los que existen en tiempo de ejecución que representan algún tipo de recurso computacional (capacidad de memoria y procesamiento) y los dispositivos. A continuación se muestra el diagrama de despliegue del sistema.



Figura 29 Diagrama de Despliegue.

4.3 Modelo de Pruebas.

Durante la disciplina de prueba es objetivo verificar que el sistema cumpla con los requerimientos de los clientes y usuarios finales para lograr una mayor calidad, el principal objetivo es realizar y evaluar las pruebas como se describen en el modelo de pruebas. El modelo de pruebas describe principalmente cómo se prueban los componentes ejecutables en el modelo de implementación. El modelo de pruebas es una colección de casos de prueba, procedimientos de prueba y componentes de prueba.

4.3.1 Métodos de Prueba.

Prueba de Caja Negra: Pruebas que se llevan a cabo sobre la interfaz del software, se centra principalmente en los requisitos funcionales del software. Pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. Las pruebas de caja negra verifican las especificaciones funcionales y no consideran la estructura interna del programa, además se hacen sin el conocimiento interno del producto y no se validan funciones ocultas.

4.3.2 Diseño de Casos de Prueba.

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada y es a su vez la entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final. Los casos de pruebas verifican si el producto satisface los requerimientos del usuario y si se comporta como se desea. Para comprobar el funcionamiento del sistema se realizaron diferentes casos de prueba, a continuación se muestra un caso de prueba de caja negra.

Caso de Uso: Registrar Noticia

Descripción General: El administrador registra las noticias en el sistema según el tipo de evento al cual pertenece.

Precondiciones: No Aplica.

Secciones a probar en el Caso de Uso.

Nombre de la Sección	Escenarios de la Sección	Texto de la Noticia	SMS asociado	Tipo de Evento	Respuesta del Sistema	Resultado de la Prueba
SC 1: Registrar Noticia.	EC 1.1: Registrar Noticia.	V	V	V	El sistema registra la noticia en la base de datos.	
		I	I	I	El sistema muestra el mensaje de error "El mensaje enviado no contiene datos".	

Tabla 11 Caso de Prueba del CU_RegistrarNoticia.

Caso de Uso: Mostrar Noticia.

Descripción General: Muestra las noticias existentes en el sistema según el tipo de evento.

Precondiciones: No Aplica.

Secciones a probar en el Caso de Uso.

Nombre de la Sección	Escenarios de la Sección	Tipo de Evento	Respuesta del Sistema	Resultado de la Prueba
SC 1: Mostrar Noticia.	EC 1.1: Mostrar Noticia.	V	El sistema muestra la noticia del tipo de evento.	
	EC 1.2: Evento no seleccionado.	I	El sistema muestra una ventana diciendo que seleccione el tipo de evento.	
	EC 1.3: Evento sin noticias almacenadas	I	El sistema muestra el mensaje "El evento seleccionado no tiene noticias almacenadas".	

Tabla 12 Caso de Prueba del CU_MostrarNoticias.

Caso de Uso: Suscribir Usuario.

Descripción General: El usuario se suscribe al sistema para recibir el servicio de información cultural.

Precondiciones: No Aplica.

Secciones a probar en el Caso de Uso.

Nombre de la Sección.	Escenarios de la Sección.	Mensaje.	Respuesta del Sistema.	Resultado de la Prueba.
SC 1: Suscribir Usuarios.	EC 1.1: Suscribir Usuarios.	V	El sistema suscribe al usuario y le envía un mensaje con la noticia asociada al evento al cual se suscribió.	
	EC 1.2: El usuario envía un mensaje incorrecto.	I	El sistema almacena el número de teléfono y el mensaje.	

Tabla 13 Caso de Prueba del CU_SuscribirUsuario.

Caso de Uso: Mostrar Usuarios.

Descripción General: Muestra los usuarios existentes en el sistema según los criterios de búsqueda introducidos por el usuario.

Precondiciones: No Aplica.

Secciones a probar en el Caso de Uso.

Nombre de la Sección.	Escenarios de la Sección.	SMS enviado en un intervalo de fecha.	Suscritos en un intervalo de fecha determinado.	Suscritos a un tipo de evento determinado.	Número de teléfono.	Respuesta del Sistema.	Resultado de la Prueba.
SC 1: Mostrar Usuarios	EC 1.1: Mostrar Usuarios.	V	V	V	V	El sistema muestra un listado con los usuarios encontrados.	
	EC 1.2: Criterio de búsqueda no seleccionado.	I	I	I	I	El sistema muestra el mensaje "Tiene que seleccionar algún criterio de búsqueda".	

Tabla 14 Caso de Prueba del CU_MostrarUsuarios.

Caso de Uso: Enviar Noticia.

Descripción General: Se le envía al usuario las noticias solicitadas.

Precondiciones: No Aplica.

Secciones a probar en el Caso de Uso.

Nombre de la Sección.	Escenarios de la Sección.	Mensaje.	Respuesta del Sistema.	Resultado de la Prueba.
SC 1: Enviar Noticia.	EC 1.1: Enviar Noticia.	V	El sistema le envía la noticia al usuario del evento al cual se suscribió.	
	EC 1.2: El usuario envía la palabra incorrecta.	I	El sistema almacena el mensaje para si el usuario hace algún reclamo.	

Tabla 15 Caso de Prueba del CU_EnviarNoticia.

Caso de Uso: Mostrar Mensajes.

Descripción General: Se muestran los mensajes enviados por un usuario determinado.

Precondiciones: Que se haya ejecutado antes el CU Mostar Usuarios.

Secciones a probar en el Caso de Uso.

Nombre de la Sección.	Escenarios de la Sección.	Número de teléfono.	Respuesta del Sistema.	Resultado de la Prueba.
SC 1: Mostrar Mensajes.	EC 1.1: Mostrar Mensajes.	V	El sistema muestra un listado con todos los mensajes enviados por el usuario seleccionado.	
	EC 1.2: Usuario no seleccionado.	I	El sistema envía un mensaje de error "Debe seleccionar algún usuario".	

Tabla 16 Caso de Prueba del CU_MostrarMensajes.

Caso de Uso: Gestionar Suscripción.

Descripción General: Se registra, modifica o elimina la forma de los usuarios suscribirse a un evento determinado.

Precondiciones: No Aplica.

Secciones a probar en el Caso de Uso.

Nombre de la Sección.	Escenarios de la Sección.	Tipo de Evento	SMS para Suscribirse	Tiempo de Suscripción	Respuesta del Sistema.	Resultado de la Prueba.
SC 1: Registrar Suscripción.	EC 1.1: Registrar Suscripción.	V	V	V	El sistema registra la noticia correctamente en la en la BD.	
	EC 1.2: Campos Vacíos.	I	I	I	El sistema muestra al usuario un mensaje de error indicando que	

					todos los campos deben estar llenos.	
SC 2: Modificar Suscripción.	EC 2.1: Modificar Suscripción.	V	V	V	El sistema modifica los datos de la suscripción a partir del tipo de evento.	
	EC 2.2: Campos Vacíos.	I	I	I	El sistema muestra al usuario un mensaje de error indicando que todos los campos deben estar llenos.	
SC 3: Eliminar Suscripción.	EC 3.1: Eliminar Suscripción.	V	-	-	El sistema elimina de la BD la suscripción del evento seleccionado	

Tabla 17 Caso de Prueba del CU_GestionarSuscripción.

4.4 Conclusiones.

En este capítulo se mostraron los principales artefactos de las disciplinas de implementación y prueba, en específico los diagramas de despliegue y de componentes. Además de una descripción de los casos de prueba realizados al sistema utilizando el método de caja negra permitiendo de esta forma la detección de errores desde los primeros ciclos de desarrollo del proyecto.

CAPÍTULO 5: FACTIBILIDAD DEL SISTEMA

5.1 Introducción.

En el presente capítulo se realiza el cálculo de la estimación del tiempo de desarrollo y el análisis de la factibilidad del sistema con el objetivo de obtener una aproximación de los recursos necesarios así como del esfuerzo, costo y tiempo necesario. Para la realización de este análisis, se utiliza el método de Análisis de Puntos de Casos de Uso, es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

5.2 Método de estimación Puntos por Casos de Uso.

La estimación basada en puntos de casos de uso sigue tres pasos fundamentales los cuales son descritos a continuación.

Paso 1. Cálculo de los Puntos de casos de uso sin ajustar $UUCP = UAW + UUCW$

Donde:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

Paso 2. Cálculo de los Puntos de casos de uso ajustados $UCP = UUCP \times TCF \times EF$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso $E = UCP \times CF$

Donde:

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de Conversión.

E: Esfuerzo estimado en horas hombres.

Paso 4. Calcular el esfuerzo de todo el proyecto.

5.2.1 Cálculo de Puntos de Casos de Uso sin ajustar.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: puntos de casos de uso sin ajustar.

UAW: factor de peso de los actores sin ajustar.

UUCW: factor de peso de los casos de uso sin ajustar.

Para calcular factor de peso de los actores sin ajustar **UAW:**

Tipo	Descripción	Peso	Cant * peso:
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación. (API, Application Programming Interface).	1	1*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	1*3
Total:			4

Tabla 18 Cálculo del factor de peso de los actores sin ajustar.

Para calcular el factor de peso de los casos de uso sin ajustar **UUCW:**

No.	Nombre de Caso de Uso	Cantidad transacciones	Tipo
1	Mostrar Noticia.	1	Simple
2	Registrar Noticia.	1	Simple
3	Mostrar Usuarios.	1	Simple
4	Mostrar Mensajes.	1	Simple
5	Gestionar Suscripción	4	Medio
6	Suscribir Usuario	3	Simple
7	Enviar Noticia	2	Simple

Tabla 19 Cantidad de transacciones por casos de uso.

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	6*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10	1*10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	0*15
Total:			40

Tabla 20 Cálculo del factor de peso de los casos de uso sin ajustar.

La cantidad de transacciones se determina a partir de la descripción textual de los casos de uso. Entre más detallada esté la descripción textual, más transacciones se puede encontrar y la estimación será más exacta.

Luego:

$$UUCP = 4 + 40$$

$$UUCP = 44.$$

5.2.2 Cálculo de Puntos de Casos de Uso ajustados.

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: puntos de casos de uso ajustados.

UUCP: puntos de casos de uso sin ajustar.

TCF: factor de complejidad técnica.

EF: factor de ambiente.

Para calcular factor de complejidad técnica (**TCF**):

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i) \text{ (Donde } \textit{Valor} \text{ es un número del 0 al 5).}$$

Significado de los valores:

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.

3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

Factor	Descripción	Peso	Valor	Comentario	Σ (Peso_i * Valor_i)
T1	Sistema distribuido.	2	0	El sistema es centralizado.	0
T2	Objetivos de performance o tiempo de respuesta.	1	5	La velocidad no está limitada por las entradas de los usuarios.	5
T3	Eficiencia del usuario final.	1	2	Escasas restricciones de eficiencia.	2
T4	Procesamiento interno complejo.	1	0	No hay cálculos Complejos.	0
T5	El código debe ser reutilizable.	1	0	No se requiere que el código sea reutilizable	0
T6	Facilidad de instalación.	0.5	5	El sistema debe ser fácil de instalar.	2,5
T7	Facilidad de uso.	0.5	5	De ser un sistema amigable.	2,5
T8	Portabilidad.	2	0	No se requiere que el sistema sea portable.	0
T9	Facilidad de cambio.	1	5	Se requiere que sea un sistema flexible ante cambios.	5
T10	Concurrencia.	1	0	No hay concurrencia.	0
T11	Incluye objetivos especiales de seguridad.	1	2	Seguridad normal.	2
T12	Provee acceso directo a terceras partes.	1	5	Los usuarios web tienen acceso directo.	5
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	2	No se requieren facilidades especiales de entrenamiento a los usuarios.	2
Total:					26

Tabla 21 Cálculo del Factor de Complejidad Técnica.

$$TCF = 0.6 + 0.01 * 26$$

$$TCF = 0.86$$

Para Calcular EF

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor es un número del 0 al 5)}$$

Factor:	Descripción:	Peso:	Valor:	Comentario:	Σ (Peso_i * Valor_i):
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	2	Poca experiencia con el modelo de proyecto.	3
E2	Experiencia en la aplicación.	0.5	0	No hay mucha en la aplicación.	0
E3	Experiencia en orientación a objetos.	1	4	La mayoría del grupo ha programado Orientado a Objetos.	4
E4	Capacidad del analista líder.	0.5	3	Poca experiencia en actividad de análisis.	1.5
E5	Motivación.	1	3	Motivación normal.	3
E6	Estabilidad de los requerimientos.	2	3	Mas o menos estables.	6
E7	Personal part-time.	-1	0	Todo el equipo es full-time.	0
E8	Dificultad del lenguaje de programación.	-1	3	Se usará el lenguaje Java orientado a objetos, no estudiado. Aunque el lenguaje no es difícil de aprender.	-3
Total:					14.5

Tabla 22 Cálculo del Factor Ambiente.

$$EF = 1.4 - 0.03 * 14.5$$

$$EF = 0.965$$

Luego:

$$UCP = 44 * 0.86 * 0.965$$

$$UCP = 36.5156.$$

5.2.3 Cálculo del Esfuerzo.

$$E = UCP * CF$$

Donde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

Para calcular Factor de Conversión (CF):

CF = 20 horas-hombre (si Total EF \leq 2).

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4).

CF = abandonar o cambiar proyecto (si Total EF \geq 5).

Total **EF** = Cant. EF < 3 (entre E1 – E6) + Cant. EF > 3 (entre E7 – E8)

Como:

Total **EF** = 2 + 0

Total **EF** = 2.

CF = 20 horas-hombre (porque Total EF = 2).

Luego:

$$E = 36.5156 * 20 \text{ horas-hombre}$$

$$E = 730,312 \text{ horas-hombre.}$$

5.2.4 Distribución del Esfuerzo entre las diferentes actividades del sistema.

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10	182,578
Diseño	20	365,156

Implementación	40	730,312
Prueba	15	273,867
Sobrecarga	15	273,867
Total	100	1825,78

Tabla 23 Distribución del esfuerzo estimado entre los flujos de trabajo de RUP.

5.3 Beneficios tangibles e intangibles.

El desarrollo del sistema de servicios de información cultural para dispositivos móviles aporta para la Empresa ETECSA grandes beneficios económicos. El mayor aporte del servicio está en brindarle a los usuarios las noticias más recientes sobre los eventos culturales que van a estar ocurriendo en el país en esos momentos. Además, no es necesaria la descarga de herramientas de alto costo monetarias y limitadas en tiempo de uso para el desarrollo del sistema.

5.4 Análisis de costos y beneficios.

Salario medio de la fuerza de trabajo: \$100.

Cantidad de trabajadores: 2.

Como se tiene 1825,78 horas-hombre según lo estimado, se divide este valor entre 8 horas (cantidad real de horas que le dedica cada trabajador al sistema) para calcular la cantidad de días de trabajo-hombre que se necesitan para realizar la aplicación. Al realizar esta operación matemática se concluye que para realizar el sistema se necesitan 228,22 días de trabajo-hombre. Este valor se divide entre la cantidad de trabajadores para obtener la cantidad de días de trabajo que tiene cada trabajador para el desarrollo. De este resultado se concluye que con 2 trabajadores que trabajan 8 horas diarias, el sistema se terminaría en 114 días. Debido a que sólo se trabajan 6 días a la semana, se debe sumar a esta estimación 16 días más, que se corresponden con los 16 domingos que están dentro de esa planificación y que no se trabajan. Por tanto finalmente para realizar el módulo se necesitan aproximadamente 130 días (4 meses y medio). El costo final del sistema es:

Costo total del sistema = salario medio * cantidad de trabajadores * duración proyecto (meses)

Costo total del sistema = \$ 100 * 2 * 4.5 = \$ 900.

El desarrollo de la aplicación no constituye un gasto considerable pues la mayoría de las herramientas y tecnologías que se han empleado en su desarrollo son libres. El sistema que se propone en este trabajo no conlleva a grandes gastos, puesto que solo es influyente el salario de los desarrolladores. Una vez implantada la herramienta les brindará a los usuarios de Cubacel un servicio de información cultural sobre temas de interés, la empresa tendría ganancias económicas mediante el uso de servicio de mensajería por parte de los usuarios y mayor promoción de los servicios a los cuales se puede acceder mediante móviles.

Teniendo en cuenta el análisis realizado se concluye que es factible la implementación de la propuesta del sistema.

5.5 Conclusiones

En este capítulo se obtuvieron los estimados de horas por hombre que se requieren para desarrollar la solución y el costo aproximado. Se realizó un análisis de los distintos indicadores y se concluyó que era factible la realización del sistema para el envío de mensajes sobre información cultural para dispositivos móviles.

CONCLUSIONES

Con el propósito de darle cumplimiento al objetivo general y a la problemática planteada de este trabajo se han llevado a cabo satisfactoriamente cada una de las tareas que fueron trazadas al comienzo del mismo. En la realización de esta investigación se profundizó en el conocimiento de las herramientas para la creación de un sistema para el envío de sms sobre información cultural para dispositivos móviles, el proceso de desarrollo de software fue guiado por la metodología RUP, con el cual se obtuvo la implementación de un servicio de envío de mensajes sobre información cultural. El diseño y la implementación se rigieron por los estándares internacionales establecidos para la elaboración de portales con capacidad de ser visualizados de manera correcta en dispositivos móviles. Se puede concluir que se ha cumplido satisfactoriamente el objetivo trazado para este trabajo.

RECOMENDACIONES

Con el objetivo de efectuar un seguimiento y perfeccionamiento de esta investigación, se recomienda.

- Agregarle al sistema la funcionalidad de permitirle al administrador gestionar los tipos de eventos.

BIBLIOGRAFÍA.

1. Tempos 21. [En línea] [Citado el: 15 de Enero de 2010.] http://www.tempos21.com/web/files/PRESENTACION_CORPORATIVA_TEMPOS21.pdf.
2. **Junquera., Rafael A.** Servicios de valor agregado en la telefonía móvil. [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.key4biz.it/files/000051/00005133.pdf>.
3. **Cosio Cruz, Harley, Roques del Valle, Dianelys.** *Desarrollo del Portal WAP para la plataforma de gestión de contenidos Gina.* 2009.
4. Manual de Java. *Características de Java.* [En línea] [Citado el: 15 de enero de 2010.] <http://www.manual-java.com/manualjava/caracteristicas-java.html>.
5. **Niska, Petri.** JSR-212: SAMS Messaging. [En línea] [Citado el: 20 de Enero de 2010.] http://phoenix.labri.fr/documentation/sip/Documentation/Papers/Programming_SIP/Presentation/Jain/JSR-212.pdf.
6. introduccion_spring_framework. [En línea] [Citado el: 23 de Enero de 2010.] http://www.palermo.edu/ingenieria/downloads/introduccion_spring_framework_v1.0.pdf.
7. Hibernate. [En línea] [Citado el: 15 de enero de 2010.] <http://hibernate.bluemars.net/>.
8. PostGreSQL vs. MySQL. [En línea] [Citado el: 12 de Enero de 2010.] http://www.netpecos.org/docs/mysql_postgres/x15.html.
9. **Booch Grady, Jacobson Ivar, Rumbaugh James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2000.
10. Netbeans. [En línea] [Citado el: 12 de Diciembre de 2010.] http://www.techblog.com/talks/netbeans65es_cl.pdf.
11. Foundation Eclipse. [En línea] 2009. [Citado el: 15 de Enero de 2010.] <http://www.eclipse.org/>.
12. visual paradigm para UML. *visual paradigm para UML.* [En línea] [Citado el: 26 de Enero de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
13. CUBACEL. [En línea] [Citado el: 20 de Noviembre de 2009.] <http://www.cubacel.cu/>.
14. ETECSA. *Telefonía móvil.* [En línea] [Citado el: 12 de Noviembre de 2009.] <http://www.etecsa.cu/servicios.asp?codigo=147&padre=147>.

15. La evolucion de la telefonia movil. [En línea] [Citado el: 20 de Noviembre de 2009.] <http://www.eveliux.com/mx/la-evolucion-de-la-telefonia-movil.php>.
16. Plataforma de Eclipse. [En línea] [Citado el: 22 de Enero de 2010.] <http://eclipse.org/whitepapers/eclipse-overview.pdf>.
17. Telefonía Móvil y servicio de valor agregado. [En línea] [Citado el: 12 de Noviembre de 2009.] http://www.carrieryasoc.com/index.php?option=com_content&task=view&id=735&Itemid=1.
18. Ventajas de PostgreSQL. [En línea] [Citado el: 25 de Enero de 2010.] http://soporte.tiendalinux.com/porta/Portfolio/postgresql_ventajas_html.
19. **Biswas, Rahul ,Ort, Ed.** Oracle. [En línea] [Citado el: 20 de Enero de 2010.] <http://java.sun.com/developer/technicalArticles/J2EE/jpa/>.
20. **Cimadevilla Valdés, Odalys.** Hacia la expansión celular en Cuba . *CubaSi*. Diaria, 2009.
21. **González, Carlos D.** Curso Base de Datos PostgreSQL, SQL avanzado y PHP. [En línea] [Citado el: 25 de Enero de 2010.] <http://www.usabilidadweb.com.ar/postgre.php>.
22. **Lago Bagüés, Ramiro.** <http://www.proactiva-calidad.com/java/spring/introduccionSpring.html>. [En línea] [Citado el: 20 de Enero de 2010.] <http://www.proactiva-calidad.com/java/spring/introduccionSpring.html>.
23. **Larman, Craig.** *UML y Patrones.UML y Patrones*. s.l. : Prentice Hall, 2002.
24. **Pérez, Carlos Alejandro.** Java Platform Micro Edition (Java ME) Mensajes MMS y SMS. [En línea] [Citado el: 20 de Enero de 2010.] <http://www.sicuma.uma.es/sicuma/independientes/argentina08/Gaona-Perez/inicio.html>.
25. **Rafael A. Junquera.** Servicios de valor agregado en la telefonía móvil. [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.key4biz.it/files/000051/00005133.pdf>.

REFERENCIAS BIBLIOGRÁFICAS.

1. Tempos 21. [En línea] [Citado el: 15 de Enero de 2010.] http://www.tempos21.com/web/files/PRESENTACION_CORPORATIVA_TEMPOS21.pdf.
2. **Junquera., Rafael A.** Servicios de valor agregado en la telefonía móvil. [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.key4biz.it/files/000051/00005133.pdf>.
3. **Cosio Cruz, Harley, Roques del Valle, Dianelys.** *Desarrollo del Portal WAP para la plataforma de gestión de contenidos Gina.* 2009.
4. Manual de Java. *Características de Java.* [En línea] [Citado el: 15 de enero de 2010.] <http://www.manual-java.com/manualjava/caracteristicas-java.html>.
5. **Niska, Petri.** JSR-212: SAMS Messaging. [En línea] [Citado el: 20 de Enero de 2010.] http://phoenix.labri.fr/documentation/sip/Documentation/Papers/Programming_SIP/Presentation/Jain/JSR-212.pdf.
6. introduccion_spring_framework. [En línea] [Citado el: 23 de Enero de 2010.] http://www.palermo.edu/ingenieria/downloads/introduccion_spring_framework_v1.0.pdf.
7. Hibernate. [En línea] [Citado el: 15 de enero de 2010.] <http://hibernate.bluemars.net/>.
8. PostgreSQL vs. MySQL. [En línea] [Citado el: 12 de Enero de 2010.] http://www.netpecos.org/docs/mysql_postgres/x15.html.
9. **Booch Grady, Jacobson Ivar, Rumbaugh James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2000.
10. Netbeans. [En línea] [Citado el: 12 de Diciembre de 2010.] http://www.techblog.com/talks/netbeans65es_cl.pdf.
11. Foundation Eclipse. [En línea] 2009. [Citado el: 15 de Enero de 2010.] <http://www.eclipse.org/>.
12. visual paradigm para UML. *visual paradigm para UML.* [En línea] [Citado el: 26 de Enero de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).

GLOSARIO DE TÉRMINOS.

API: Una interfaz de programación de aplicaciones (Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Aplicación Web: Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

Aplicación: En informática es un tipo de software diseñado para facilitar al usuario la realización de una determinada tarea o trabajo.

J2EE: Java 2 Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java.

RUP: Rational Unified Process es una metodología de desarrollo de software, es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

UML: Unified Modeling Language es un lenguaje de modelado gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

User: representa a los usuarios del sistema.

Framework: Es una estructura de soporte definida, representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

GSM: Sistema global para la comunicación entre móviles, actualmente el estándar más extendido entre los teléfonos móviles a nivel mundial, con un total estimado de 1800 millones de usuarios repartidos en todo el orbe. Así los operadores telefónicos son capaces de ofrecer un servicio más seguro y con compatibilidad internacional o 'roaming'.

IDE: Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

MVC: Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

MMS: El sistema de mensajería multimedia es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia, incorporando sonido, video y fotos. El límite de cada mensaje multimedia suele ser de 100 o 300 KB, dependiendo de

cada móvil, si bien ese límite lo definen el operador o las características del terminal y no el protocolo.

SMS: Servicio de Mensaje Corto. Disponible en redes digitales como Sistema global para la comunicación entre móviles (GSM) permitiendo enviar y recibir mensajes de texto de hasta 160 caracteres a teléfonos móviles vía el centro de mensajes de un operador de red.

EMS: servicio de mensaje mejorado, es un sms con la opción de enviar música, vídeo, una grabación o una imagen.

SVA: El valor agregado es el valor que se le adiciona al ya plasmado, en este caso los nuevos servicios que pueden brindar los dispositivos móviles a demás de los que ya poseen.

WAP Push: Es una tecnología de nueva generación que emplea SMS, el cual se envía al móvil por parte del operador generalmente conteniendo algún tipo de dirección en el cuerpo del mensaje mediante la cual el usuario puede acceder al contenido. Está especificación permite enviar un mensaje a un dispositivo móvil sin previa autorización del usuario.

WAP: Es el protocolo de aplicaciones inalámbricas, es un estándar global y abierto. El WAP permite que dispositivos móviles, tales como los teléfonos o los PDAs, accedan a información o servicios especialmente creados en Internet.

AMPS: Es un sistema de telefonía móvil de primera generación (1G, voz analógica) desarrollado por los laboratorios Bell.