

Universidad de las Ciencias Informáticas

Facultad 6



Sistema de notificación de eventos

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

Autor: Javier Pérez Barroso

Tutor: Ing. Edmis Deivis Semanat Aldana

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2011.

Autor:

Javier Pérez Barroso

Tutor:

Edmis Devis Semanat Aldana.

DEDICATORIA

Dedico este trabajo a mi mamá por darme todo el cariño, confianza y fuerza para lograr los objetivos que me he trazado en los 5 años de universidad.

A mi papá por enseñarme a ser una mejor persona y darme todo el apoyo para seguir adelante.

A mi hermana por enseñarme a dar 3 brinquitos y transferirme sus conocimientos de la universidad.

A mi novia por darme el apoyo que necesité aquí en la escuela y tener mucha paciencia conmigo.

A toda mi familia por estar pendiente de mis resultados y apoyarme en todo lo que me hizo falta.

A mis amigos por estar cerca de mi y ayudarme con los trabajos de la universidad.

A todos muchísimas gracias...

AGRADECIMIENTOS

A mis padres por darme todo la fuerza necesaria para lograr mis objetivos.

A mi hermana por ayudarme con todo lo que me hizo falta.

*A mi novia por darme apoyo, confianza, ayudarme con la ortografía y con la imposible
Introducción.*

A Rene por joderme tanto con el so~~k~~ker y sacarme cada 5 minutos del lab.

A Deivis por tener paciencia conmigo y ayudarme con todo lo de la tesis.

*A Pupo por ayudarme con la programación y hacerme comprender que Linux es bueno, al
igual que Qt.*

A Rainer por ayudarme reiteradamente con los errores de compilación.

A Figuera por ayudarme a trabajar con hilos,...que difícil!!!

Al grupo de profesores del laboratorio por aclarar mis dudas.

*Al tribunal de tesis por ayudarme ha hacer posible que se desarrollara este trabajo con buena
calidad, incluyendo las conclusiones.*

A todos mis amigos por ayudarme que todo lo que me hizo falta...

A todos muchísimas gracias.

Un científico debe tomarse la libertad de plantear cualquier cuestión, de dudar de cualquier afirmación, de corregir errores.

Julius Robert Oppenheimer (1904-1967)

RESUMEN

Los componentes de envío de notificación presentan cada vez más demanda con el creciente aumento de las tecnologías de las comunicaciones, especialmente para la comunicación tanto interna como externa de empresas e instituciones. Al ser tan efectivos estos componentes y con la evolución de la tecnologías actuales, se han combinado para formar parte todo aquel software que se desarrolle.

El presente trabajo tiene como objetivo implementar varios plugins para el envío de notificaciones por diferentes medios de comunicación. Para el desarrollo se propone la utilización de librerías gratuitas y de código abierto compatible con los estándares escogidos.

Se obtuvo como resultado un componente que permita el envío de mensajes haciendo uso de diferentes medios de comunicación. Esto facilitaría una mejor comunicación entre el personal de cualquier empresa o institución que utilice los componentes en un determinado software.

ABSTRACT

The delivery of service components appear increasingly with the increasing demand for communications technologies, especially for both internal and external communication of companies and institutions. To be as effective these components and the evolution of current technologies, have combined to join anyone to develop software.

This work aims to implement several plugins for sending notifications for different media. Development proposes the use of free libraries and open source standards-compliant chosen.

The result was an element that allows sending messages using different media. This would facilitate better communication between the staff of any company or institution that uses a particular software component.

Contenido

Dedicatoria	iii
Agradecimientos	iv
Resumen	vi
Abstract	vii
Introducción	13
CAPÍTULO 1: Fundamentación Teórica de las herramienta la creación de un componente para el envío de notificaciones.	17
1.1 Introducción.....	17
1.2 Conceptos asociados al dominio del problema.	17
1.3 Objeto de Estudio	19
1.3.1 Descripción General	19
1.3.2 Descripción actual del dominio del problema.	20
1.3.3 Situación Problemática	20
1.4 Análisis de otras soluciones existentes.....	20
1.4.1 ACTIVEXPERTS SMS MESSAGING SERVER.....	21
1.4.2 SMSLibX 2.0	21
1.4.3 SMS Server software	22
1.4.4 Atomic Mail	23
1.4.5 Valoración general	23
1.5 Conclusiones del Capítulo.	24
Capítulo 2: Fundamentación de la propuesta	25
2.1 Introducción.....	25
2.2 Metodología de Desarrollo de Software.	25
2.2.1 Rational Unified Process (RUP).	25

2.2.2	Extreme Programming (XP).....	27
2.2.3	Ventajas y desventajas de las metodologías.....	28
2.2.4	¿Por qué RUP?	28
2.3	Tecnologías a usar para el desarrollo del componente.....	29
2.3.1	Java.....	29
2.3.2	Visual C++ .NET	30
2.3.3	C# 2.0	30
2.3.4	QT.....	31
2.3.5	¿Por qué Qt?	31
2.4	Herramientas	31
2.4.1	Herramientas CASE.	31
2.4.2	Rational Rose	32
2.4.3	Enterprise Architect 6.5.....	32
2.4.4	Visual Paradigm	33
2.4.5	Microsoft Office Visio 2007	34
2.5	IDE de desarrollo.....	34
2.5.1	Qt Creator	34
2.5.2	Netbeans	34
2.5.3	Eclipse	35
2.5.4	KDevelop.....	35
2.5.5	Microsoft Visual Studio 2008:	36
2.5.6	¿Por qué Qt Creator, Enterprise Architect 6.5 y Microsoft Office Visio 2007?	36
2.6	Modelo del dominio.....	36
2.6.1	Conceptos fundamentales.....	37

2.7	Requisitos	38
2.7.1	Requisitos funcionales	38
2.7.2	Requisitos no funcionales	39
2.8	Descripción del Componente. Modelos de Casos de Uso del Sistema	40
2.8.1	Determinación y justificación de los actores del sistema.....	40
2.8.2	Caso uso del sistema	40
2.8.3	Diagrama de caso de uso del sistema	41
2.1	Descripción de los caso de uso del sistema	42
2.8.4.1	Descripción del caso de uso: Enviar notificación vía TCP/IP	42
2.8.4.2	Descripción del caso de uso: Crear configuración.....	43
2.9	Conclusiones del capítulo.....	44
Capítulo 3: Análisis y diseño del sistema.....		46
3.1	Introducción.....	46
3.1	Análisis.....	46
3.2.1	Modelo del análisis	46
3.2.2	Diagramas de clases del análisis.....	47
3.2.2.1	Diagramas de clases del análisis CU Crear Configuración.	47
3.2.2.2	Diagrama de clases del análisis CU Crear TCP/IP.	47
3.2.3	Diagrama de interacción.	48
3.2.3.1	Diagrama de secuencia.	48
3.2.3.1	Diagrama de secuencia del análisis CU. Crear Configuración.....	48
3.2.3.2	Diagrama de secuencia del análisis CU. Enviar TCP/IP.....	49
3.3	Diseño.....	49
3.3.1	Diagrama de clase general	49

3.3.2	Patrones.....	50
3.3.2.1	Patrones utilizados.	51
3.4	Diagrama de clase del diseño	51
3.5	Conclusión del capítulo	52
Capítulo 4: Implementación.		53
4.1	Modelo de despliegue.....	53
4.1.1	Descripción de los componentes.	54
4.1.2	Descripción de los protocolos.	55
4.2	Diagrama de componentes.	56
4.3	Conclusiones del capítulo.....	57
Conclusiones Generales.....		58
Recomendaciones		59
Anexos I.....		60
Diagramas de clases del análisis CU Crear Configuración SMS.		60
Diagrama de clases del análisis CU Crear Configuración Beeper.		60
Diagrama de clases del análisis CU Crear Configuración Correo Electrónico.		60
Diagrama de clases del análisis CU Enviar SMS.		61
Diagrama de clases del análisis CU Enviar Beeper.....		61
Diagrama de clases del análisis CU Enviar Correo Electrónico.		61
Diagrama de clases del análisis CU Enviar TCP/IP.		62
Anexo II		63
Diagrama de secuencia del análisis CU. Crear Config. SMS		63
Diagrama de secuencia del análisis CU. Crear Config. Beeper.		63
Diagrama de secuencia del análisis CU. Crear Config. Correo Electrónico.		63

Diagrama de secuencia del análisis CU. Crear Config. TCP/IP.....	64
Diagrama de secuencia del análisis CU. Enviar SMS.....	64
Diagrama de secuencia del análisis CU. Enviar Beeper.....	65
Diagrama de secuencia del análisis CU. Enviar Correo Electrónico.....	65
Anexo III	66
Descripción del caso de uso: Enviar notificación via SMS	66
Descripción del caso de uso: Enviar notificación vía correo electrónico.....	66
Descripción del caso de uso: Enviar notificación vía beeper.	67
Descripción del caso de uso: Crear configuración TCP/IP.....	68
Descripción del caso de uso: Crear configuración beeper.	69
Descripción del caso de uso: Crear configuración SMS.....	70
Descripción del caso de uso: Crear configuración correo electrónico.....	72
Glosario de término.....	74
Referencia Bibliográfica.....	76
Bibliografía	78

INTRODUCCIÓN

La comunicación de información entre dos personas distantes, ya sea oral, escrita o gestual precisa que al menos una de ellas se desplace a un punto de encuentro con la otra. En dependencia de la distancia puede ser necesario contar con una vía y un sistema de desplazamiento eficiente para que se cumpla una de las premisas básicas de la comunicación: que sea rápida y eficaz.

Se puede considerar el origen de las telecomunicaciones desde tiempos muy remotos, cuando la información a transmitir se enviaba a través de mensajeros, que a pie o a caballo recorrían grandes distancias. El uso de mensajeros fue poco eficiente, los mensajes podían perderse o ser interceptados, además el retraso los hacían poco válidos en ciertas situaciones.

Las redes de telecomunicación tratan de crear medios dedicados que ahorren tiempo evitando el desplazamiento físico del mensajero a lo largo de todo el recorrido, proporcionando así una comunicación eficiente. Las primeras redes de telecomunicación propiamente surgen con la aparición de la telegrafía óptica, que permite sustituir a la mensajería, por tanto facilitan la transmisión de cualquier tipo de mensaje.

En febrero de 1876, Alexander Graham Bell inventó el teléfono, el cual evolucionó rápidamente, permitiendo incrementar la calidad de voz transmitida y la distancia de alcance. Posteriormente aparecen las primeras máquinas programables manualmente o con procedimientos electrónicos, las que evolucionan hasta hoy, para convertirse en potentes computadoras digitales que se introducen en casi todas las áreas de la sociedad como la industria, comercio, educación, transporte y en los medios de comunicación.

Actualmente la forma en que los seres humanos se relacionan e intercambian información cambió con el surgimiento del correo electrónico, la mensajería instantánea de texto, teléfonos celulares, las redes sociales y los diarios virtuales o blog. El uso de la tecnología de la comunicación permite el surgimiento de nuevas necesidades y deseos en los usuarios, aspectos que conducen al desarrollo de nuevas tecnologías de este tipo de forma aceleradas especialmente en los últimos años, las que a su vez crean nuevas posibilidades de intercambio de información.

Las organizaciones y empresas han implementado el uso de sistemas de comunicaciones con el fin de mejorar la efectividad y eficiencia de sus productos, ejemplo de ellos son las páginas web que brindan el

servicio de envío de SMS¹, sistemas de alertas contra incendio, envío de notificaciones hacia el correo electrónico y chat online.

Hoy el país se encuentra en un proceso de desarrollo de la telecomunicación que trae consigo nuevos avances a nivel educacional, cultural, relaciones políticas y/o sociales y continuos avances de niveles tecnológicos a nivel de gran escala.

En la Universidad de Ciencias Informáticas (UCI) se comenzó el desarrollo de un sistema de Video Vigilancia con el objetivo de mejorar la seguridad en instalaciones del MINTUR², pero por propuesta de dicha entidad requieren que el sistema envíe notificaciones por distintos medios de comunicación, éste constituye un valor agregado para dicho proyecto en desarrollo, posibilitando mejor rendimiento y confiabilidad del sistema.

La situación existente lleva a plantearse el siguiente problema a resolver. **¿Cómo lograr que el sistema de Video Vigilancia envíe notificaciones a los usuarios?** Teniendo en cuenta el objetivo general de **desarrollar un componente que permita enviar notificaciones por diferentes medios de comunicación y que el soporte de dichos medios sea basado en plugins.** Para dar solución a este problema se plantea como objeto de estudio **las tecnologías de transmisión de mensajes por redes informáticas**, delimitando como campo de acción **la creación de componentes de transmisión de mensajes por correo electrónico, SMS, HTTP³, Beeper y Sockets⁴.** Siendo la idea a defender; **si se implementa un componente de transmisión de notificaciones basado en plugins se satisfacen las**

¹ SMS: Servicio de Mensaje Corto

² MINTUR: Ministerio del Turismo

³ HTTP: Hypertext Transfer Protocol

⁴ Sockets: Es un método para la comunicación entre un programa del cliente y un programa del servidor en una red.

necesidades actuales de enviar notificaciones, además está preparado para futuras necesidades en cuanto a medios de transmisión. Como resultado se obtiene un componente capaz de enviar notificaciones por distintas vías de comunicación.

Para dar solución al problema planteado se propone como objetivo general desarrollar un componente que permita enviar notificaciones por diferentes medios de comunicación, el soporte para dicho medio debe ser mediante plugins, llevando a cabo las siguientes tareas de investigación:

- ✚ Realizar un análisis valorativo de componentes utilizados para el envío de notificaciones por diferentes medios de comunicación en la actualidad.
- ✚ Valorar las herramientas y tecnologías existentes para la realización de las aplicaciones necesarias y seleccionar las que se utilizarán.
- ✚ Realizar un análisis de las nuevas tecnologías que se usarán para la implementación del sistema.
- ✚ Realizar el levantamiento de requisitos del sistema representado en lenguaje natural.
- ✚ Obtener los modelos de dominio, análisis y diseño del sistema.
- ✚ Diseñar una aplicación que tenga una interfaz amigable e intuitiva para el usuario.
- ✚ Implementar el software con todas las funcionalidades requeridas.

Los métodos utilizados en la investigación se explicarán a continuación:

Métodos Teóricos.

Analítico-sintético: Es utilizado para estudiar y analizar documento y bibliografías de diferentes autores, sustentar desde el punto de vista teórico y práctico los elementos que se relacionan con el objeto de estudio.

Análisis histórico lógico: Se utiliza para hacer un estudio sobre la evolución y desarrollo que han tenido los componentes de envío de notificación, lo que permite valorar en qué etapa se encuentran dichos componentes, cuál es la tecnología más factible para el desarrollo de los mismos y realizar durante el proceso de estudio un análisis del pasado y el presente del objeto de estudio para realizar la proyección futura en el campo de acción escogido.

Modelación: Permite descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio. La modelación es justamente el proceso mediante el cual se crean modelos con vista a investigar la realidad, por lo que es el más importante en el proceso de construcción del software.

Métodos Empíricos

Observación: Permite realizar valoraciones y obtener información a partir de lo observado. Esto se manifiesta principalmente cuando se realizan observaciones sobre el funcionamiento de componentes similares al desarrollador, lo que da visión de cómo tiene que ser el componente.

Entrevista (estructurada): Permite tener una comunicación con otras personas, para obtener información relativa al sistema que se quiere construir.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LAS HERRAMIENTA LA CREACIÓN DE UN COMPONENTE PARA EL ENVÍO DE NOTIFICACIONES.

1.1 INTRODUCCIÓN

En este capítulo se hará una descripción de los principales sistemas de envío de notificación de eventos, así como la argumentación sobre varios de los más reconocidos productos que se están utilizando en el mundo y que representan una alternativa para ser usado en el desarrollo del componente informático que da solución al problema planteado en la presente investigación. Además de una descripción del objeto de estudio, la situación problemática y del entorno donde se desarrolla esta. Se llevará a cabo una conceptualización de los términos y procesos más importantes que rodean al problema científico.

1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.

Dentro de la investigación se encontrarán palabras claves que son de vital importancia conocer su significado debido a la complejidad de las mismas:

Sockets

- ✚ Es un método para la comunicación entre un programa del cliente y un programa del servidor en una red. (1)
- ✚ Un socket se define como el punto final en una conexión.
- ✚ Un socket es también una dirección de Internet, combinando una dirección IP⁵ y un número de puerto (el número que identifica una aplicación de Internet particular, como FTP⁶, o WWW⁷). (2)

⁵ IP: La dirección numérica única de cuatro partes que identifica a un ordenador particular en Internet.

- ✚ Número de identificación compuesto por dos números: La dirección IP y el número de puerto TCP⁸. En la misma red, el nº IP es el mismo, mientras el nº de puerto es el que varía.

Para interés del trabajo se tomará que sockets es: **un método para la comunicación entre un programa del cliente y un programa del servidor en una red.**

Plugins

- ✚ Se define como complementos de programas informáticos. Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande. Estos subprogramas no se instalan al igual que cualquier otro software, no funcionan por su cuenta y necesitan trabajar dentro de un programa que sea compatible con la tecnología de ese plugin. (3)
- ✚ Es un programa que extiende las capacidades del navegador de un modo específico, dado por ejemplo la capacidad de mostrar video, audio, ficheros de un determinado formato (ficheros PDF, presentaciones de ASAP, fichero VRML, etc.). (3)
- ✚ Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande. (4)

Teniendo en cuenta los conceptos mencionados anteriormente un plugins es: **un complemento de programas informáticos. Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande. Estos subprogramas no se instalan al igual que**

⁶ FTP: Protocolo de Transferencia de Archivos. Es un sistema que permite enviar y recibir ficheros entre computadores a través de la red Internet.

⁷ WWW: World Wide Web, también conocido como la telaraña, red o malla mundial.

⁸ TCP: Protocolo de Control de Transmisión.

cualquier otro software, no funcionan por su cuenta y necesitan trabajar dentro de un programa que sea compatible con la tecnología de ese plugin.

1.3 OBJETO DE ESTUDIO

El desarrollo de la computación y su integración con las telecomunicaciones en la telemática han propiciado el surgimiento de nuevas formas de comunicación. El desarrollo de las redes informáticas permitió su conexión mutua y, finalmente, la existencia de Internet, una red de redes gracias a la cual una computadora puede intercambiar fácilmente información con otras situadas en regiones lejanas del planeta.

La información a la que se accede a través de Internet combina el texto con la imagen y el sonido, es decir, se trata de una información multimedia, una forma de comunicación que está conociendo un enorme desarrollo gracias a la generalización de computadores personales dotados del hardware y software necesarios.

1.3.1 Descripción General

El desarrollo de la computación y su integración con las telecomunicaciones en la telemática, propician el surgimiento de nuevas formas de comunicación que son aceptadas cada vez por más personas como: el correo electrónico, envío de SMS, la mensajería instantánea, etc. El desarrollo de las redes informáticas permitió su conexión mutua y finalmente, la existencia de Internet, una red de redes a través de una computadora se puede intercambiar fácilmente información con otras usando protocolos; estos son un conjunto de reglas, secuencias, formatos de mensajes y procedimientos bien detallados que posibilitan la transferencia de datos entre dos o más sistemas de computación.

El Protocolo de Control de Transmisión y Protocolo de Internet (TCP/IP, siglas en inglés), es el conjunto de conexiones lógicas empleados por Internet, la red de redes. El TCP/IP, basado en software de aplicación de igual a igual, crea una conexión entre dos computadoras cualquiera, por donde se iniciará la transmisión de datos. Para que fluya la información usando la nueva tecnología de comunicación mediante redes informáticas en conjunto con el protocolo TCP/IP se tiene que tener en cuenta las capas del Modelo OSI y sus protocolos. Dichas capas describen el

proceso de transmisión de los datos de la red. Las dos únicas capas del modelo con que interactúa el usuario son las dos primeras capas: la capa Física y la última, la capa de Aplicación, con la Presentación de forma transparente; Sesión, Transporte, Red y Enlace de Datos. (5)

1.3.2 Descripción actual del dominio del problema.

En el proyecto de Video Vigilancia Inteligente de la UCI tiene un contrato con el MINTUR en la creación de un sistema de video vigilancia, pero el sistema no cuenta con la funcionalidad de envío de notificaciones, por lo que se quiere implementar un componente que envíe notificaciones por diferentes medios de comunicación para que pueda fluir la información entre el sistema y el usuario dicho componente debe soportar los medios de transmisión mediante plugins.

1.3.3 Situación Problemática

Con el aumento del turismo y los recursos que este aporta hacia nuestro país, no se tiene un control estricto de dichos recursos, por lo el MINTUR ha tenido la necesidad de buscar alguna vía de solución para preservar estos intereses nacionales. Recientemente esta institución ha firmado un contrato con la UCI para adquirir un sistema de video vigilancia con el objetivo de mantener controlado las 24 horas los ingresos hacia nuestro país.

Para tener una mejor comunicación entre el software de Video Vigilancia y el personal asignado para la interacción con éste, se ha decidido crear un componente con el objetivo de enviar a una determinada persona asignada una notificación sobre algún hecho que haya ocurrido en el área o recurso asignado para su protección y así poder prevenir algún hecho significativo.

1.4 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES.

Actualmente se habla de un gran desarrollo tecnológico, particularmente en las ramas de la Informática y las Telecomunicaciones, evidenciándose que es esta la era con mayor velocidad de evolución de todas aquellas que se haya conocido. Este acelerado desarrollo tecnológico ha dado surgimiento a las Nuevas

Tecnologías de la Información y las Comunicaciones, a la vez han conquistado los conocimientos y acciones que ayer mismo parecían inaccesibles pero, de la misma manera, le están condicionando y obligando a adaptaciones y replanteamientos en todos los órdenes de su existencia. Este aumento del desarrollo tecnológico ha propiciado que las industrias del software hayan tenido la necesidad de adaptarse a las nuevas demandas y crear nuevos productos con variadas funcionalidades para satisfacer las necesidades de los consumidores.

1.4.1 *ACTIVEXPERTS SMS MESSAGING SERVER*

Descripción

Este software provee un servicio de mensajería mediante el envío y recibo de SMS.

Características

- ✚ Enviar mensajes de texto SMS alfanuméricos.
- ✚ Enviar mensajes multimedia SMS, incluyendo imágenes y logos.
- ✚ Enviar nuevas indicaciones MMS de correo de voz, correo electrónico, fax.
- ✚ Verificar la entrega de mensajes salientes.
- ✚ Recibir mensajes SMS desde diferentes ubicaciones de almacenamiento.

Valoración

Luego de analizar las principales características del software **ACTIVEXPERTS SMS MESSAGING SERVER**, se pudo apreciar que dicho software presenta características similares con respecto al componente que se desea desarrollar como: el envío de SMS hacia teléfonos móviles. Esto permitirá tener una noción de cómo se debe desarrollar e implementar el componente de envío de notificación por medio de teléfonos móviles.

1.4.2 *SMSTLibX 2.0*

Descripción

- ✚ Componente de ActiveX (DLL) para enviar y recibir SMS de la PC usando un módem.
- ✚ Compañía: MENHIR Informática.
- ✚ Plataforma: Win98, WinME, WinNT 4.x, Win2000, WinXP
- ✚ Tamaño del archivo: 3234 KB

Características

- ✚ Enviar SMS largos.
- ✚ Enviar 20 SMS por minuto.

Valoración

Teniendo en cuenta las características mencionadas anteriormente el componente **SMSLibX 2.0** contiene características similares al componente que se desea desarrollar siendo esto un punto de referencia para el desarrollo de componente de envío de notificación por diferentes medios de comunicación.

1.4.3 SMS Server software

Descripción

Software para el envío y recibo de SMS desde una computadora hacia un teléfono móvil.

Características

- ✚ Libro de direcciones
- ✚ Archivos de registro
- ✚ Funcionalidad de la notificación
- ✚ Manija larga de los mensajes de SMS
- ✚ Conexión de TCP/IP.

Valoración

Luego de analizar las principales características del **SMS Server software** se pudo apreciar que presentan características similares a las del componente de envío de notificaciones que se desea desarrollar porque lo que se debe tener en cuenta dicho software para tomarlo como punto de referencia y obtener sus mejores características para el desarrollo del componente de envío de notificación por diferentes medios de comunicación.

1.4.4 Atomic Mail

Descripción

Es un poderoso software para enviar correo electrónico masivo, el cual acepta listas de envío en formato de archivos de texto. A su vez, Atomic Mail Sender puede utilizar listas generadas por nuestro extractor de direcciones de e-mail. Provee una gran variedad de rutinas básicas que le ayudarán a la administración de listas de envío: agregar nuevas direcciones, remover direcciones no usadas, etc.

Atomic Mail Sender es una herramienta que requiere una conexión a Internet para poder utilizarlo. Debido a que el programa dispone de su propio servidor SMTP interno, puede entregar e-mails de forma directa a los destinatarios sin utilizar la casilla de correo de su proveedor de internet.

1.4.5 Valoración general

Luego de analizar algunas de las principales soluciones existentes, las cuales presentan características de envío de notificaciones por diferentes medios de comunicación, se aprecia que presentan características similares por lo que se debe tener en cuenta a la hora de desarrollar el componente de envío de notificación por diferentes medios de comunicación. El país no está en condiciones de afrontar software o productos que estén representados por altos intereses comerciales, por lo que la industria del software conjuntamente con la dirección del país tiene que redimensionar el desarrollo hacia las tecnologías libres.

1.5 CONCLUSIONES DEL CAPÍTULO.

Con el transcurso del tiempo los medios de comunicación han tenido un papel fundamental en el desarrollo de software con el fin de mejorar la comunicación entre los usuarios y los productos que se vinculen con éste.

En este capítulo se realizó un profundo estudio de los principales componentes existente a nivel mundial, así como el objeto de estudio y la situación problemática por lo que se concluye que:

- ✚ Los componentes analizados no cumplen con las especificaciones solicitadas por el cliente, aunque presentan un alto interés comercial en el mercado mundial, por lo que es necesario la creación de un componente propio en la Universidad.
- ✚ Amplió los conocimientos sobre el ambiente donde se desarrolla el componente y el alcance que podría tener este a la hora de su construcción.
- ✚ El envío de sms haciendo uso de los teléfonos celulares no son una revolución sino mas bien una evolución de los medios de comunicación.

CAPÍTULO 2: FUNDAMENTACIÓN DE LA PROPUESTA .

2.1 INTRODUCCIÓN

En este capítulo se hace un análisis sobre las principales metodologías y herramientas utilizadas en la aplicación. Debido a la poca estructuración de los procesos de negocio y para poder comprender el contexto en el cual se desarrolla el sistema se determinó desarrollar un Modelo de Dominio, donde se expone un marco conceptual y las relaciones entre estas definiciones. Por otra parte, se enumeran los requerimientos funcionales y no funcionales, agrupándose los primeros en Casos de Uso, con el fin de estructurar el Diagrama de Casos de Uso del Sistema.

2.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE.

Todo el desarrollo de software es guiado por metodologías de desarrollo que guían todos los procesos a realizar. Las metodologías de desarrollo de software se dividen en dos grandes grupos de acuerdo al énfasis realizado en la documentación del desarrollo, estos son ágiles, tradicionales y robustos. Para el desarrollo del sistema se considera entre los principales exponentes de los dos grupos: XP (Extreme Programming, Programación Extrema) y RUP (Rational Unified Process, Proceso Unificado de Desarrollo).

2.2.1 Rational Unified Process (RUP).

Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (**UML**), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

El RUP está basado en 5 principios claves que son:

Adaptar el proceso: El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

Balancear prioridades: Los requerimientos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos. Debido a este balanceo se podrán corregir desacuerdos que surjan en el futuro.

Demostrar valor iterativamente: Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.

Elevar el nivel de abstracción: Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcos de referencia (Frameworks) por nombrar algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la mejor manera los requerimientos y sin comenzar desde un principio pensando en la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Estas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.

Enfocarse en la calidad: El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Principales características:

- ✚ Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- ✚ Pretende implementar las mejores prácticas en Ingeniería de Software
- ✚ Desarrollo iterativo
- ✚ Administración de requisitos
- ✚ Uso de arquitectura basada en componentes
- ✚ Control de cambios
- ✚ Modelado visual del software
- ✚ Verificación de la calidad del software

2.2.2 *Extreme Programming (XP)*.

Es un enfoque de la **ingeniería de software** formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, la programación extrema se diferencia de las metodologías tradicionales en que pone más énfasis en la adaptabilidad que en la previsibilidad. Se puede considerar como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

Los principios básicos de la programación extrema son: Simplicidad, Comunicación, Retroalimentación y Coraje.

Simplicidad: Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Para mantener la simplicidad es necesaria la refactorización del código, esta es la manera de mantener el código simple a medida que crece. También se aplica la simplicidad en la documentación, de esta manera el código debe comentarse en su justa medida, intentando que el código esté autodocumentado.

Comunicación: La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor mientras más simple sea. El código autodocumentado es más fiable que los comentarios ya que estos últimos pronto quedan desfasados con el código a medida que es modificado.

Retroalimentación: Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante.

Coraje o Valentía: Los puntos anteriores parecen tener sentido común, entonces, ¿por qué coraje? Para los gerentes la programación en parejas puede ser difícil de aceptar, es como si la productividad se fuese a reducir a la mitad ya que solo la mitad de los programadores está escribiendo código. Se necesita ser valiente para confiar en que la programación por parejas beneficia la calidad del código sin repercutir negativamente en la productividad. Se requiere coraje

para implementar las características que el cliente quiere ahora sin caer en la tentación de optar por un enfoque más flexible que permita futuras modificaciones.

2.2.3 Ventajas y desventajas de las metodologías.

Ventajas	
XP	RUP
comunicación	mayor documentación
simplicidad	verificar calidad del software
realimentación	configuración y control de cambio
coraje	modelado , guiado por caso de uso
disminuye trazas de errores	centrado en arquitectura
alta calidad mínimo tiempo	guiado por riesgo
Desventajas	
dificultad para determinar el costo del proyecto	los cambios son en una fase
se usa principalmente en proyectos pequeño	proyectos grandes

2.2.4 ¿Por qué RUP?

En correspondencia con el análisis realizado anteriormente sobre las metodologías seleccionadas, se puede evidenciar la presencia de diferentes características para cada una de ellas y la particularidad de poder ser empleadas para diferentes situaciones respectivamente. RUP es perfectamente idónea para proyectos de largo plazo, aunque es caracterizada por ser una

metodología compleja. La metodología XP está destinada a proyectos de corto plazo, se basa en iteraciones pequeñas y está dirigida fundamentalmente a los clientes.

Una vez estudiado lo referente a estas metodologías de desarrollo por lo que se considera por las características, utilizar RUP debido a que aporta todos los elementos para desarrollar aplicaciones grandes y que requieren de mucha documentación, permite desarrollar aplicaciones mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software, es la más utilizada en el mundo para el desarrollo de sistemas orientados a objeto ya que utiliza UML para el modelado, y la herramienta CASE a utilizar , Enterprise Architect , lo soporta completamente.

2.3 TECNOLOGÍAS A USAR PARA EL DESARROLLO DEL COMPONENTE

Las tecnologías a usar a la hora de desarrollar cualquier aplicación están en estrecha dependencia con respecto a los lenguajes de programación a utilizar.

2.3.1 Java

Potente lenguaje orientado a objetos y multiplataforma. Consta con una librería de clases base para el desarrollo, pero para aplicaciones de envergadura se requiere de Frameworks⁹ externos, los cuales son en su mayoría propietarios o libres que están en desarrollo por una comunidad de programadores y cuentan con escasa documentación. Las aplicaciones resultantes requieren de gran cantidad de memoria en las computadoras clientes. (9)

⁹Framework: Librería de código (posiblemente ya compilado) que sirve como base para el desarrollo de un tipo determinado de aplicaciones.

2.3.2 Visual C++ .NET

Reconocido lenguaje, no totalmente orientado a objetos, pero muy eficaz en cuanto a rapidez y uso de memoria en las aplicaciones que se obtienen. Carece de una librería de clases que realmente acelere el desarrollo de aplicaciones empresariales, y tiene muchas dificultades en cuanto a la interfaz de usuario. (10)

2.3.3 C# 2.0

Lenguaje estrella de la plataforma de desarrollo .NET. Cuenta con una extensa librería base para el desarrollo de todo tipo de aplicaciones. C# es un lenguaje de programación simple pero eficaz, este toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en un solo lenguaje.

C# presenta entre otras características:

- ✚ Sencillez.
- ✚ Modernidad.
- ✚ Orientado a Objetos.
- ✚ Orientado a componentes.
- ✚ Gestión automática de memoria.
- ✚ Seguridad de tipos.
- ✚ Instrucciones seguras.
- ✚ Sistema de tipos unificado.
- ✚ Extensibilidad de tipos básicos.
- ✚ Extensibilidad de operadores.
- ✚ Extensibilidad de modificadores.
- ✚ Posibilidad de crear versiones.
- ✚ Eficiencia
- ✚ Compatibilidad.

2.3.4 QT

Es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario y también para el desarrollo de programas sin interfaz gráfica como herramientas de la consola y servidores. Qt es utilizada en KDE, un entorno de escritorio para sistemas como GNU/Linux o FreeBSD, entre otros. Qt utiliza el lenguaje de programación C++ de forma nativa, adicionalmente puede ser utilizado en varios otros lenguajes de programación a través de *bindings*. Funciona en todas las principales plataformas, y tiene un amplio apoyo. El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red, una API multiplataforma unificada para la manipulación de archivos y una multitud de otros para el manejo de ficheros, además de estructuras de datos tradicionales.

Un avanzado editor de código C++, además soporta los lenguajes: C#/.NET Languages (Mono), Python: PyQt y PySide, Ada, Pascal, Perl, PHP y Ruby. Posee una GUI integrada y diseñador de formularios, herramienta para proyectos y administración, ayuda sensible al contexto integrado, depurador visual, resaltado y auto-completado de código, soporte para refactorización de código. (6)

2.3.5 ¿Por qué Qt?

Por las características que tiene el sistema se considera que el desarrollo de la aplicación se realice con Qt como lenguaje de desarrollo ya que el mismo “es simple pero eficaz”. Además porque este “combina las mejores características de lenguajes preexistentes como Visual Basic, Java o C++”, además se considera que el componente se debe crear en una plataforma libre para así poder usarla en cualquier otro software que quiera tener la capacidad de enviar notificaciones por diferentes medios de comunicación .

2.4 HERRAMIENTAS

2.4.1 Herramientas CASE.

Para el modelado de los artefactos y diagramas generados a lo largo del ciclo de vida del proyecto se hace necesario utilizar una herramienta CASE. Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones

informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

2.4.2 *Rational Rose*

Es una herramienta software para el modelado visual mediante UML de sistemas software. Permite especificar, analizar y diseñar el sistema antes de Codificarlo.

Algunas de las características que presenta Rational Rose son:

- ✚ Mantiene la consistencia de los modelos del sistema software
- ✚ Chequeo de la sintaxis UML
- ✚ Generación de documentación automáticamente
- ✚ Generación de código a partir de los modelos
- ✚ Ingeniería inversa (crear modelo a partir código)

Costo de la Licencia: Alto

Cubre todo el ciclo de vida de un proyecto:

- ✚ Concepción y formalización de un proyecto.
- ✚ Construcción de los componentes.
- ✚ Transición a los usuarios y certificación de las distintas fases.

2.4.3 *Enterprise Architect 6.5*

Es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo, proporcionando una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento. También provee soporte para pruebas, mantenimiento y control de cambio. Entre las principales características que este brinda se pueden mencionar:

- ✚ Crea elementos del modelo UML para un amplio alcance de objetivos.
- ✚ Ubica esos elementos en diagramas y paquetes.

- ✚ Documenta los elementos que ha creado.
- ✚ Genera código para el software que está construyendo.
- ✚ Realiza ingeniería directa e inversa de código en lenguajes como ActionScript, C++, C#, Java, PHP, entre otros.
- ✚ Soporta todos los diagramas y modelos del UML. Puede modelar procesos de negocio, sitios web, interfaces de usuario, redes, configuraciones de hardware y mensajes. Estima el tamaño del proyecto en esfuerzo de trabajo en horas. Captura y traza requisitos, recursos, planes de prueba, solicitudes de cambio y defectos.

2.4.4 Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado ayuda a construir con mayor rapidez aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas de UML y proyectos. (25)

La **versión 6.4** incluye un paquete de aplicaciones ofreciendo varias herramientas que facilitan el desarrollo de proyectos de cualquier dimensión.

Las principales características son:

- ✚ Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en patrones de diseño: Elementos de reusabilidad en software orientado a objeto.
- ✚ Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- ✚ Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- ✚ La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- ✚ Soporte Enterprise Java Beans™ 2.0.
- ✚ Capacidad de análisis de calidad de código.

- ✚ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- ✚ Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
- ✚ Integración con otras herramientas de desarrollo de Rational.
- ✚ Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.
- ✚ Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

2.4.5 *Microsoft Office Visio 2007*

Este software de creación de dibujos y diagramas facilita a los profesionales empresariales la visualización, el análisis y la comunicación de información compleja. Los textos y tablas complicadas y difíciles de comprender en diagramas de Visio pueden comunicar información de un vistazo. En lugar de imágenes estáticas, se pueden crear diagramas de Visio conectados a datos que muestran información, son fáciles de actualizar y pueden hacer la diferencia en cuanto a productividad del equipo de desarrollo. (7)

2.5 *IDE DE DESARROLLO*

2.5.1 *Qt Creator*

Es un IDE creado por Trolltech para el desarrollo de aplicaciones con las bibliotecas Qt, requiriendo su versión 4.x. Los sistemas operativos que soporta en forma oficial son: GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además hay una versión para GNU/Linux. Mac OS X 10.4 o superior, requiriendo Qt 4.x. Windows XP y Vista, requiriendo el compilador MinGW y Qt 4.4.3 para MinGW. Este IDE tiene la ventaja de ser soportado por la misma compañía que soporta a Qt, por lo que propicia una mayor integración y aumenta el número de funcionalidades.

2.5.2 *Netbeans*

El entorno de desarrollo Netbeans es una herramienta libre y gratuita que permite a los programadores escribir, compilar, depurar y ejecutar programas. Especialmente fue desarrollado para el lenguaje Java pero perfectamente se puede utilizar en otros como C/C++, Ruby, PHP, Perl,

entre otros. NetBeans IDE es fácil de instalar y de uso instantáneo y se ejecuta en varias plataformas incluyendo Windows, Linux, Mac OS X y Solaris. Existe además un número importante de módulos que posibilitan la extensión de este IDE (21). Aparte de la filosofía de distribución y desarrollo que respalda a NetBeans, el IDE ofrece a los desarrolladores numerosas ventajas, en la creación de nuevas aplicaciones multiplataforma. Las versiones más recientes de NetBeans integran el framework Hibernate 3.0 que posibilita un desarrollo más eficiente en los términos de accesos a bases de datos por lo que se ha determinado su uso en el desarrollo de la solución propuesta.

2.5.3 Eclipse

Eclipse es una comunidad Open Source (*código abierto*), cuyos proyectos se centran en la construcción de una plataforma de desarrollo abierta formada por marcos extensibles, herramientas y rutinas para crear, desplegar y gestionar software en todo el ciclo de vida. El IDE es compatible con varios de los sistemas operativos más conocidos como son, todas las versiones de Windows a partir del 98, GNU Linux y Mac OS X. Es un IDE muy usado en la actualidad por su portabilidad y por la gran cantidad de lenguajes en los que se puede desarrollar por lo que brinda un gran servicio de soporte a través de la web. (21)

Principales ventajas

Mediante plugins se le pueden añadir multitud de funcionalidades, además es libre y tiene una amplia comunidad que ofrece considerable documentación sobre el trabajo con él.

Principal Desventaja

Entre sus principales desventajas está que consume gran cantidad de recursos del ordenador lo cual no es despreciable cuando se trata de aplicaciones que utilizarán funcionalidades del sistema operativo que ralentizan la máquina.

2.5.4 KDevelop

Surgió en 1998 con el fin de desarrollar un IDE fácil de usar para KDE (K Desktop Environment). Desde entonces está públicamente disponible bajo licencia GPL y soporta lenguajes de programación como: C, C++, Java, Ada, SQL, Python, Perl y Pascal. Solo existe en sistemas

GNU/Linux y otros sistemas Unix. Su última versión es la 3.5.5 y salió el 5 de agosto del 2009. Tiene como limitantes principales que su entorno gráfico es algo pobre y sólo corre sobre plataforma GNU/Linux.

2.5.5 Microsoft Visual Studio 2008:

Microsoft Visual Studio 2008 es el nuevo IDE que Microsoft ha desarrollado enfocado a las nuevas necesidades que involucra un nuevo mundo dentro del Desarrollo de Software y que viene con muchas mejoras y funcionalidades. Ha sido desarrollado específicamente para la plataforma .NET y en especial para el lenguaje C#. Es el IDE líder a nivel mundial en cuanto a funcionalidades y rendimiento. (12)

2.5.6 ¿Por qué Qt Creator, Enterprise Architect 6.5 y Microsoft Office Visio 2007?

Qt Creator porque cumple con muchas funcionalidades incluyendo soporte para sistemas de gestión de proyectos como Subversión y Git, depurador integrado, integración de ayuda sensible al contexto acerca de Qt, poderoso editor de código con múltiples funcionalidades, es mucho más ligero que el Eclipse e integra todas las herramientas de Qt de forma que hace mucho más fácil la programación y a abstracción de los pasos de generación y compilación del código. Para la realización de la Ingeniería y Documentación se seleccionó el Enterprise Architect ya que combina el poder de las últimas especificaciones UML 2.0 con un alto rendimiento, interfaz intuitivo, para brindar modelamiento avanzado al escritorio, y a todo el equipo de desarrollo e implementación. Como complemento se seleccionó el Microsoft Office Visio 2007 ya que documenta y diseña fácilmente los procesos y los sistemas para ofrecer una información completa y actualizada, explora rápidamente la información compleja para identificar tendencias excepcionales y puntos de vistas clave y comparte visualmente la información compleja para poder tomar decisiones más eficaces.

2.6 MODELO DEL DOMINIO

El Modelo de Dominio o Modelo Conceptual, permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Un Modelo del Dominio es una

representación de las clases conceptuales del mundo real, no de componentes software. Este modelo no se trata de un conjunto de diagramas que describen clases de software u objetos de software con responsabilidades, sino que puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las bondades de los diagramas UML para representar conceptos, el Modelo de Dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema en cuestión.

2.6.1 Conceptos fundamentales

Para una mejor comprensión del Diagrama del Modelo de Dominio estructurado en el siguiente epígrafe. A continuación se proporciona un marco conceptual con las definiciones identificadas, estas son:

- ✚ **Mensaje:** información a enviar.
- ✚ **Sistema de mensajería:** aplicación encargada de manipular las peticiones de mensajes enviadas por la fuente.
- ✚ **Fuente:** envía una petición de mensajes.
- ✚ **Receptor:** es el encargado de recibir las notificaciones enviadas por la fuente.

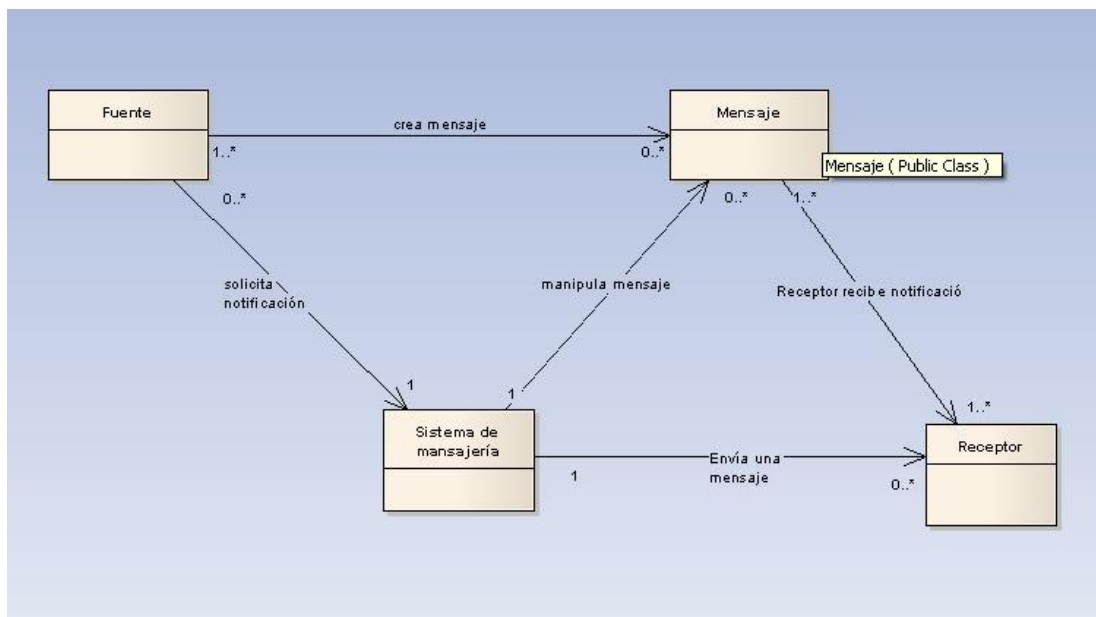


Figura 1: Diagrama de clase del dominio

2.7 REQUISITOS

Los requisitos son los que definen que es lo que debe hacer el sistema. Son las capacidades y condiciones que el sistema debe cumplir. Existen dos tipos de requisitos, los funcionales y los no funcionales. Los requisitos funcionales son los que especifican que es lo que el sistema debe hacer. Los no funcionales son los que especifican algo propio del sistema y como va a realizar sus funciones. A continuación se especificarán los requisitos funcionales y no funcionales que el componente de envío de notificaciones debe cumplir:

2.7.1 Requisitos funcionales

RF1. Enviar notificaciones

Este requerimiento envía notificaciones haciendo uso de diferentes medio de comunicación.

RF1.1 Enviar SMS

Este requerimiento envía una notificación a una fuente específica haciendo uso del SMS.

RF1.2 Enviar notificación vía TCP/IP

Este requerimiento envía una notificación a una fuente específica haciendo uso de la conexión TCT/IP.

RF1.3 Enviar notificación vía correo electrónico

Este requerimiento envía una notificación a una fuente específica haciendo uso del correo electrónico.

RF1.4 Enviar notificación vía Beeper

Este requerimiento envía una notificación a una fuente específica haciendo uso del beeper.

RF2 Crear configuración

Este requerimiento crea una configuración independientemente del tipo de notificación que se desea enviar.

RF2.1 Crear configuración SMS

Este requerimiento crea una configuración para enviar una notificación por SMS.

RF2.2 Crear configuración TCP/IP

Este requerimiento crea una configuración para enviar notificación por vía TCP/IP.

RF2.3 Crear configuración correo electrónico

Este requerimiento crea una configuración para enviar notificación por correo electrónico.

RF2.4 Crear configuración beeper

Este requerimiento crea una configuración para el envío de notificaciones por beeper.

RF2.5 Guardar configuración

Este requerimiento guarda la configuración de todos de los diferentes envíos con sus respectivos valores.

RF2.6 Actualizar configuración

Este requerimiento actualizar los valores de la configuración que se encuentre en el sistema.

2.7.2 Requisitos no funcionales

Rendimiento

El componente debe ser capaz de enviar notificaciones por diferentes medios de comunicación.

Fiabilidad

✚ **Disponibilidad:** El componente debe estar disponible las 24 horas.

✚ **Tiempo medio de reparación:** El componente no debe estar fuera de servicio más de 2 horas después del fallo.

Eficiencia

El componente debe ser eficiente de acuerdo con las solicitudes que se realicen en cada momento.

Usabilidad

La aplicación debe ser concebida para ser utilizada por personas que tengan conocimientos básicos de informática.

Tiempo de adiestramiento

Para que los usuarios puedan utilizar óptimamente el componente, necesitarán un tiempo de adiestramiento y estudio de los conceptos principales asociados al dominio del componente.

Software

✚ Framework: Qt

✚ Sistema Operativo: Windows XP SP2 o superior.

✚ Servidor de correo: ArgoSoft Mail Server

Hardware

✚ Procesador: Core2Duo

✚ Memoria: 2 GB

✚ Almacenamiento: 160 GB

Seguridad

Se debe garantizar la integridad, disponibilidad y confidencialidad de la información manipulada por el sistema.

Soporte

✚ Se necesita Windows XP SP2 o superior instalado.

2.8 DESCRIPCIÓN DEL COMPONENTE. MODELOS DE CASOS DE USO DEL SISTEMA

Después de haber mostrado las características y funcionalidades principales del componente de envío de notificaciones que debe cumplir, se hará una descripción del modelo de caso de uso del sistema. Para ello es necesario conocer acerca de los actores que interactúan en dicho componente.

2.8.1 Determinación y justificación de los actores del sistema

Actor	Descripción
Cliente	Representa una sistema externo del sistema que requiere el envío de una notificación por cualquier medio de comunicación.
Administrador	Encargado de configurar el envío de notificación por los diferentes medios de comunicación.

Tabla1: Actor del Sistema y su funcionalidad.

2.8.2 Caso uso del sistema

Un caso de uso del sistema es aquel que define que va a realizar el software pero no especifica como lo va hacer. Describe en un documento la secuencia de acciones que van ocurriendo entre el

actor que lo inicializa y el sistema, o sea, las respuestas que le da el sistema a las acciones del actor.

Caso de Uso	Prioridad
Enviar SMS	Crítico
Enviar correo electrónico	Crítico
Enviar beeper	Crítico
Enviar Notif. TCP/IP	Crítico
Crear configuración	Crítico
Crear Configuración TCP/IP	Crítico
Crear Configuración beeper	Crítico
Crear Configuración SMS	Crítico
Crear Configuración Correo Electrónico	Crítico

Tabla 2: Prioridad de caso de uso

2.8.3 Diagrama de caso de uso del sistema

Los Diagramas de Casos de Usos del Sistema representa la relación que existe entre el caso de uso y el actor que lo inicializa. Tienen gran importancia porque pueden modelar el comportamiento de un sistema, subsistema o una clase.

Un caso de uso describe un conjunto de secuencias, en las cuales cada secuencia representa la interacción de cosas fuera del sistema (actores) con el sistema (y sus abstracciones clave). Estos comportamientos son funciones a nivel del sistema que acostumbra visualizar, especificar, construir y documentar en la fase de obtención y análisis de los requerimientos. Un caso de uso representa un (o más) requerimiento funcional completo del sistema. (8)

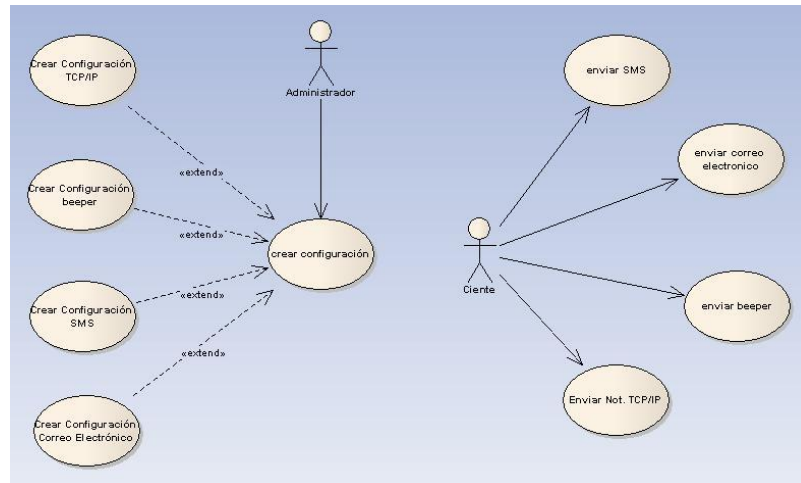


Figura 2: Diagrama de caso de uso del Envío de Notificación

2.1 Descripción de los caso de uso del sistema

Posteriormente se les mostrará una descripción bien detallada de los caso de uso del sistema considerados los más importantes perteneciente el componente de envío de notificación y las demás descripciones estarán referenciadas en los Anexos III.

2.8.4.1 Descripción del caso de uso: Enviar notificación vía TCP/IP

Caso de Uso:	Enviar notificación vía TCP/IP
Actores:	Actor
Propósito:	Envía solicitud de notificación vías TCP/IP.
Resumen:	El actor solicita el envío de una notificación vía TCP/IP
Precondiciones:	No se aplica
Referencias:	RF1.2
Prioridad:	Crítica

Flujo Normal de los Eventos:	
Acción del actor	Respuesta del sistema
1. El actor solicita "Enviar notificación TCP/IP".	2. El sistema envía una notificación cuando el mensaje se haya enviado.
Flujo Alternativo 1 "Cancelar"	
1.1 Cancela la solicitud	2.1 El sistema cancela la solicitud.
Flujo Alternativo 2 "Datos incorrectos"	
Pos condiciones	Envío de notificaciones

2.8.4.2 Descripción del caso de uso: Crear configuración.

Caso de Uso:	Crear configuración
Actores:	Administrador
Propósito:	Crear una configuración.
Resumen:	El administrador crea una configuración de acuerdo con el tipo de notificación que desea enviar.
Precondiciones:	No se aplica
Referencias:	RF2.0
Prioridad:	Crítica
Flujo Normal de los Eventos:	
Acción del actor	Respuesta del sistema
	1. El sistema muestra una interfaz de configuración de acuerdo con

	la configuración que se desea enviar la notificación.
2. El administrador introduce los datos de acuerdo con la configuración correspondiente.	3. El sistema valida los datos que sean correctos. 4. Guarda la configuración.
Flujo Alternativo 1 "Cancelar"	
2.1 El administrador selecciona la opción de "Cancelar".	2.2 Cierra la interfaz "Configuración" sin guardar ningún cambio.
Flujo Alternativo 2 "Datos incorrectos"	
	3.1 Muestra un mensaje datos incorrectos y se ejecuta el paso 3 del flujo normal de los eventos.
Pos condiciones	Crear configuración

2.9 CONCLUSIONES DEL CAPÍTULO

En el capítulo se abordaron los principales puntos para la continuidad del desarrollo del componente de envío de notificación por diferentes medios de comunicación, haciendo énfasis en la metodología escogida (RUP) la cual nos brinda disímiles ventajas, manteniendo una estructura basada en caso de uso para una mejor organización y composición del componente por lo que se concluyó que :

- ✚ La metodología es la más usada por los desarrolladores de softwares en el mundo.
- ✚ Teniendo en cuenta que el proyecto Video Vigilancia Inteligente usa como metodología de desarrollo RUP por su fortaleza y eficiencia en el análisis, diseño e implementación se decidió la construcción del componente usando esta metodología aunque podría usarse XP por su rapidez de construcción y por su poca utilización de recursos.

Teniendo en cuenta el IDE de desarrollo (QT Creator) y el lenguaje de programación (C++) seleccionado para la construcción del componente se concluye que:

- ✚ Tanto el IDE de desarrollo como el lenguaje de programación seleccionado presenta una gran documentación e información para su consulta.
- ✚ Provee un grupo de funcionalidades que hace más fácil al trabajo a la hora de implementar este.

La obtención de los requisitos funcionales y la creación de los diagramas en el capítulo tiene gran importancia en el desarrollo del componente, por lo que se concluye que:

- ✚ Estos nos brindan de forma gráfica la información para un mejor entendimiento de lo que se desea desarrollar y tener un punto de partida para el desarrollo del componente.
- ✚ Generan reportes detallados y completos con la información necesaria para la fase de desarrollo.
- ✚ Con la información brindada por el cliente se formulan los requisitos funcionales y no funcionales los cuales brindan una breve descripción de las características que presenta el componente y nos prepara para la creación del diagrama de caso de uso, el cual tiene un alto grado de información del componente y su creación.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.

3.1 INTRODUCCIÓN

Como resultado del flujo de trabajo de requisitos se obtiene una vista externa del sistema, expresado en el lenguaje del cliente, describiendo lo que se espera de él a través de casos de uso. A partir de aquí se debe profundizar en los casos de usos detallándolos de manera que permitan reflejar una vista interna del sistema descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidos. Este proceso se desarrolla fundamentalmente dentro de la fase de elaboración y se corresponde principalmente con el flujo de trabajo de análisis y diseño.

3.1 ANÁLISIS.

En el análisis se estructura los requisitos de manera que nos facilite su comprensión, su preparación, su modificación y en general su mantenimiento. Esta estructura (basada en clases de análisis y paquetes) es independiente de la estructura que se dio a los requisitos (basada en casos de uso). Sin embargo existe una trazabilidad directa entre esas distintas estructuras, la cual se define entre casos de uso del modelo de casos de uso y realizaciones del caso de uso en el modelo de análisis.

3.2.1 Modelo del análisis

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución.

- ✚ **Entidad:** Modelan información que posee larga vida y que es a menudo persistente.
- ✚ **Interfaz:** Modelan la interacción entre el sistema y sus actores.
- ✚ **Control:** Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso. (15)

3.2.2 Diagramas de clases del análisis.

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución. (9)

A continuación se hará un descripción de los diagrama del análisis más criticos y los demás estarán representados en los Anexos I.

3.2.2.1 Diagramas de clases del análisis CU Crear Configuración.

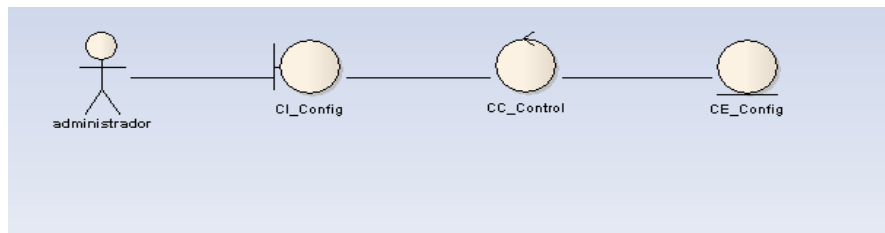


Figura 3: Diagrama de clases del análisis CU Crear Configuración.

Por lo que se pudo apreciar en la **figura 2** referente al **diagrama de clase del análisis del CU crear configuración** el administrador por intermedio de una interfaz configuraría el envío de una notificación por cualquier medio de comunicación.

3.2.2.2 Diagrama de clases del análisis CU Crear TCP/IP.

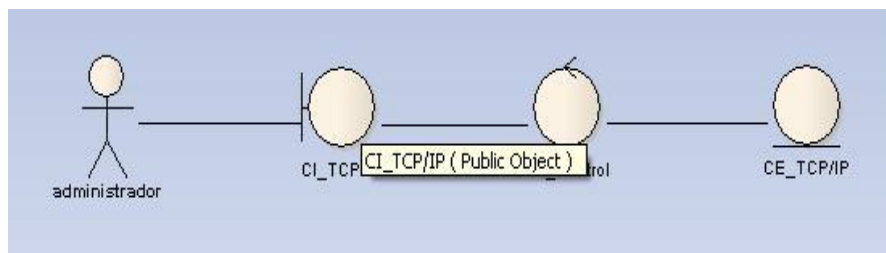


Figura 4: Diagrama de clase de análisis CU Crear Configuración TCP/IP.

En la **figura 4** el administrador es el encargado de configurar mediante una interfaz Configuración la notificación que se enviará por vía TCP/IP.

3.2.3 Diagrama de interacción.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. (9)

Estos no son sólo importantes para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa.

3.2.3.1 Diagrama de secuencia.

Los diagramas muestran gráficamente las interacciones del actor y de las operaciones a que dan origen, este diagrama se prepara durante la fase de análisis de un ciclo de desarrollo y su creación depende de la formulación previa de los casos de uso. El diagrama de secuencia muestra un determinado escenario de un caso de uso, los eventos generados por lo actores externos, su orden y los eventos internos del sistema. (9) En este epígrafe se hará una descripción de los diagramas de secuencia más significativos los demás estarán referenciados en los Anexos II.

3.2.3.1 Diagrama de secuencia del análisis CU. Crear Configuración.

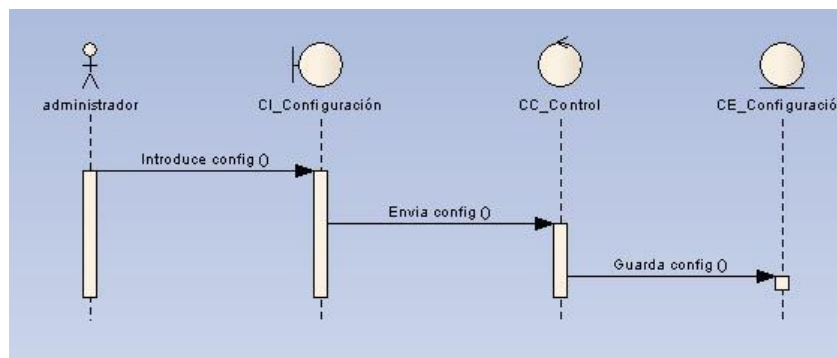


Figura 5: Diagrama de secuencia Crear Configuración.

En la **figura 12** referente al **diagrama de secuencia del CU crear configuración** se muestra como el administrador sigue una secuencia de pasos para crear una configuración para el envío de una notificación por un determinado medio de comunicación.

3.2.3.2 Diagrama de secuencia del análisis CU. Enviar TCP/IP.

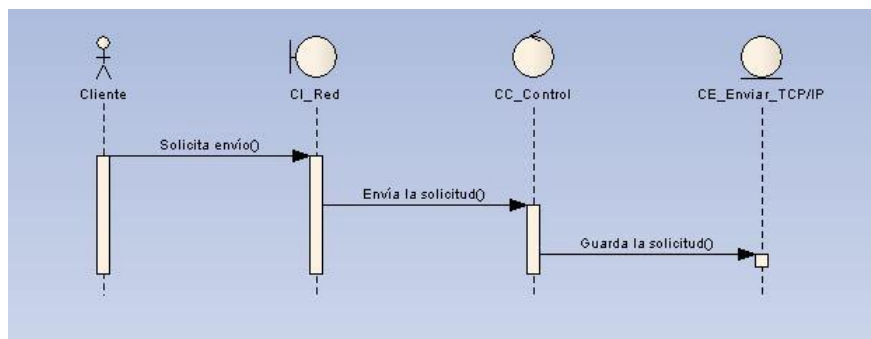


Figura 6: Diagrama de secuencia Enviar TCP/IP.

En la **figura 13 Diagrama de secuencia del CU enviar TCP/IP** muestra los pasos a seguir del cliente al solicitar un envío de una notificación por medio de TCP/IP.

3.3 DISEÑO.

El diseño crea una representación o modelo de software, pero a diferencia del modelo de análisis (que se enfoca en la descripción de los datos, las funcionalidades y el comportamiento requerido), el modelo de diseño proporciona detalles acerca de la estructura de datos, las arquitecturas, las interfaces y los componentes del software que son necesarios para implementar el sistema. (10)

3.3.1 Diagrama de clase general

En el diagrama de clase general visualiza la interacción física entre las clases persistentes teniendo en cuenta sus funcionalidades que brinda cada una de ellas en específico y sus relaciones con otras clases. Este artefacto se centra en la realización de los requisitos funcionales y no funcionales junto

con otras restricciones relacionadas con el entorno de implementación, por lo que se considera como la entrada fundamental utilizada para el correcto desarrollo del componente.

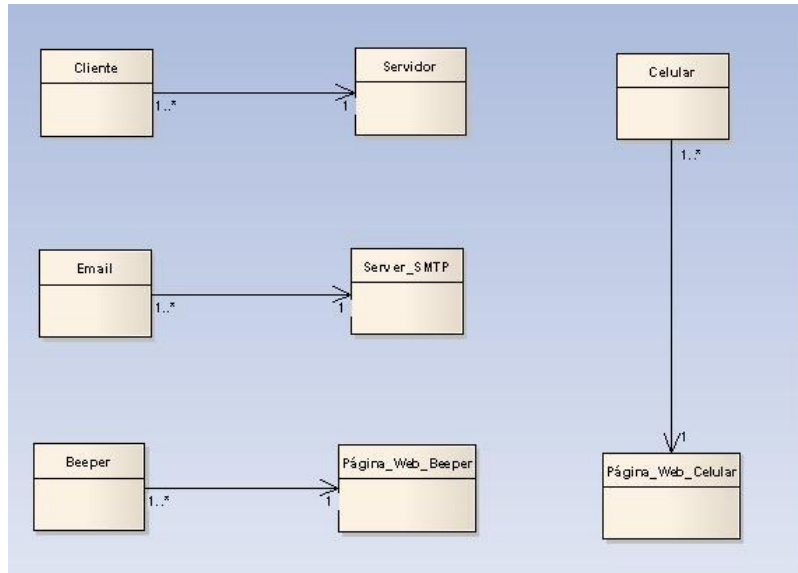


Figura 7: Diagrama de clase general.

3.3.2 Patrones.

Los patrones de diseño pueden usarse durante el diseño del software. Una vez que se ha desarrollado el modelo de análisis, el diseñador puede examinar una representación detallada del problema que debe resolver y las restricciones que impone el problema. (10)

- ✚ **Patrones arquitectónicos:** estos patrones definen la estructura general del software, indican las relaciones entre los subsistemas y los componentes del software y definen las reglas para especificar las relaciones entre los elementos (clases, paquetes, componentes, subsistemas) de la arquitectura. (10)
- ✚ **Patrones de diseño:** estos patrones se aplican a un elemento específico del diseño como un agregado de componentes para resolver algún problema de diseño, relaciones entre los componentes o los mecanismos para efectuar la comunicación de componente a componente. (10)

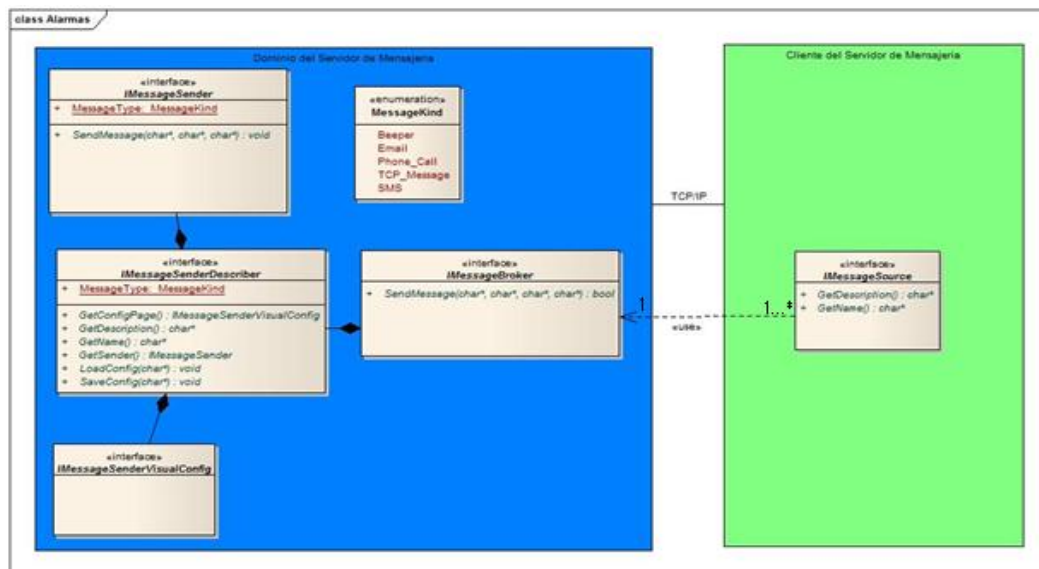
- ✚ **Idiomas:** a veces llamados patrones de código, estos patrones específicos de lenguaje por lo general implementan un elemento algorítmico o un componente, un protocolo de interfaz específico o un mecanismo de comunicación entre los componentes. (10)

3.3.2.1 Patrones utilizados.

- ✚ **Factory Method:** Define una interfaz para crear un objeto, pero deja que sean las subclases quienes decidan qué clase instanciar. Permite que una clase delegue en sus subclases la creación de objetos. (11)

3.4 Diagrama de clase del diseño

Un Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema. Este artefacto constituye la entrada fundamental utilizada para el correcto desarrollo de las entradas de implementación. En algunos de los diagramas no se verán los atributos y métodos de las clases, por el gran volumen de información por lo que se recomienda ver el fichero adjunto al documento.



3.5 CONCLUSIONES DEL CAPÍTULO

Teniendo en cuenta que en este capítulo se realizó un profundo estudio sobre el análisis y diseño del componente de envío de notificación se concluyó que:

- ✚ Se obtuvo un refinamiento de los requisitos sin tener en cuenta el lenguaje de programación a usar en la construcción.
- ✚ Se obtuvo una mejor comunicación entre los programadores y los patrones a utilizar y así completar el ciclo de la arquitectura y enfocarnos en la implementación.
- ✚ Se obtuvo una mejor comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionados con los componentes reutilizables, sistemas operativos, tecnología de distribución, concurrencia y tecnología de interfaz de usuario.

CAPÍTULO 4: IMPLEMENTACIÓN.

En la implementación, se parte de los resultados obtenidos en el diseño y se implementa el sistema en términos de componentes, como son los ficheros de código fuente, ficheros de código binario, ejecutables, entre otros. Primeramente, se detalla el modelo de datos del sistema, que es el modelo donde se ve la estructura en la cual se almacena toda la información requerida en el sistema. Luego se muestra el modelo de implementación que está compuesto por el diagrama de despliegue y el diagrama de componentes. Estos diagramas, describen los componentes a construir, su organización y dependencias entre los nodos físicos en la que funcionará la aplicación.

La mayor parte de la arquitectura del sistema es capturada durante el diseño, siendo un propósito principal de la implementación el desarrollar la arquitectura y el sistema como un todo. De forma más específica, los propósitos de la implementación son:

- ✚ Planificar las integraciones de sistemas necesarias a cada iteración. Seguimos para ello un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.
- ✚ Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue. Esto se basa fundamentalmente en las clases activas encontradas durante en el diseño.
- ✚ Implementar las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementan como componentes de fichero que contiene código fuente.
- ✚ Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema.

4.1 MODELO DE DESPLIEGUE

Modelo de Despliegue que es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos. El mismo está compuesto por:

Nodos: Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.

Dispositivos: Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.

Conectores: Expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

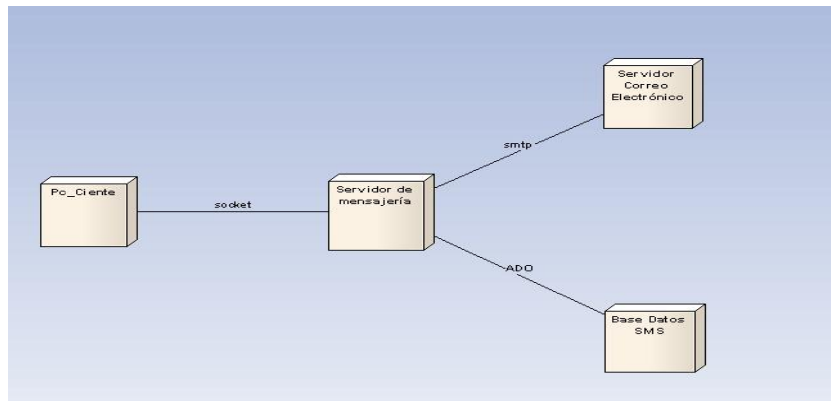


Figura 8: Diagrama de despliegue.

4.1.1 Descripción de los componentes.

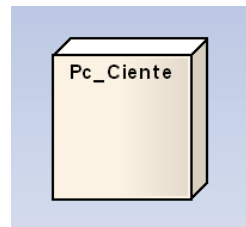


Figura 9: Nodo. **PC Cliente** del diagrama de despliegue.

Este nodo representa una PC cliente desde la cual se realizarán las peticiones de un envío de notificación.



Figura 10: Nodo. **Servidor de Correo Electrónico** del diagrama de despliegue.

Este nodo representa el servidor de correo electrónico el cual servirá para el envío de notificación por este medio de comunicación.

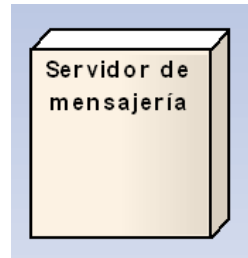


Figura 11: Nodo. **Servidor de mensajería** del diagrama de despliegue.

Este nodo representa donde se guardará toda aquella información del envío de mensajería.



Figura 12: Nodo. **Base de Datos SMS** del diagrama de despliegue.

Este nodo representa una base de datos donde se gestionará el envío de SMS.

4.1.2 Descripción de los protocolos.

- ✚ **SMTP:** Simple Mail Transfer Protocol (SMTP) Protocolo Simple de Transferencia de Correo, es un protocolo de la capa de aplicación. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA's, teléfonos móviles, etc.). (13)
- ✚ **Socket:** un método para la comunicación entre un programa del cliente y un programa del servidor en una red.
- ✚ **ADO:** ActiveX Data Objects (ADO) es uno de los mecanismos que usan los programas de computadoras para comunicarse con las bases de datos, darles órdenes y obtener resultados de ellas. (13)

4.2 DIAGRAMA DE COMPONENTES.

Los Diagramas de Componentes modelan la vista estática de un sistema. Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes, por lo cual cada diagrama describe un apartado del sistema. Estos tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Dado que los Diagramas de Componentes muestran los componentes software que constituyen una parte reusable, sus interfaces y sus interrelaciones, en muchos aspectos se puede considerar que un Diagrama de componentes es un Diagrama de clases a gran escala.

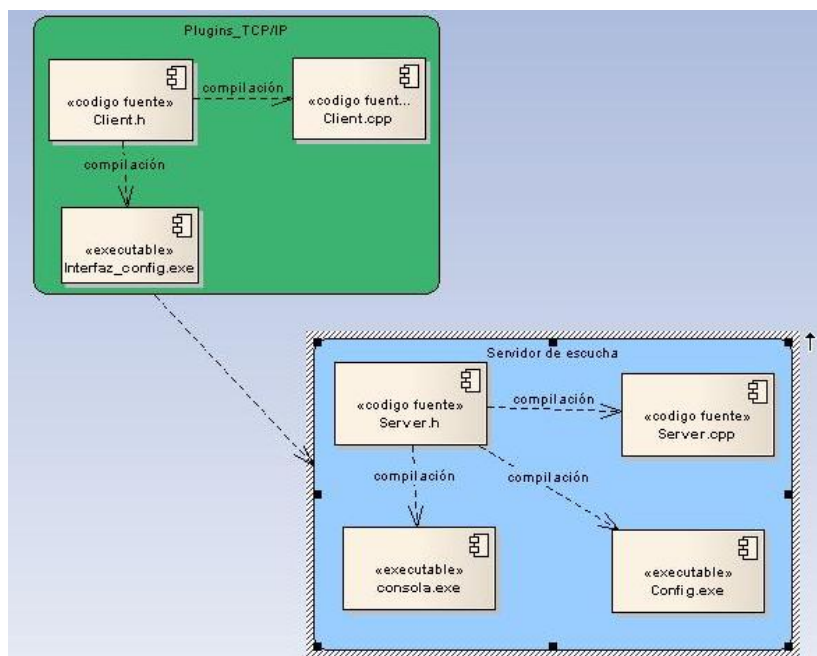


Figura 13: Diagrama de componente.

4.3 CONCLUSIONES DEL CAPÍTULO.

Teniendo en cuenta que el capítulo de implementación es el escalón fundamental para la construcción del componente de envío de notificación por lo que se concluye que:

- ✚ Se obtuvo un subsistema de implementación y sus dependencias, interfaces y contenido.
- ✚ Se obtuvo un componente de envío de notificación con todas las funcionalidades propuestas por el cliente y recogidas en los requisitos funcionales.
- ✚ Se obtuvo la integración de todos los resultados producidos en un servicio listo para su uso.

CONCLUSIONES GENERALES

Con el desarrollo del presente trabajo se obtuvo un componente capaz de enviar notificaciones haciendo uso de distintos medios de comunicación y el soporte de dichos medios es mediante plugins, se concluye que:

- ✚ Este componente puede ser usado por cualquier institución que necesite una vía rápida de comunicación entre los usuarios, especialmente los sistemas de video vigilancia los cuales presentan un constante movimiento de información y avisos al personal.
- ✚ Aumentó los conocimientos sobre los componentes de envío de notificación por lo que se considera un gran avance en la Universidad.
- ✚ Aumentó los conocimientos sobre toda aquella tecnología y herramientas usadas en el desarrollo del presente trabajo, considerando ésta para ser usada en otro componente que se desee desarrollar.

RECOMENDACIONES

- ✚ Se recomienda que el servicio de envío de notificaciones por diferentes medios de comunicación se vincule con el Gestor del proyecto de Video Vigilancia Inteligente con el objetivo de aumentar la efectividad de este.
- ✚ Hacer uso de otras formas de comunicación como el envío de MMS¹⁰ para hacer más extensible nuestro servicio.
- ✚ Generalizar los componentes para que otros proyectos de la Universidad y otras instituciones haga uso de nuestro servicio.

¹⁰ MMS: Sistema de mensajería multimedia.

ANEXOS I

Diagramas de clases del análisis CU Crear Configuración SMS.

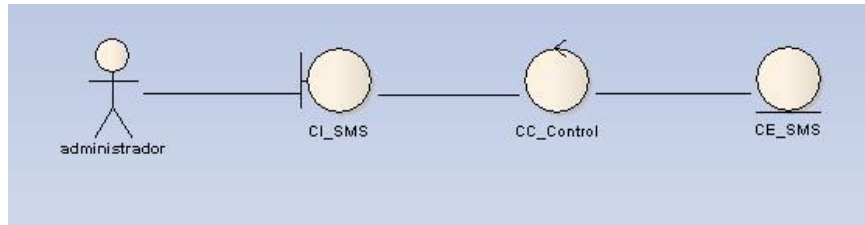


Figura 14: Diagrama de clase de análisis CU Crear Configuración SMS.

Diagrama de clases del análisis CU Crear Configuración Beeper.

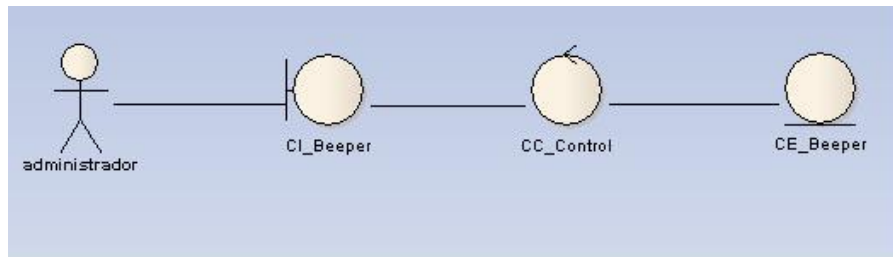


Figura 15: Diagrama de clase de análisis CU Crear Configuración Beeper.

Diagrama de clases del análisis CU Crear Configuración Correo Electrónico.

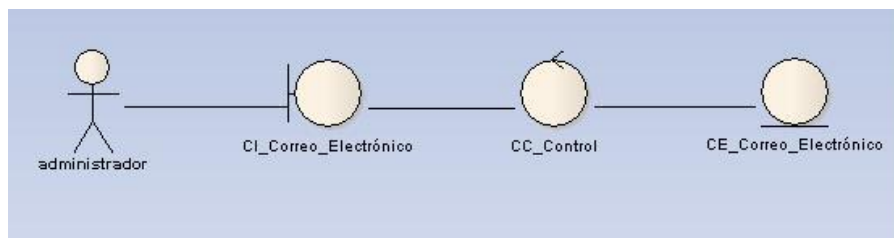


Figura 16: Diagrama de clase de análisis CU Crear Configuración Correo Electrónico.

Diagrama de clases del análisis CU Enviar SMS.

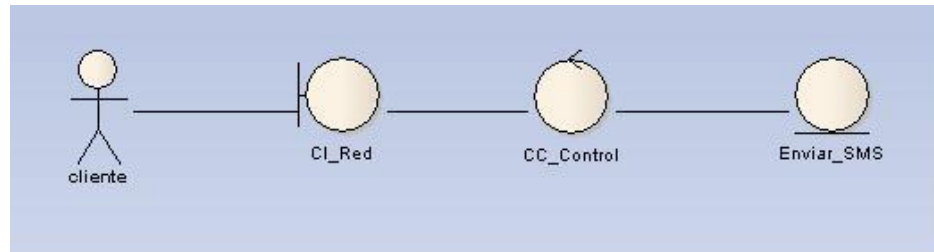


Figura 17: Diagrama de clase de análisis CU Enviar SMS.

Diagrama de clases del análisis CU Enviar Beeper.

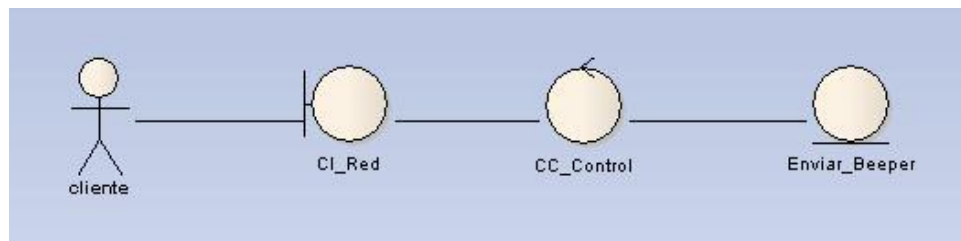


Figura 18: Diagrama de clase de análisis CU Enviar Beeper.

Diagrama de clases del análisis CU Enviar Correo Electrónico.

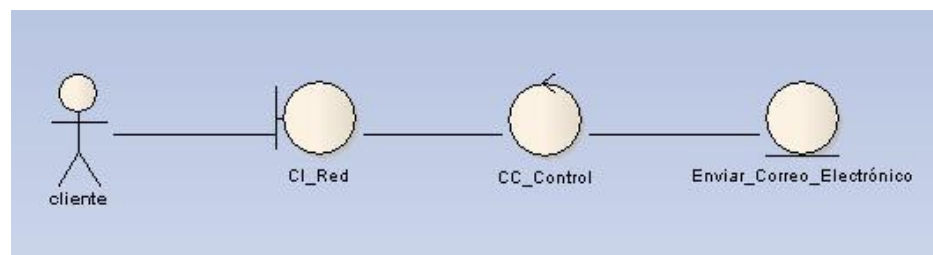


Figura 19: Diagrama de clase de análisis CU Enviar Correo Electrónico.

Diagrama de clases del análisis CU Enviar TCP/IP.

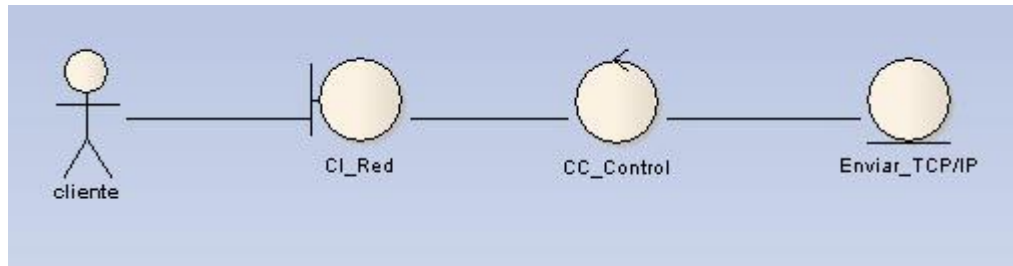


Figura 20: Diagrama de clase de análisis CU Enviar TCP/IP.

ANEXO II

Diagrama de secuencia del análisis CU. Crear Config. SMS

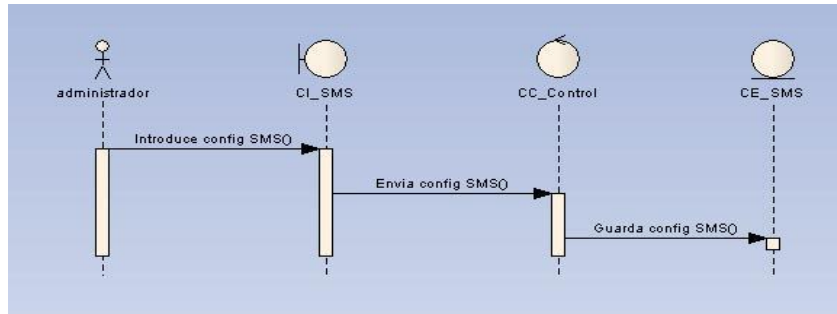


Figura 21: Diagrama de secuencia Crear Configuración SMS.

Diagrama de secuencia del análisis CU. Crear Config. Beeper.

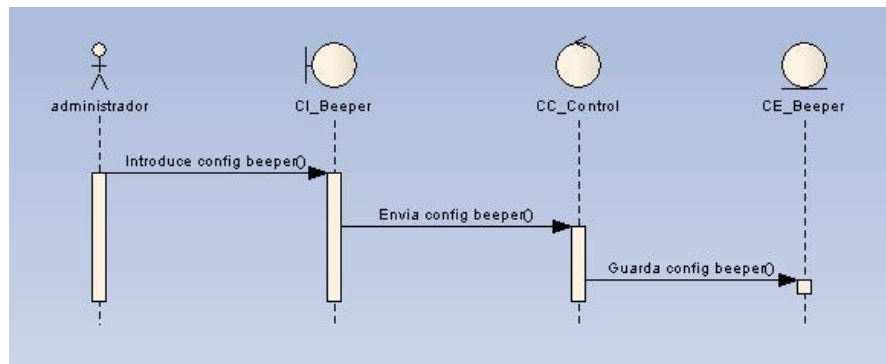


Figura 22: Diagrama de secuencia Crear Configuración Beeper.

Diagrama de secuencia del análisis CU. Crear Config. Correo Electrónico.

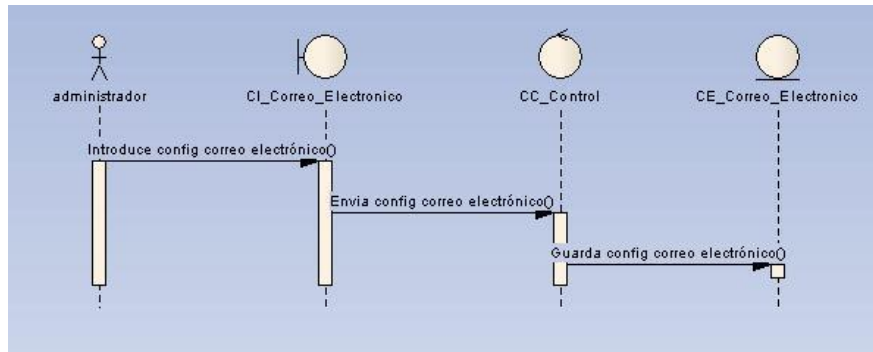


Figura 23: Diagrama de secuencia Crear Configuración Correo Electrónico.

Diagrama de secuencia del análisis CU. Crear Config. TCP/IP.

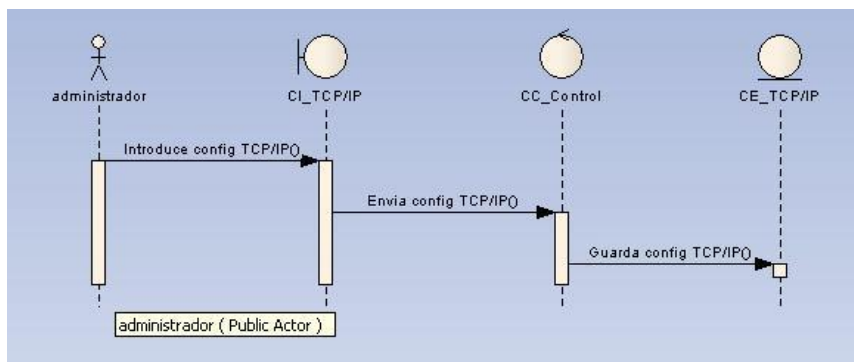


Figura 24: Diagrama de secuencia Crear Configuración TCP/IP.

Diagrama de secuencia del análisis CU. Enviar SMS.

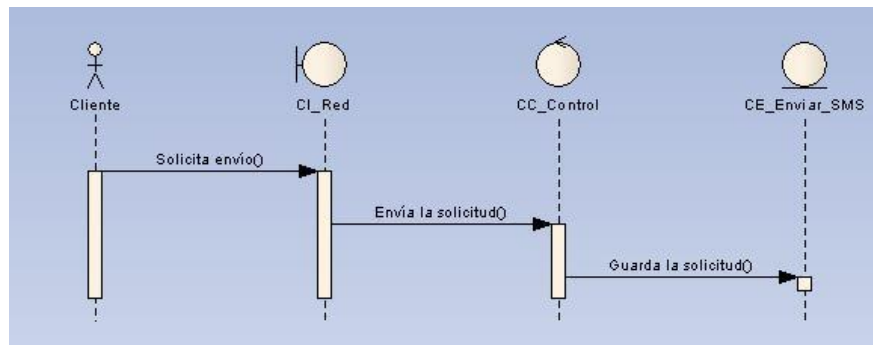


Figura 25: Diagrama de secuencia Enviar SMS.

Diagrama de secuencia del análisis CU. Enviar Beeper.

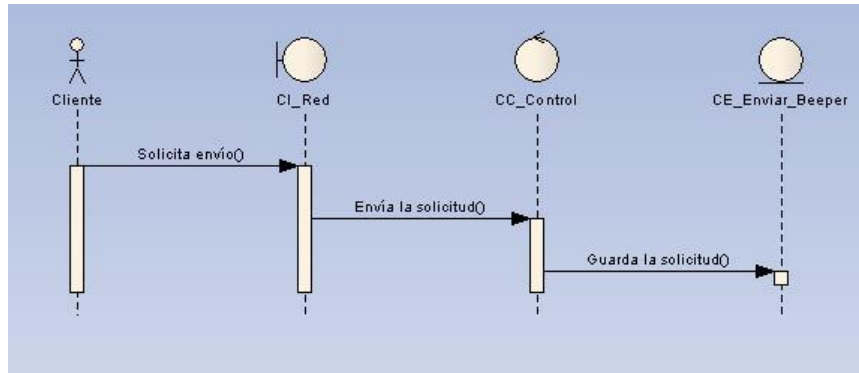


Figura 26: Diagrama de secuencia Enviar Beeper.

Diagrama de secuencia del análisis CU. Enviar Correo Electrónico.

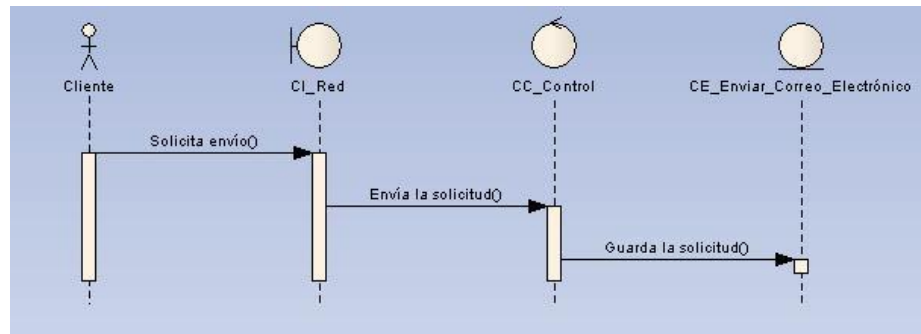


Figura 27: Diagrama de secuencia Enviar Correo Electrónico.

ANEXO III

Descripción del caso de uso: Enviar notificación vía SMS .

Caso de Uso:	Enviar notificación vía SMS	
Actores:	Actor	
Propósito:	Envía solicitud de notificación de SMS	
Resumen:	El actor solicita el envío de una notificación haciendo uso del SMS.	
Precondiciones:	No se aplica	
Referencias:	RF1.1	
Prioridad:	Crítica	
Flujo Normal de los Eventos:		
Acción del actor	Respuesta del sistema	
1. El actor solicita “Enviar SMS”.	2. El sistema envía una notificación cuando el mensaje se haya enviado.	
Flujo Alternativo 1 “Cancelar”		
1.1 Cancela la solicitud	2.1 El sistema cancela la solicitud.	
Flujo Alternativo 2 “Datos incorrectos”		
Pos condiciones	Envío de notificaciones	

Descripción del caso de uso: Enviar notificación vía correo electrónico

Caso de Uso:	Enviar notificación vía correo electrónico
---------------------	---

Sistema de notificación de eventos

Actores:	Actor
Propósito:	Envía solicitud de notificación de correo electrónico.
Resumen:	El actor solicita el envío de una notificación haciendo uso del correo electrónico.
Precondiciones:	No se aplica
Referencias:	RF1.3
Prioridad:	Crítica
Flujo Normal de los Eventos:	
Acción del actor	Respuesta del sistema
1. El actor solicita “Enviar correo electrónico”.	2. El sistema envía una notificación cuando el mensaje se haya enviado.
Flujo Alternativo 1 “Cancelar”	
1.1 Cancela la solicitud	2.1 El sistema cancela la solicitud.
Flujo Alternativo 2 “Datos incorrectos”	
Pos condiciones	Envío de notificaciones

Descripción del caso de uso: Enviar notificación vía beeper.

Caso de Uso:	Enviar notificación vía Beeper
Actores:	Actor
Propósito:	Envía solicitud de notificación de beeper.
Resumen:	El actor solicita el envío de una notificación haciendo uso del beeper.

Precondiciones:	No se aplica	
Referencias:	RF1.3	
Prioridad:	Crítica	
Flujo Normal de los Eventos:		
Acción del actor	Respuesta del sistema	
1. El actor solicita “Enviar beeper”.	2. El sistema envía una notificación cuando el mensaje se haya enviado.	
Flujo Alternativo 1 “Cancelar”		
1.1 Cancela la solicitud	2.1 El sistema cancela la solicitud.	
Flujo Alternativo 2 “Datos incorrectos”		
Pos condiciones	Envío de notificaciones	

Descripción del caso de uso: Crear configuración TCP/IP.

Caso de Uso:	Crear Configuración TCP/IP
Actores:	Administrador
Propósito:	Crear una configuración TCP/IP.
Resumen:	El administrador crea una configuración TCP/IP para el envío de una notificación por vía TCP/IP.
Precondiciones:	No se aplica

Referencias:	RF2.2
Prioridad:	Crítica
Flujo Normal de los Eventos:	
Acción del actor	Respuesta del sistema
	1. El sistema muestra una interfaz " Crear Configuración TCP/IP ".
2. El administrador introduce los datos correspondientes con la configuración.	3. El sistema valida los datos que estén correctos. 4. El sistema guarda la configuración.
Flujo Alternativo 1 "Cancelar"	
2.1 El administrador selecciona la opción de "Cancelar".	2.1 Cierra la interfaz " Crear Configuración TCP/IP " sin guardar ningún cambio.
Flujo Alternativo 2 "Datos incorrectos"	
	3.1 Muestra un mensaje de datos incorrecto y se ejecuta el paso 3.
Pos condiciones	Crear configuración

Descripción del caso de uso: Crear configuración beeper.

Caso de Uso:	Crear Configuración beeper
Actores:	Administrador
Propósito:	Crear una configuración beeper.

Sistema de notificación de eventos

Resumen:	El administrador crea una configuración de beeper para el envío de una notificación haciendo uso del beeper.	
Precondiciones:	No se aplica	
Referencias:	RF2.4	
Prioridad:	Crítica	
Flujo Normal de los Eventos:		
Acción del actor	Respuesta del sistema	
	1 El sistema muestra una interfaz " Crear Configuración beeper "	
2 El administrador introduce los datos correspondientes con la configuración.	3 El sistema valida los datos que estén correctos. 4 El sistema guarda la configuración.	
Flujo Alternativo 1 "Cancelar"		
2.1 El administrador selecciona la opción de "Cancelar".	2.1 Cierra la interfaz " Crear Configuración TCP/IP " sin guardar ningún cambio.	
Flujo Alternativo 2 "Datos incorrectos"		
	3.1 Muestra un mensaje de datos incorrecto y se ejecuta el paso 3.	
Pos condiciones	Crear configuración	

Descripción del caso de uso: Crear configuración SMS.

Caso de Uso:	Crear Configuración SMS
---------------------	--------------------------------

Sistema de notificación de eventos

Actores:	Administrador
Propósito:	Crear una configuración para SMS
Resumen:	El administrador crea una configuración para enviar una notificación haciendo uso del SMS.
Precondiciones:	No se aplica
Referencias:	RF2.1
Prioridad:	Crítica
Flujo Normal de los Eventos:	
Acción del actor	Respuesta del sistema
	1 El sistema muestra una interfaz " Crear Configuración SMS ".
2 El administrador introduce los datos correspondientes con la configuración.	3 El sistema valida los datos que estén correctos. 4 El sistema guarda la configuración.
Flujo Alternativo 1 "Cancelar"	
2.1 El administrador selecciona la opción de "Cancelar".	2.1 Cierra la interfaz " Crear Configuración SMS " sin guardar ningún cambio.
Flujo Alternativo 2 "Datos incorrectos"	
	3.1 Muestra un mensaje de datos incorrecto y se ejecuta el paso 3.
Pos condiciones	Crear configuración

Descripción del caso de uso: Crear configuración correo electrónico.

Caso de Uso:	Crear Configuración Correo Electrónico	
Actores:	Administrador	
Propósito:	Crear una configuración de correo electrónico.	
Resumen:	El administrador crea una configuración de correo electrónico para enviar una notificación por este medio.	
Precondiciones:	No se aplica	
Referencias:	RF2.3	
Prioridad:	Crítica	
Flujo Normal de los Eventos:		
Acción del actor	Respuesta del sistema	
	1 El sistema muestra una interfaz “ Crear Configuración Correo Electrónico ”	
2 El administrador introduce los datos correspondientes con la configuración.	3 El sistema valida los datos que estén correctos. 4 El sistema guarda la configuración.	
Flujo Alternativo 1 “Cancelar”		
2.2 El administrador selecciona la opción de “Cancelar”.	2.1 Cierra la interfaz Crear Configuración Correo Electrónico sin guardar ningún cambio.	
Flujo Alternativo 2 “Datos incorrectos”		

		3.1 Muestra un mensaje de datos incorrecto y se ejecuta el paso 3.
Pos condiciones	Crear configuración	

GLOSARIO DE TÉRMINO

Artefactos

Un artefacto es un producto tangible resultante del proceso de desarrollo de software.

Componente

Objeto o programa reutilizable que efectúa una función específica y que está diseñado para funcionar con otros componentes y aplicaciones.

HTTP

Es un conjunto de reglas acordadas para transferir texto con atributos propios de la Internet. Este protocolo opera a través de solicitudes y respuestas, entre un "cliente" y un "servidor".

Interfaz

Forma o modo en la que interactúan dos programas o las personas y las aplicaciones informáticas.

Metodología

Una metodología es aquella guía que se sigue a fin realizar las acciones propias de una investigación. En términos más sencillos se trata de la guía que nos va indicando qué hacer y cómo actuar cuando se quiere obtener algún tipo de investigación.

Modelo

Se denomina modelo al resultado del proceso de generar una representación abstracta, conceptual, gráfica o visual.

Multiplataforma

Multiplataforma es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Servidor

Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final.

SMS

Son las siglas de Servicio de Mensaje Corto, el que permite enviar y recibir mensajes de texto de hasta 160 caracteres a teléfonos móviles. Además se pueden enviar mensajes cortos a través de Internet, usando un sitio web de SMSs.

Subprograma.

En computación, una subrutina o subprograma (también llamada procedimiento, función o rutina), como idea general, se presenta como un sub algoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica.

REFERENCIA BIBLIOGRÁFICA

1. **Guglielmetti, Marcos.** Master magazine. *Master magazine*. [En línea] 2009. [Citado el: 11 de 12 de 2010.] <http://www.mastermagazine.info/termino/5288.php>.
2. **Alsina, Guillem.** Master magazine. *Master magazine*. [En línea] 2009. [Citado el: 1 de 12 de 2010.] <http://www.mastermagazine.info/termino/6745.php>.
3. **Lanzillota, Analia.** [En línea] 2007. [Citado el: 16 de 11 de 2010.] <http://www.mastermagazine.info/termino/6333.php>.
4. **Yanover, David.** [En línea] <http://www.masadelante.com/faqs/plugin>.
5. El modelo OSI y los protocolos de red. [En línea] [Citado el: 2 de 12 de 2010.] blyx.com/public/docs/pila_OSI.pdf.
6. **Creator, Qt.** Qt Creator. [En línea] http://developer.qt.nokia.com/wiki/Category:Tools::QtCreator_Spanish.
7. Las 10 ventajas principales de Microsoft Office Visio 2007. *Microsoft Office Online*. [En línea] [Citado el: 25 de November de 2008.] <http://office.microsoft.com/es-es/visio/HA101650313082.aspx>.
8. **Mireles, Gabriel Alberto García.** *Introducción a los casos de uso*. Sonora : s.n., 2004.
9. Entorno Virtual de Aprendizaje(EVA). *Entorno Virtual de Aprendizaje(EVA)*. [En línea] [Citado el: 1 de 2 de 2011.] <http://eva.uci.cu/mod/resource/view.php?id=35381>.
10. Mas Adelante <http://www.masadelante.com/faqs/sms>
11. Plataforma de envío de SMS. [En línea] [Citado el: 3 de 12 de 2010.] <http://cri2sms.com/documentos.htm>.
12. **López, Diego M.** *Un Sistema de Vigilancia en Salud Pública*. Colombia : s.n., 2007.
13. Mediosfera. [En línea] <http://mediosfera.wordpress.com/>.
14. Avances tecnológicos. [En línea] [Citado el: 29 de 11 de 2010.] <http://apuntes.rincondelvago.com/avances-tecnologicos.html>.
15. Mensajería Instantanea. [En línea] [Citado el: 27 de 11 de 2010.] <http://www.wikilearning.com/>.

16. Cyta. [En línea] [Citado el: 1 de 12 de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
17. Evolución de las Telecomunicaciones. [En línea] [Citado el: 1 de 12 de 2010.] <http://tribyte.blogspot.com/2007/11/evolucin-de-las-telecomunicaciones.html>.
18. [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.marblestation.com/?p=644>.
19. **Lopez, Jose María Romero.** *Evolucion de las telecomunicaciones en el mundo.* 2008.
20. Solución de aviso para equipos de escritorio. *Solución de aviso para equipos de escritorio.* [En línea] [Citado el: 29 de 11 de 2010.] www.netsupportnotify.com.
21. **Torrez, Gonzalez, Carlos.** *Evolución histórica de las telecomunicaciones.* 2004.
22. Logic Code Soft. *Logic Code Soft.* [En línea] 30 de 8 de 2010. www.logiccodesoft.com.
23. Gtdownload.com. *Gtdownload.com.* [En línea] <http://www.gtdownload.com/es/libre-descargue/Comunicaciones-categoria/Telefonia-subcategoria>.

BIBLIOGRAFÍA

- ✚ Las 10 ventajas principales de Microsoft Office Visio 2007. *Microsoft Office Online*. [En línea] [Citado el: 25 de November de 2008.] <http://office.microsoft.com/es-es/visio/HA101650313082.aspx>.
- ✚ Visual SourceSafe 2005. *Visual Studio*. [En línea] [Citado el: 25 de November de 2008.] <http://msdn.microsoft.com/es-es/vcsharp/aa718670.aspx>.
- ✚ Antonio Albiol, Cristina Sandoval, and Alberto Albiol. Robust motion detector for video surveillande applications. Technical report, Departamento de comunicaciones, Universidad Politécnica de Valencia, 2003.
- ✚ **Lopez, Jose María Romero**. *Evolucion de las telecomunicaciones en el mundo*. 2008.
- ✚ **Mireles, Gabriel Alberto García**. *Introducción a los casos de uso*. Sonora : s.n., 2004.
- ✚ **Lanzillota, Analia**. [En línea] 2007. [Citado el: 16 de 11 de 2010.] <http://www.mastermagazine.info/termino/6333.php>.
- ✚ Plataforma de envío de SMS. [En línea] [Citado el: 3 de 12 de 2010.] <http://cri2sms.com/documentos.htm>.
- ✚ Avances tecnológicos. [En línea] [Citado el: 29 de 11 de 2010.] <http://apuntes.rincondelvago.com/avances-tecnologicos.html>.
- ✚ B.G. Batchelor and P.F. Whelan. Machine visión systems: proverbs, principles, prejudices and priorities. In *Machine Vision Applications, Architectures, and Systemtems Integration III*. Number 2347 in Proceeding of the SIPE, pages 374-383, Boston, 1994. The International Society for optical Engineering.
- ✚ Cyta. [En línea] [Citado el: 1 de 12 de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.