**Universidad de las Ciencias Informáticas**

**Facultad 2**


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

**Título:** Sistema integrado para el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP.

**Autor(es):**

Dianet Díaz Oduardo

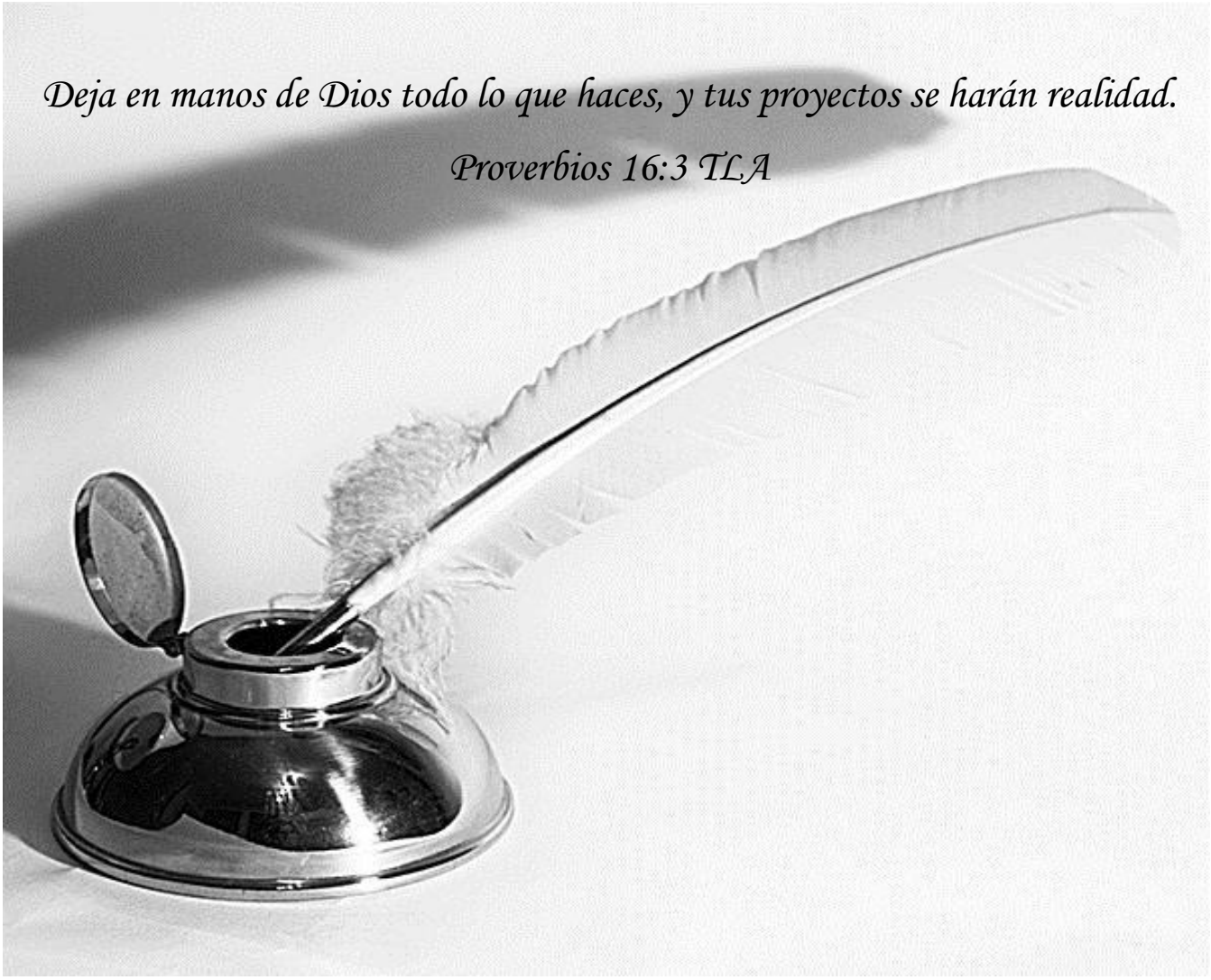
Alejandro García Núñez

**Tutor(es):**

Ing. Erick Pérez Castillo.

Ing. Rainer Segura Peña.

"La Habana, Junio, 2013"


*Deja en manos de Dios todo lo que haces, y tus proyectos se harán realidad.*
*Proverbios 16:3 TLA*

Declaración de autoría

Declaramos ser los autores de la presente tesis, reconociendo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de manera exclusiva.

Para que así conste, firmamos la presente a los ___ días del mes de _____ de _____

| | |
|---|---|
| _____ | _____ |
| Dianet Díaz Oduardo | Alejandro García Núñez |
| Firma Autor | Firma Autor |
| _____ | _____ |
| Ing. Eric Pérez Castillo | Ing. Rainer Segura Peña |
| Firma Tutor | Firma Tutor |

Datos de contacto

**Autor:**

Dianet Díaz Oduardo

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: ddiaz@estudiantes.uci.cu

**Autor:**

Alejandro García Núñez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: agnunez@estudiantes.uci.cu

**Tutor:**

Ing. Eric Pérez Castillo

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: epcastillo@uci.cu

**Tutor:**

Ing. Rainer Segura Peña

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: rsegura@uci.cu

Agradecimientos

***Dianet***

*A Dios por el amor, la paz, sus respuestas y su presencia durante toda mi vida. Que este trabajo sea para su Gloria.*

*A toda mi familia y en especial a mi mamá, Ibrahin y Diana por ser tan lindos y cariñosos. Gracias por toda lo confianza en mí. Los amo.*

*A mi macu por estar en las buenas y en las malas, por su paciencia y amor. Te amo.*

*A mis tutores Erick y Rainer por todo el apoyo y consejos.*

*A la familia de Alejandro por todos los esfuerzos que han hecho por nosotros.*

*A todos los miembros del proyecto en especial a la profesora Arianna.*

*A todos los profesores.*

*A Elaine, Linny, Aile, Reinier, Diannet, Adrian, Wendy y a todos mis compañeros de aula.*

*Y a todos los que de una manera u otra me han ayudado durante todos estos años.*

***Alejandro***

*Agradezco a mi familia, a mis padres, mi hermano por ser el motor impulsor de mi vida.*

*A mi esposa linda y querida por ser la ayuda incondicional en estos años de carrera.*

*A mi prima Yordi.*

*A mi amigo Arian.*

*A mi amigo y hermano Yordan, compañero de tantas batallas en la universidad.*

*A mis compañeros de aula de 1er año, que nunca los olvidé a pesar de la distancia.*

*A mis compañeros de apartamento, Crespo, Ramírez, Fonseca, Jimmy y Luis Angel.*

*A las compañeras de apartamento de mi esposa, Linny, Ailé y Elaine.*

*A mis compañeros de aula.*

*A mis tutores Rainer y Erick doy gracias por toda la ayuda y conocimientos brindados.*

*A mis profesores, Vladimir, Dagoberto, Osmel, Ariel, Arce, Michel, Yaima, Bárbara Triana, Maida Gil, Aimel y Osmany por contribuir a la formación de un mejor estudiante.*

*A los compañeros del proyecto.*

*A todos los que ayudaron en el desarrollo de este trabajo.*

*A la UCI por ser el ente fundamental en mi formación como profesional.*

Dedicatorias

*Dianet*

*A Dios, mi mamá, Ibrahin, Diana, mima, papá y Alejandro. Los amo.*

*Alejandro*

*Dedico este trabajo de diploma a toda mi familia y a mi esposa que son el mayor apoyo en mi vida.*

**Resumen**

El desarrollo de múltiples servicios de comunicación permite establecer diferentes vías de transmisión de la información a todos los usuarios del mundo independientemente del lugar donde se encuentren. Esto facilita la realización de disímiles actividades desde dispositivos electrónicos, como los teléfonos celulares y las computadoras personales. Los usuarios de estos servicios, se sienten altamente atraídos por el nuevo cúmulo de funcionalidades que son integradas en estos dispositivos. Las aplicaciones que se desarrollan actualmente en la Universidad de las Ciencias Informáticas que hacen uso del envío de notificaciones, utilizan estos servicios de comunicación para enviar reportes, eventos, alertas y noticias. Estos sistemas necesitan implementar la comunicación con los servidores o dispositivos de cada uno de estos servicios, desaprovechando tiempo, trabajo y esfuerzo. Este trabajo permite la creación de un sistema que integra el envío de notificaciones vía correo electrónico, mensajería instantánea, servicio de mensajes cortos y llamadas de voz sobre el protocolo de Internet sin tener en cuenta las tecnologías que se utilicen en el desarrollo de estas aplicaciones.

**Palabras clave:**

Desarrollo, comunicación, integra, notificaciones, servicios.

## Índice

Introducción ..... 1

Capítulo 1 Fundamentación teórica ..... 6

1.1 Introducción ..... 6

1.2 Desarrollo ..... 6

1.2.1 Conceptos ..... 6

1.2.1.1 Voz sobre el protocolo de Internet ..... 6

1.2.1.2 Correo electrónico ..... 6

1.2.1.3 Mensajería Instantánea ..... 7

1.2.1.4 Servicio de Mensajes Cortos ..... 7

1.2.1.5 Protocolo ..... 7

1.2.1.6 Listas de Control de Acceso ..... 7

1.2.1.7 Plugin ..... 7

1.2.1.8 Socket ..... 8

1.2.1.9 Archivos log ..... 8

1.2.2 Sistemas integrados para el envío de notificaciones a nivel internacional. ..... 8

1.2.3 Sistemas integrados para el envío de notificaciones a nivel nacional. ..... 9

1.2.4 Conclusiones del estudio de sistemas integrados para el envío de notificaciones a nivel nacional e internacional. ..... 9

1.2.5 Metodología de desarrollo ..... 10

1.2.5.1 Metodologías ágiles ..... 10

1.2.5.2 Programación Extrema (XP) ..... 10

1.2.5.2.1 Selección de la metodología de desarrollo ..... 10

1.2.6 Framework de desarrollo ..... 11

1.2.6.1 Qt....11
1.2.7 Herramientas de desarrollo....12
1.2.7.1 Qt Creator....12
1.2.7.2 Visual Paradigm....13
1.2.8 Sistema Gestor de Bases de Datos (SGBD)....13
1.2.8.1 SQLite....13
1.2.9 Lenguajes de desarrollo....14
1.2.9.1 Lenguaje de programación: C++....14
1.2.9.2 Lenguaje de Etiquetado: XML....15
1.3 Conclusiones....15
Capítulo 2 Características del sistema, exploración y planificación....16
2.1 Introducción....16
2.2 Desarrollo....16
2.2.1 Objeto de automatización....16
2.2.2 Propuesta del sistema....16
2.2.2.1 Servicio de envío de notificaciones.....17
2.2.2.2 Gestión y configuración.....19
2.2.3 Personas relacionadas con el sistema....19
2.2.4 Características del Sistema....19
2.2.4.1 Funcionalidades del sistema....19
2.2.4.2 Lista de reserva del producto....21
2.2.4.2.1 Apariencia o interfaz externa....21
2.2.4.2.2 Usabilidad....22
2.2.4.2.3 Rendimiento....22

2.2.4.2.4 Portabilidad ..... 22
2.2.4.2.5 Seguridad ..... 22
2.2.4.2.6 Disponibilidad ..... 23
2.2.4.2.7 Soporte ..... 24
2.2.4.2.8 Software ..... 24
2.2.4.2.9 Hardware ..... 24
2.2.5 Fase de exploración ..... 25
2.2.5.1 Historias de usuario ..... 25
2.2.6 Planificación ..... 27
2.2.6.1 Estimación de esfuerzo por historias de usuario ..... 27
2.2.6.2 Plan de iteraciones ..... 28
2.2.6.3 Plan de duración de las iteraciones ..... 29
2.2.6.4 Plan de entregas ..... 29
2.3 Conclusiones ..... 30
Capítulo 3 Diseño e implementación del sistema. ..... 31
3.1 Introducción ..... 31
3.2 Desarrollo ..... 31
3.2.1 Arquitectura del sistema ..... 31
3.2.1.1 Arquitectura cliente servidor ..... 31
3.2.1.2 Arquitectura n capas ..... 32
3.2.2 Patrones de diseño ..... 33
3.2.2.1 Patrones para Asignar Responsabilidades (GRASP) ..... 33
3.2.2.1.1 Experto ..... 33
3.2.2.1.2 Creador ..... 34

3.2.2.1.3 Controlador ..... 34
3.2.2.1.4 Alta cohesión ..... 34
3.2.2.1.5 Bajo acoplamiento ..... 34
3.2.3 Tarjetas Clase – Responsabilidad – Colaborador ..... 35
3.2.4 Diagrama de clases persistentes ..... 36
3.2.5 Modelo físico de la base de datos ..... 37
3.2.6 Tareas de Ingeniería ..... 37
3.3 Conclusiones ..... 39
Capítulo 4 Prueba. ..... 40
4.1 Introducción ..... 40
4.2 Desarrollo ..... 40
4.2.1 Pruebas unitarias ..... 40
4.2.2 Pruebas de aceptación ..... 40
4.2.3 Pruebas de carga y estrés ..... 47
4.2.3.1 Pruebas de carga ..... 48
4.2.3.2 Pruebas de estrés ..... 49
4.3 Conclusiones ..... 52
Conclusiones ..... 53
Recomendaciones ..... 54
Referencias bibliográficas ..... 55
Anexo 1 ..... 59
Anexo 2 ..... 60
Anexo 3 ..... 61
Anexo 4 ..... 62

**Índice de figuras**

Figura 1: Arquitectura cliente servidor. ..... 32
Figura 2: Arquitectura n capas ..... 32
Figura 3: Diagrama de clases persistentes ..... 36
Figura 4: Modelo de datos ..... 37
Figura 5: No conformidades ..... 47
Figura 6: Gráfico de resultados de la prueba de carga. ..... 49
Figura 7: Gráfico de resultados de la prueba de estrés número 1. ..... 51
Figura 8: Gráfico de resultados de la prueba de estrés número 2. ..... 52

**Índice de tablas**

Tabla 1: Disponibilidad del sistema. ..... 23
Tabla 2: Requisitos de Software. ..... 24
Tabla 3: Requisitos de Hardware. ..... 24
Tabla 4: HU #1: Aceptar comunicación de aplicaciones clientes. ..... 26
Tabla 5: HU #2: Establecer comunicación con el servidor del servicio. ..... 27
Tabla 6: Estimación por historia de usuario ..... 28
Tabla 7: Plan de duración estimada de las iteraciones. ..... 29
Tabla 8: Plan de entregas. ..... 29
Tabla 9: Tarjeta CRC. Clase: Server_Socket_Manager. ..... 35
Tabla 10: Tarjeta CRC. Clase: Intermediate_Manager. ..... 36
Tabla 11: Tarea de Ingeniería #3: Escuchar y responder a la aplicación cliente. ..... 38
Tabla 12: Tarea de Ingeniería #8: Enviar notificación. ..... 38
Tabla 13: Prueba #1: Aceptar comunicación de aplicaciones clientes. ..... 41
Tabla 14: Prueba #2: Establecer comunicación con el servidor del servicio. ..... 43
Tabla 15: Prueba #3: Enviar notificación. ..... 45
Tabla 16: Recursos necesarios para realizar la prueba de carga. ..... 48
Tabla 17: Recursos necesarios para realizar la prueba de estrés número 1. ..... 50
Tabla 18: Recursos necesarios para realizar la prueba de estrés número 2. ..... 51

Tabla 19: HU #3: Enviar notificación. ..... 59
Tabla 20: HU #4: Iniciar, detener y reiniciar comunicación. ..... 59
Tabla 40: Tarjeta CRC. Clase: Acl_Dao. ..... 60
Tabla 41: Tarjeta CRC. Clase: Trace_Dao. ..... 60
Tabla 42: Tarjeta CRC. Clase: Rule_Dao. ..... 60
Tabla 43: Tarea de Ingeniería #1: Verificar dirección IP de la aplicación cliente. ..... 61
Tabla 44: Tarea de Ingeniería #2: Autenticar aplicación cliente. ..... 61
Tabla 81: Prueba #4: Iniciar comunicación. ..... 62

## Introducción

La comunicación es una de las necesidades fundamentales del hombre, por lo que en el transcurso del tiempo, ha desarrollado diversas formas y vías para interactuar con sus semejantes, independientemente del lugar en que se encuentren. Esto ha traído consigo el surgimiento de las Tecnologías de la Información y las Comunicaciones (TIC).

Las TIC abarcan todos los elementos, métodos, técnicas y herramientas usadas en el tratamiento y transmisión de la información, haciendo uso de la informática y las telecomunicaciones. Las Telecomunicaciones permiten transmitir mensajes desde un lugar a otro, mediante señales, imágenes, voz, sonidos o información, para ello, utilizan diferentes medios de comunicación, seleccionados de acuerdo a su facilidad de uso, costos, capacidad, desempeño, distancia y seguridad.

Uno de los mayores avances en el tema de las TIC y las telecomunicaciones fue la creación de Internet. Su desarrollo ha devenido en el surgimiento de nuevas formas de comunicación que han revolucionado en alguna medida las tradicionales, como la televisión, el correo y el teléfono. Uno de los servicios más difundidos es el correo electrónico, el cual permite a los usuarios enviar y recibir mensajes o cartas electrónicas en un corto tiempo. Haciendo uso de este, los usuarios pueden enviar todo tipo de información como por ejemplo imágenes, audio, video y texto. Otro servicio de gran demanda es el (SMS[1]), por medio de este, los usuarios pueden enviar y recibir desde sus dispositivos telefónicos mensajes cortos. También la mensajería instantánea ha tomado gran popularidad entre los usuarios de la red, posibilitando la comunicación en tiempo real entre dos o más personas. En el caso de la transmisión de datos usando la telefonía, existe el término VoIP[2], que es usado en telefonía IP[3] para definir los servicios que se usan para transmitir voz usando el protocolo IP.

En Cuba desde el triunfo de la Revolución se han llevado a cabo innumerables esfuerzos por lograr la informatización de la sociedad. Ha sido una preocupación prioritaria de la Revolución, la generalización del conocimiento y el acceso a las TIC. Se ha llevado a cabo el desarrollado de diferentes programas

[1] SMS: (Short Message Service o Servicio de Mensajes Cortos)

[2] VoIP: (Voice over Internet Protocol o Voz sobre el Protocolo de Internet)

[3] IP: (Internet Protocol o Protocolo de Internet)

socioeconómicos para aplicar el uso de la informática y las telecomunicaciones a todas las esferas de la sociedad, como la educación, la salud, la medicina, la ciencia, la cultura, el deporte y los servicios comunitarios.

La Universidad de las Ciencias Informáticas (UCI), es la casa de altos estudios más joven de Cuba, lo que no ha sido ningún impedimento para la creación de soluciones informáticas que incrementen el impacto de la informatización del país y las exportaciones de software. La producción de software en la UCI se encuentra dividida en centros productivos. El centro de Telemática (TLM) es el encargado del desarrollo de sistemas y servicios informáticos en las ramas de las Telecomunicaciones y la Seguridad Informática.

Las aplicaciones informáticas que se desarrollan en la UCI que hacen uso del envío de notificaciones, utilizan diferentes vías de comunicación como correo electrónico, mensajería instantánea, SMS y llamadas VoIP para hacer llegar un sinfín de mensajes a disímiles destinos, sin importar cuan distantes estén del emisor y mostrar alertas que hayan sido emitidas por algún sistema o servicio, permitiendo tomar un conjunto de decisiones. La utilización de los servicios de comunicación existentes en la actualidad, permiten que las personas puedan ser contactadas por diferentes vías y en cualquier instante. En dependencia de la situación o del tipo de información que se necesita transmitir, el remitente decide cuál o cuáles medios de comunicación utilizar para realizar el envío de la notificación. Actualmente en la universidad no existe una aplicación que integre estos servicios de comunicación, lo cual trae consigo que las aplicaciones deban implementar su comunicación con cada uno de los servidores o dispositivos de los servicios que empleará, donde se debe conocer de antemano los parámetros necesarios para la conexión como dirección, puerto, usuario y contraseña de cada servidor. Estas utilizan diversas tecnologías y herramientas para el envío de notificaciones adaptadas a su entorno. Por consiguiente, si se necesita realizar otra aplicación que no utilice las mismas tecnologías o herramientas, el trabajo anteriormente realizado no se puede reutilizar.

Debido a esta situación problemática anteriormente expuesta, surge el siguiente **problema a resolver:**

¿Cómo integrar el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP en la UCI, independientemente de las tecnologías que se utilicen en el desarrollo de las aplicaciones?

De acuerdo al problema a resolver, el **objeto de estudio** del presente trabajo de diploma, son los procesos asociados al envío de notificaciones de los servicios de comunicación.

Teniendo en cuenta el problema a resolver, se establece como **objetivo general**, desarrollar un sistema informático que integre el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP en la UCI, independientemente de las tecnologías que se utilicen en el desarrollo de las aplicaciones.

Dentro del objeto de estudio se especifica como **campo de acción**, la integración del envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP en la UCI, independientemente de las tecnologías que se utilicen en el desarrollo de las aplicaciones.

La **idea a defender** plantea que el desarrollo de un sistema informático que integre el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP permitirá que en la UCI la comunicación con los servidores o dispositivos de estos servicios sea independientemente de las tecnologías que se utilicen en el desarrollo de las aplicaciones.

Para dar solución al objetivo propuesto se definen las siguientes **tareas de la investigación:**

1. Realización del estudio de sistemas informáticos que integren el envío de notificaciones para analizar sus características.
2. Selección de la metodología y herramientas para el desarrollo de un sistema integrado para el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP.
3. Diseño e implementación de un modelo para la integración del proceso de envío de notificaciones.
4. Implementación de un plugin para cada uno de los servicios que se utilizarán en el envío de notificaciones.
5. Diseño e implementación de la gestión y configuración del sistema para el control de usuarios, servicios y trazas notificaciones realizadas.
6. Realización de pruebas a cada una de las funcionalidades del sistema para detectar posibles errores.

7. Realización de pruebas de carga y stress para conocer el comportamiento del sistema ante el envío excesivo de peticiones y la ejecución en escenarios de hardware limitados.

Con el objetivo de que la investigación arroje buenos resultados, se emplean los siguientes **métodos de investigación científica:**

**Métodos empíricos:**

- **Entrevista:** Esta técnica se utilizará para la recopilación de información mediante una conversación con profesionales que posean conocimientos relacionados con servicios de comunicación.
- **Simulación:** Utilización de una aplicación cliente con datos artificiales que solicite el envío de notificaciones.

**Métodos lógicos:**

- **Analógico:** Se utilizará para el estudio de características y requisitos de aplicaciones semejantes a la que se desea desarrollar.
- **Analítico:** Se utilizará para estudiar y analizar elementos asociados al proceso de envío de notificaciones de los servicios de comunicación.
- **Sintético**: Se unificarán los elementos estudiados y analizados en relación con el proceso de envío de notificaciones de los servicios de comunicación, para poder crear el sistema.
- **Modelación**: Se crearán modelos y diagramas para representar los componentes, procesos y relaciones que necesitará el sistema.

El trabajo de diploma se divide en 4 capítulos, los cuales estarán estructurados de la siguiente forma:

Capítulo1. "**Fundamentación teórica**": incluye el análisis de teorías, enfoques teóricos, conceptos, investigaciones y antecedentes que se consideran importantes para la confección del trabajo de diploma. Se definirán la metodología y herramientas usadas para el desarrollo de la aplicación.

Capítulo 2. "**Características del sistema, exploración y planificación**": Se identifican procesos existentes en los servicios de comunicación que son empleados en el envío de notificaciones para así lograr posteriormente el diseño e implementación de los mismos.

Capítulo 3. “**Diseño e implementación del sistema**”: se realiza el diseño del sistema donde se describe la arquitectura y patrones seleccionados. Se adapta el diseño al entorno de implementación y se describen las tareas de Ingeniería.

Capítulo 4. “**Prueba**”: se centra principalmente en la evaluación o la valoración de la calidad del producto. Se explicarán las pruebas realizadas al software para comprobar que el producto de software funciona según lo diseñado y que las funcionalidades se han implementado de forma adecuada.

# Capítulo 1 Fundamentación teórica

## 1.1 Introducción

En este capítulo se explican todos los conceptos relacionados con el desarrollo del trabajo de diploma. Se lleva a cabo un estudio acerca de las diferentes herramientas a nivel internacional y nacional, para la creación de un sistema que integre el envío de notificaciones utilizando los servicios de comunicación. Se especifica la metodología de desarrollo escogida, así como las tecnologías, software y lenguajes utilizados para la implementación del sistema.

## 1.2 Desarrollo

### 1.2.1 Conceptos

#### 1.2.1.1 Voz sobre el protocolo de Internet

Conocida como VoIP, define la tecnología que permite encapsular la voz en paquetes para poder ser transportados sobre redes IP sin necesidad de disponer de circuitos conmutados como es el caso de la red de telefonía conmutada. La red convencional se basa en la conmutación de circuitos, es decir, al iniciarse la comunicación se establece un circuito físico durante el tiempo que dura la conversación. Esto implica la reserva de los recursos hasta que finalice la comunicación no pudiendo ser utilizado por otra, incluso durante los silencios que suceden dentro de una conversación típica. En cambio, la telefonía IP no utiliza circuitos físicos para la conversación, sino que envía múltiples conversaciones a través del mismo canal (circuito virtual) codificadas en paquetes y en flujos independientes. Cuando se produce un silencio en una conversación, los paquetes de datos de otras conversaciones pueden ser transmitidos por la red. (1)

#### 1.2.1.2 Correo electrónico

El correo electrónico es un servicio de red que permite a los usuarios enviar y recibir mensajes en fracciones de segundos, con textos, sonidos e imágenes. El emisor envía los mensajes a un servidor y éste, a su vez, se encarga de enviarlos al servidor del receptor. Para poder ver el correo electrónico es necesario que el receptor se conecte con su servidor. Se puede enviar el mensaje a uno o varios remitentes al mismo tiempo, con dirección visible o encriptado, con listas de distribución públicas o privadas. (2)

#### 1.2.1.3 Mensajería Instantánea

La mensajería instantánea conocida también en inglés como IM[4], permite conversar con otra persona mediante mensajes instantáneos, unirse a foros de debate con temas específicos, enviar ficheros y comunicarse con otros usuarios en tiempo real sin importar las distancias. (2)

#### 1.2.1.4 Servicio de Mensajes Cortos

Conocido por la sigla SMS, con este servicio se puede enviar mensajes de texto desde cualquier móvil o aplicación a otro, sin importar de qué compañía sea. Si el teléfono al que se envía el mensaje está apagado o fuera de cobertura el mensaje se almacena en la red y se entrega en cuanto el teléfono se conecta de nuevo a la red. (2)

#### 1.2.1.5 Protocolo

Un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red por medio de intercambio de mensajes. Es una regla o estándar que controla o permite la comunicación, puede ser definido como las reglas que dominan la sintaxis, semántica y sincronización de la comunicación. Los protocolos pueden ser implementados por hardware, software o una combinación de ambos. A su más bajo nivel, éste define el comportamiento de una conexión de hardware. (3)

#### 1.2.1.6 Listas de Control de Acceso

Una lista de control de acceso o ACL es un concepto de Seguridad Informática usado para fomentar la separación de privilegios. Es una forma de determinar los permisos de acceso apropiados a un determinado objeto, dependiendo de ciertos aspectos del proceso que hace el pedido.

Las ACL permiten controlar el flujo del tráfico en equipos de redes, tales como enrutadores y conmutadores. Su principal objetivo es filtrar tráfico, permitiendo o denegando el tráfico de red de acuerdo a alguna condición. (4)

#### 1.2.1.7 Plugin

Pequeño programa que añade alguna función a otro programa, habitualmente de mayor tamaño. Son muy utilizados en los navegadores Web[5] como Mozilla Firefox para ampliar sus funcionalidades. (5)

[4] IM: (Instant Messaging o Mensajería Instantánea)

#### 1.2.1.8 Socket

Un socket es un mecanismo que permite la conexión entre distintos procesos, habitualmente se utilizan para establecer comunicaciones entre diferentes máquinas que estén conectadas a través de la red. Cuando se utilizan sockets para comunicar procesos, se emplea la arquitectura cliente servidor. Así pues, se establecen dos sockets, uno será la parte servidor que recibirá la transmisión del cliente y otro será la parte cliente que recibirá la respuesta del servidor. (6)

#### 1.2.1.9 Archivos log

Log Files, se refiere al archivo que registra toda la actividad de un servidor, aplicación o software. El mismo es presentado cronológicamente con datos adicionales muy detallados que se utilizan generalmente para llevar estadísticas de uso de un determinado sitio, aplicación o software. (7)

### 1.2.2 Sistemas integrados para el envío de notificaciones a nivel internacional.

AlertFind es un software para el envío de notificaciones que la compañía Dell Incorporated brinda por suscripción. Tiene como objetivos automatizar y facilitar el uso de los servicios de comunicación, las operaciones y las alertas comerciales. Mediante AlertFind los clientes pueden enviar mensajes por medio de varios servicios de comunicación definidos en el perfil de notificación de cada destinatario. Entre estos canales de comunicación se incluyen llamadas de voz a teléfonos fijos o móviles mediante TTS (texto a voz) o funciones de voz grabada, SMS, correo electrónico y fax. Incluye aceptaciones e informes de estado en tiempo real, autenticación de usuarios y alertas a dispositivos alternativos o a usuarios. (8)

AlertFind presenta auditorías en tiempo real, las cuales cuentan con un registro detallado de transacciones y la información sobre las notificaciones pendientes y enviadas. Los informes ofrecen resúmenes simples y descripciones específicas del proceso de notificación a un usuario. Permite tener un control de acceso para saber quién inicia el envío de mensajes, con quien puede ponerse en contacto y lo que los destinatarios pueden ver.

[5] Navegadores Web: Programas utilizados para navegar por Internet. Algunos de los más populares son, Mozilla Firefox e Internet Explorer.

### 1.2.3 Sistemas integrados para el envío de notificaciones a nivel nacional.

El Sistema de Integración de Servicios de Comunicación a Usuarios (SISCU) es un software creado en la Universidad de las Ciencias Informáticas con el propósito de disminuir el tiempo y esfuerzo de los usuarios al enviar mensajes. Las notificaciones pueden ser enviadas por los diferentes medios de comunicación interpersonales existentes en la UCI como correo electrónico, mensajería instantánea, telefonía fija y SMS.

El sistema es una aplicación web a la cual puede acceder cualquier usuario previamente autenticado. Permite el envío de mensajes y brinda la posibilidad de gestionar los clientes y contactos que posee cada usuario. Esto facilita el trabajo de los mismos, ya que no tendrán que utilizar cada una de las aplicaciones que posibilitan el envío de mensajes de forma individual, sino que se presentan todas en un único sistema. (2)

### 1.2.4 Conclusiones del estudio de sistemas integrados para el envío de notificaciones a nivel nacional e internacional.

Después de realizar un estudio de las soluciones existentes, se llegó a la conclusión de que:

- AlertFind muestra un conjunto de características que se pueden tener en cuenta para el desarrollo del sistema. Es el caso de, permitir enviar notificaciones haciendo uso de diferentes servicios de comunicación. Obtener informes y reportes detallados de las notificaciones de los usuarios. Tener un control de acceso a usuarios lo que posibilita regular determinadas acciones de los mismos.

  En el presente trabajo, no se puede hacer uso de esta herramienta para dar solución al problema a resolver, debido a que no admite la integración de nuevos servicios de comunicación al sistema, ya que estos están predefinidos. No incluye en envío de mensajes por medio de la mensajería instantánea. Además es un software propietario, por lo que su uso va en contra de las proyecciones del país.

- SISCU presenta características que pueden ser tomadas como base para el desarrollo del sistema. Es el caso de la integración de múltiples servicios de comunicación.

En el presente trabajo, no se puede hacer uso de esta herramienta para dar solución al problema a resolver, debido a que no permite la integración de nuevos servicios de comunicación ya que los mismos están predefinidos. Este no permite que aplicaciones clientes se comuniquen con él para enviar notificaciones. Además debido a problemas tecnológicos de la facultad, el producto resultante de esta investigación no existe.

### 1.2.5 Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores, guiando la realización del nuevo software. (9)

#### 1.2.5.1 Metodologías ágiles

Para las metodologías ágiles los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados en el desarrollo del software. Es más relevante crear un software que funcione, que escribir documentación exhaustiva. La colaboración con el cliente debe prevalecer sobre la negociación de contratos. La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan. (10)

#### 1.2.5.2 Programación Extrema (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, en la comunicación fluida entre todos los participantes, en la simplicidad de las soluciones implementadas y en la voluntad para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. (10)

##### 1.2.5.2.1 Selección de la metodología de desarrollo

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software, toda metodología debe ser adaptada al contexto del proyecto. Para el desarrollo de este sistema informático, se consideraran las siguientes características:

**Características del equipo de desarrollo**

- El equipo está compuesto por dos programadores y un cliente.
- El grado de interacción entre los miembros del equipo es alto, pues existe una gran compenetración y un buen entendimiento entre ellos.

**Características del cliente**

- El cliente pertenece al equipo de desarrollo.
- El intercambio de conocimientos entre el cliente con los miembros del equipo es alto, esto se debe a que el cliente pertenece al equipo de desarrollo.

**Características del proyecto a desarrollar**

- El tiempo del proyecto es corto, se debe terminar en menos de un año.
- Los requisitos del proyecto pueden cambiar, pues el cliente puede agregar nuevas historias de usuarios según avanza el proyecto.
- La generación de artefactos y roles es poca, ya que el tiempo de desarrollo del proyecto es corto.

Según el estudio realizado, se decide utilizar XP como metodología de desarrollo.

### 1.2.6 Framework de desarrollo

#### 1.2.6.1 Qt

Qt[6] es un framework de desarrollo integral con herramientas diseñadas para simplificar la creación de aplicaciones e interfaces de usuario para sistemas de escritorio, embebidos y plataformas móviles. Fue creado por la compañía Trolltech y actualmente es propiedad de Nokia. El código creado con Qt, se puede reutilizar de manera eficiente en múltiples sistemas operativos como Windows, Linux y Mac OS X y en algunos para dispositivos celulares como Symbian y Nokia. Permite a los desarrolladores, crear aplicaciones para una plataforma y fácilmente generar y ejecutar el despliegue en otra. (11) Facilita ciertas tareas de programación como el manejo de sockets, soporte para programación multihilo, comunicación con bases de datos, manejo de cadenas de caracteres, entre otras. Ofrece una suite de aplicaciones para facilitar y agilizar las tareas de desarrollo, estas son: (12)

[6] QT: (Quasar Technologies o Tecnologías Quasar)

- Qt Assistant: Herramienta para visualizar la documentación oficial de Qt.
- Qt Designer: Herramienta WYSIWYG[7] para crear interfaces de usuario.
- Qt Linguist: Herramienta para la traducción de aplicaciones.
- Qt Creator: IDE[8] para el lenguaje C++, pero especialmente diseñado para Qt, integra las primeras dos herramientas mencionadas.

### 1.2.7 Herramientas de desarrollo

#### 1.2.7.1 Qt Creator

Es un IDE que proporciona las herramientas necesarias para diseñar y desarrollar aplicaciones con una biblioteca de desarrollo de Interfaz Gráfica de Usuario[9]. Facilita herramientas para llevar a cabo sus tareas a lo largo de todo el ciclo de vida del desarrollo de software. Una de las mayores ventajas de Qt Creator es que permite a un equipo de desarrolladores compartir proyectos a través de diferentes plataformas de desarrollo como Linux, Mac OS X y Windows con una herramienta común para desarrollar. (13)

El objetivo principal de Qt Creator es satisfacer las necesidades de desarrollo de los programadores de Qt, que buscan simplicidad, facilidad de uso, productividad, extensibilidad y apertura mientras se reduce la barrera de entrada para los que se inician en Qt. (13)

Se decide usar Qt Creator 2.4.1 basado en QT 4.7.4 como herramienta de desarrollo por las características anteriormente mencionadas. Además permite desarrollar aplicaciones con el avanzado editor C++. Provee un plugin para depurar que actúa como interfaz entre el núcleo de Qt Creator y depuradores externos nativos como GNU[10] Symbolic Debugger (GDB), Microsoft Console Debugger (CDB) y el depurador de QML/JavaScript. Utiliza herramientas de análisis de código para comprobar si

[7] WYSIWYG: (What You See Is What You Get o Lo que ves es lo que obtienes). Se refiere a que cuando trabajas con esta herramienta, la interfaz que ves que estás creando es lo que obtienes cuando culminas.

[8] IDE: (Integrated Development Environment o Entorno de Desarrollo Integrado)

[9] Interfaz Gráfica de Usuario: (GUI o Graphical User Interface). Permite la interacción de usuarios con aplicaciones basadas en elementos visuales como ventanas, iconos, menús.

[10] GNU:(GNU is Not Unix o GNU no es Unix)

existen problemas de gestión de memoria en las aplicaciones. Brinda acceso a información mediante un módulo contextual de ayuda. (13)

#### 1.2.7.2 Visual Paradigm

Visual Paradigm es una herramienta CASE[11] que soporta el ciclo de vida completo y facilita el modelado de artefactos en un proceso de desarrollo de software. Permite una rápida construcción de aplicaciones de mejor calidad y a un menor coste. Permite dibujar diferentes tipos de diagramas de clases, generar documentación y código desde diferentes diagramas, facilita la exportación e importación de diversos archivos de diferentes formatos. (14) Es una herramienta que puede ser utilizada por ingenieros de software, analistas de sistemas y arquitectos de sistemas que están interesados en la construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. (15)

Se decide utilizar Visual Paradigm en su versión 5 como herramienta de modelado, por las características anteriormente mencionadas. Además es un software multiplataforma que permite transformar diagramas Entidad-Relación en tablas de base de datos, se integra fácilmente con diferentes lenguajes de programación como C++, PHP, C#, Java, entre otros.

### 1.2.8 Sistema Gestor de Bases de Datos (SGBD)

#### 1.2.8.1 SQLite

SQLite es una biblioteca que implementa en sí misma, sin servidor, y sin necesidad de configuración, un motor de base de datos SQL transaccional. El código de SQLite es de dominio público y por lo tanto de libre uso y para cualquier propósito, comercial o privado.

A diferencia de la mayoría de otras bases de datos SQL, SQLite no tiene un proceso servidor independiente. Es una completa base de datos SQL con varias tablas, índices, triggers (disparadores) y vistas, que está contenida en un archivo donde se lee y escribe directamente.

El tamaño de la biblioteca SQLite puede ser inferior a 350KiB, dependiendo de la plataforma de destino y la configuración de optimización del compilador. En SQLite existe un equilibrio entre el uso de memoria y velocidad. En general, funciona más rápido, mientras mayor cantidad de memoria se le asigne. Sin embargo, el rendimiento suele ser bastante bueno, incluso en entornos con poca memoria. (16)

[11] CASE:(Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora)

Se decide utilizar SQLite como Sistema Gestor de Bases de Datos por lo explicado anteriormente y además porque es compatible con múltiples plataformas como: Unix (Linux y Mac OS X), OS / 2 y Windows, al ser multiplataforma es fácil de portar el archivo de base datos entre estos sistemas. El tamaño máximo teóricamente soportado por SQLite, es de 2 Terabytes, dicha capacidad es suficiente para los datos que el sistema necesita guardar.

### 1.2.9 Lenguajes de desarrollo

#### 1.2.9.1 Lenguaje de programación: C++

C++ es un lenguaje de propósito general basado en el lenguaje de programación C. A diferencia de su antecesor este ha añadido nuevos tipos de datos, clases, plantillas, funciones virtuales, sistema de espacios de nombres, sobrecarga de operadores, referencias, sobrecarga de funciones, operadores para manejo de memoria y algunas utilidades adicionales basadas en librerías externas. (17)

C++ es un lenguaje versátil y potente en el desarrollo de aplicaciones. Una vez incorporada la programación genérica, reúne otros paradigmas de programación (programación estructurada y orientada a objetos); por lo que se puede considerar como un lenguaje que soporta múltiples paradigmas de programación. Algunas de las principales ventajas que posee este lenguaje son la versatilidad y la portabilidad, pues está estandarizado y un mismo código fuente se puede compilar en diversas plataformas. Muchos sistemas operativos, compiladores e intérpretes han sido escritos en C++ (Windows, Java). (18)

Se decide utilizar C++ como lenguaje de programación por todas las características anteriormente expuestas. Además el framework de desarrollo Qt, utiliza C++ como lenguaje de desarrollo, por tanto constituye un denominador común entre ambos.

#### 1.2.9.2 Lenguaje de Etiquetado: XML

XML[12] es un metalenguaje extensible de etiquetas desarrollado por el “World Wide Web Consortium” (W3C). Es un formato basado en texto, específicamente diseñado para almacenar y transmitir datos. Un documento XML se compone de elementos, cada uno de los cuales consta de una etiqueta de inicio, una de fin y de los datos comprendidos entre estas. Se encarga de definir el contenido y solo utiliza etiquetas para describir los datos. Es extensible, debido a que en XML se pueden definir un conjunto ilimitado de etiquetas. Los desarrolladores pueden definir sus propias estructuras de datos de acuerdo a sus necesidades. Permite el intercambio de documentos entre aplicaciones tanto en la propia PC como en una red local o extensa. (19)

Se decide utilizar XML porque las características antes mencionadas permiten que se pueda utilizar para compartir información entre aplicaciones clientes y el servidor del sistema.

## 1.3 Conclusiones

En el desarrollo de este capítulo se analizaron los principales conceptos a emplear en el desarrollo del trabajo. Se estudiaron sistemas de envío de notificaciones existentes a nivel nacional e internacional, utilizados por los servicios de comunicación. Se escogió como metodología de desarrollo XP y como lenguaje de programación C++. Dentro de las herramientas de desarrollo, se seleccionó como entorno de desarrollo Qt Creator y como herramienta de modelado Visual Paradigm. También se eligió SQLite como SGBD.

[12] XML: (eXtensible Markup Language o Lenguaje de Marcas Extensible)

# Capítulo 2 Características del sistema, exploración y planificación

## 2.1 Introducción

En este capítulo se mostrará cómo debe de funcionar el sistema, se realizará una descripción general de las historias de usuario que se proponen para dar solución a los problemas que originaron la situación problemática. Se describirán las características funcionales con que debe contar la aplicación y la lista de reserva de producto. También se construirá el plan de entregas del producto.

## 2.2 Desarrollo

### 2.2.1 Objeto de automatización

Se pretende automatizar el envío de notificaciones de diferentes servicios de comunicación como por ejemplo: correo electrónico, mensajería instantánea, SMS y llamadas VoIP. Se automatizará también, la administración y configuración de diferentes procesos que intervienen en el sistema, como la gestión de usuarios, trazas y servicios.

### 2.2.2 Propuesta del sistema

El sistema está diseñado para permitir la conexión de aplicaciones clientes y que estas a su vez puedan enviar notificaciones, alertas, mensajes, noticias y reportes, utilizando diferentes medios de comunicación como correo electrónico, SMS, mensajería instantánea, llamadas VoIP u otros que se deseen adicionar. Los desarrolladores que creen estas aplicaciones, solo tienen que enviar los datos necesarios de las notificaciones correspondientes al sistema y este se encargará de remitirlas mediante los servicios solicitados.

Por otra parte, el sistema también debe permitir la gestión y configuración de determinados aspectos relacionados con el servidor.

**El sistema está compuesto por los siguientes subsistemas:**


### 2.2.2.1 Servicio de envío de notificaciones.

La comunicación comienza, cuando una aplicación cliente se conecta con el sistema para enviar una notificación. Se verifica que la dirección IP desde donde se están tratando de conectar es permitida, si no lo es, se le notifica al cliente que no puede conectarse y se rechaza la conexión. Si la dirección es admitida, se solicita el usuario y la contraseña a la aplicación cliente, si los datos son incorrectos, se indica que existe un error de autenticación. Si son correctos, se verifica que la aplicación cliente tenga permitido enviar notificaciones desde esa IP. Si no tiene permiso, se informa que no puede enviar notificaciones y se rechaza la conexión. Si tiene permiso, se comprueba que la aplicación cliente puede utilizar el servicio para el envío de la notificación. Si lo puede utilizar, se envía la notificación, en caso contrario, se notifica al cliente.

**El proceso de envío de notificaciones se realiza del siguiente modo:**

Una vez que las notificaciones son enviadas desde las aplicaciones clientes al sistema, este se encarga de enviar cada notificación mediante el servicio que fue utilizado para enviar la notificación.

**Notificaciones enviadas mediante:**

**Correo:** El sistema utiliza un plugin que es el encargado de comunicarse con un servidor de correo (smtp.uci.cu), a través de este plugin, se envían los datos al servidor y este envía la notificación.

**Mensajería Instantánea:** El sistema utiliza un plugin que es el encargado de comunicarse con un servidor jabber (jabber@uci.cu), a través de este plugin, se envían los datos al servidor y este envía la notificación.

**SMS:** El sistema utiliza un plugin que es el encargado de comunicarse con un modem GSM, a través de este plugin, el sistema le envía los datos al modem y este envía la notificación.

**VoIP:** El sistema utiliza un plugin que es el encargado de comunicarse con la planta telefónica Asterisk[13], a través de este plugin, el sistema le envía los datos a Asterisk, en este caso son números telefónicos. Asterisk utiliza una aplicación llamada Dial[14] para comunicar los dos números telefónicos mediante un protocolo de señalización de VoIP. El protocolo utilizado en este caso es el protocolo SIP[15]. La aplicación Dial realiza una llamada a un determinado destino, si el destino acepta la llamada, Asterisk conecta el origen primario de la llamada con este nuevo número telefónico.

**Otro servicio:** El sistema es capaz de incluir nuevos servicios de comunicación, los cuales se ajustarían a las necesidades del sistema para realizar el envío de notificaciones. Necesidades como:

- ✓ Abrir conexión.

  Permite conectarse con el servicio solicitado.

- ✓ Autenticación.

  Permite autenticarse en el servicio solicitado.

- ✓ Nombre del servicio.

  Permite conocer el nombre del servicio por el cual se desea enviar la notificación.

- ✓ Enviar mensaje.

  Permite enviar la notificación por el servicio solicitado.

- ✓ Cerrar conexión.

  Cierra la conexión con el servicio solicitado.

En el caso de abrir conexión y de autenticarse, se utilizan cuentas existentes en estos servicios. Por ejemplo en el caso de correo se utiliza una cuenta registrada en el servidor de correo. Esto quiere decir

[13] Asterisk: Software que permite convertir un ordenador de propósito general, en un servidor de comunicaciones VoIP.

[14] Dial: Aplicación de Asterisk que permite comunicar dos números telefónicos.

[15] SIP: (Session Initiation Protocol o Protocolo de Inicio de Sesiones): Es utilizado para el control de multimedia, sesiones de comunicación , tales como voz y vídeo llamadas a través del protocolo de internet.

que las aplicaciones clientes, no tienen que enviar datos del servidor (puerto, servidor, contraseña) donde están registradas sus cuentas, solo será necesario el servicio, el remitente, el destinatario y el mensaje.

#### 2.2.2.2 Gestión y configuración.

Este subsistema permite a los administradores, realizar un seguimiento, tener un control, configurar y conocer, todo lo relativo al sistema. Estos iniciarán el servidor para que las aplicaciones clientes puedan realizar el envío de las notificaciones, podrán obtener trazas de los diferentes clientes, permitiendo conocer por ejemplo: que servicios de notificación son más utilizados, que clientes tienen acceso al sistema y mediante que vía, estos realizan sus notificaciones. Los administradores podrán establecer permisos de acceso, dependiendo del servicio de comunicación que se desee emplear y de acuerdo a la dirección IP de los clientes.

El acceso al sistema es mediante un proceso de autenticación. Si este paso es correcto, se podrán gestionar servicios, usuarios, trazas, archivos log y listas de control de acceso mediante un menú donde se selecciona la opción de la operación que desea realizar y el sistema le muestra la ventana correspondiente. El administrador introduce los datos necesarios y el sistema verifica que sean válidos, si son válidos, realiza la operación y en caso contrario, le notifica al administrador que los datos son incorrectos.

### 2.2.3 Personas relacionadas con el sistema

| Usuario | Justificación |
|---|---|
| Administrador | Es el encargado de la gestión de servicios, usuarios, trazas y archivos log y de la configuración del servidor. |

### 2.2.4 Características del Sistema

#### 2.2.4.1 Funcionalidades del sistema

- **Gestionar comunicación.**
  - ✓ Aceptar comunicación de aplicaciones clientes.
  - ✓ Establecer comunicación con el servidor del servicio.

- ✓ Enviar notificación.
- ✓ Iniciar comunicación.
- ✓ Detener comunicación.
- ✓ Reiniciar comunicación.

- **Gestionar servicios.**
  - ✓ Insertar servicio.
  - ✓ Activar servicio.
  - ✓ Desactivar servicio.
  - ✓ Mostrar servicios.
  - ✓ Modificar servicio.
  - ✓ Mostrar servicios más utilizados.
- **Gestionar usuarios.**
  - ✓ Insertar usuario.
  - ✓ Buscar usuario
  - ✓ Eliminar usuario.
  - ✓ Modificar usuario.
  - ✓ Mostrar usuarios.
  - ✓ Asignar servicio al usuario.
  - ✓ Denegar servicio al usuario.
  - ✓ Autenticar usuario.
  - ✓ Mostrar usuarios de más notificaciones.
- **Gestionar listas de control de acceso.**
  - ✓ Insertar ACL.

- ✓ Mostrar ACL.
- ✓ Buscar ACL.
- ✓ Insertar regla.
- ✓ Eliminar regla.
- ✓ Modificar regla.

➢ **Gestionar trazas.**

- ✓ Insertar traza.
- ✓ Mostrar trazas.
- ✓ Buscar trazas.

➢ **Gestión de archivos log.**

- ✓ Guardar datos en el archivo log.
- ✓ Mostrar datos guardados en el archivo.

➢ **Gestionar configuración.**

- ✓ Copiar plugin.
- ✓ Guardar puerto y número máximo de conexiones del servidor.
- ✓ Mostrar configuración.

#### 2.2.4.2 Lista de reserva del producto

##### 2.2.4.2.1 Apariencia o interfaz externa

- **Interfaz de usuario**

  La información en esta interfaz, está organizada por menús, los cuales indican las opciones y acciones que pueden realizar los usuarios. A través de esta interfaz se pueden acceder a otras interfaces, las cuales permiten configurar y gestionar el sistema.

- **Interfaz de hardware**

Se necesita una tarjeta de red de 100 Mbps como mínimo.

- **Interfaz de software**

  El sistema utiliza SQLite3 en su versión 5.7.13 para realizar las conexiones con la base de datos.

#### 2.2.4.2.2 Usabilidad

Para la gestión y configuración del sistema el administrador debe de tener al menos conocimientos mínimos de administración de redes. En caso de que los usuarios no posean estos conocimientos, no necesitarán mucha capacitación.

#### 2.2.4.2.3 Rendimiento

El sistema debe responder en el mínimo de tiempo posible, ante las solicitudes de información y notificación por parte de las aplicaciones clientes y usuarios predeterminados. La eficiencia de la aplicación estará determinada en gran medida por la velocidad de las consultas a la base de datos, la rapidez en el envío de las notificaciones y las características de la computadora donde se encuentre instalado el sistema.

#### 2.2.4.2.4 Portabilidad

El sistema podrá ser empleado en plataformas Windows y Linux, pero deberá ser compilado en cada una de ellas al menos una vez. Para su correcto funcionamiento, las diferentes plataformas de sistemas operativos o las diversas computadoras donde sea instalado, deben contar con determinadas características asociadas al Framework de desarrollo Qt.

#### 2.2.4.2.5 Seguridad

El sistema empleará técnicas criptográficas para proteger los usuarios, las contraseñas y para verificar la integridad de los mensajes que llegan al servidor. Las contraseñas de las aplicaciones clientes, no viajarán en texto plano hacia el servidor. Para la encriptación de estos datos se usará MD5[16], como algoritmo de encriptación.

[16] MD5: (Message-Digest Algorithm 5 o Algoritmo de Resumen del Mensaje 5)

La aplicación guardará en archivos log todos los eventos que ocurran en el proceso de envío de notificaciones. Los mismos serán presentados cronológicamente.

El sistema empleará listas de control de acceso, para evitar y bloquear accesos indebidos de aplicaciones clientes.

#### 2.2.4.2.6 Disponibilidad

La aplicación debe estar disponible las 24 horas del día, excepto los días que se le realicen auditorías al sistema informático. Estas se realizarán una vez al mes, con el objetivo de extraer información y datos importantes relacionados con la actividad del sistema, de los usuarios, de los servicios y del envío de notificaciones.

**Determinación del porciento de disponibilidad del sistema.**

Teniendo en cuenta que el sistema detendrá sus servicios una vez por mes, el porciento de disponibilidad del mismo es:

SLA[17] del sistema es: 24*365=8760

Disponibilidad = ((A – B)/A) x 100)

Dónde:

- **SLA→** Acuerdo de Nivel de Servicio
- **A** = **SLA**
- **B=TFS** → Tiempo Fuera de Servicio
- **D→** Disponibilidad del sistema

Así entonces:

- D = ((8,760 – 288)/8,760) x 100) = 96.7%

**Tabla 1: Disponibilidad del sistema.**

| % Disponibilidad | TFS/Año | TFS/Mes | TFS/Día | Información |
|---|---|---|---|---|

[17]SLA: (Service Level Agreement): Nivel de acuerdo a que se llega para determinar la disponibilidad de un sistema.

| | | | | |
|---|---|---|---|---|
| **96.7%** | 12días / 288hs | 24hs | 24hs | Una auditoría por mes |

#### 2.2.4.2.7 Soporte

La aplicación contará con un manual de ayuda, con el objetivo de explicar cómo funciona el servidor para el envío de notificaciones, cómo se realiza la gestión y configuración del sistema y cómo se pueden conectar las aplicaciones clientes con el servidor, garantizando el soporte de la herramienta.

Los servicios de mantenimiento solo serán realizados por el personal calificado para estos fines, en este caso los administradores del sistema.

#### 2.2.4.2.8 Software

El sistema podrá ser utilizado en sistemas operativos, Windows o Linux.

**Tabla 2: Requisitos de Software.**

| | **Especificaciones** |
|---|---|
| Sistema | • Sistema Operativo: Windows o Linux<br>• Asterisk |

La plataforma telefónica Asterisk no necesariamente debe estar situada en la misma computadora para realizar las llamadas telefónicas VoIP.

#### 2.2.4.2.9 Hardware

Para la instalación del sistema, es necesario contar con una computadora. Esta debe tener las siguientes características:

**Tabla 3: Requisitos de Hardware.**

| | **Especificaciones** |
|---|---|
| Sistema | • **RAM:** 1 GB o superior.<br>• **Disco Duro:** 160 GB o superior.<br>• **UPS:** 1.<br>• **CPU:** 2.0 GHz o superior |

| | |
|---|---|
| Módem GSM | • Cantidad: 1 o más. |

El módem GSM deberá estar situado en la misma computadora para poder realizar el envío de SMS.

### 2.2.5 Fase de exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. (20)

#### 2.2.5.1 Historias de usuario

Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. (10)

Las HU serán representadas mediante tablas divididas por las siguientes secciones:

- ✓ **Número:** número de la historia de usuario incremental en el tiempo.
- ✓ **Nombre de historia de usuario:** el nombre de la historia de usuario sería para identificarlas mejor entre los desarrolladores y el cliente.
- ✓ **Modificación de historia de usuario número:** si sufrió alguna modificación anterior.
- ✓ **Usuario:** involucrados en el desarrollo de la HU.
- ✓ **Iteración asignada:** número de la iteración.
- ✓ **Prioridad en negocio**:
    - Las historias de usuarios que son de funcionalidades imprescindibles en el desarrollo del sistema tienen prioridad alta.
    - Las historias de usuarios que son de funcionalidades que debe de tener el sistema, pero que no son necesarias para su funcionamiento, tienen prioridad media.

- Las historias de usuarios que son de funcionalidades auxiliares y que son independientes del sistema, tienen prioridad baja.

✓ **Riesgo en desarrollo*:***

- Las historias de usuarios que en caso de tener algún error de implementación, puedan afectar la disponibilidad del sistema, tienen riesgo de desarrollo alto.
- Las historias de usuarios que puedan presentar errores y retrasan la entrega de la versión, tienen riesgo de desarrollo medio.
- Las historias de usuario que puedan presentar errores, pero estos son tratados con facilidad y no afectan en desarrollo del proyecto, tienen riesgo de desarrollo bajo.

✓ **Puntos estimados**: tiempo estimado que se demorará el desarrollo de la HU.

✓ **Puntos reales**: tiempo que se demoró en realidad el desarrollo de la HU.

✓ **Descripción**: breve descripción de la HU.

✓ **Observaciones**: señalamiento o advertencia del sistema.

✓ **Prototipo de interfaz**: Prototipo de interfaz si aplica.

**Descripción de las historias de usuario**

A continuación se presentan las historias de usuarios más importantes, las otras se encuentran en el Anexo 1:

**Tabla 4: HU #1: Aceptar comunicación de aplicaciones clientes.**

| **Historia de usuario** | |
|---|---|
| **Número**: 1 | **Nombre de historia de usuario:** Aceptar comunicación de aplicaciones clientes |
| **Modificación de historia de usuario número:** Ninguna | |
| **Usuario:** Dianet Díaz Oduardo, Alejandro García Núñez | **Iteración asignada:**1 |
| **Prioridad en negocio**: Alta | **Puntos estimados**: 3 |
| **Riesgo en desarrollo:** Alto | **Puntos reales:** 3 |
| **Descripción**: La aplicación cliente solicita conectarse al sistema y el sistema acepta la solicitud de conectarse. | |
| **Observaciones:** El sistema acepta la comunicación, si la dirección IP, el usuario y la contraseña de la | |

| |
|---|
| aplicación cliente son válidos. |
| **Prototipo de interfaz:** No aplica |

**Tabla 5: HU #2: Establecer comunicación con el servidor del servicio.**

| **Historia de usuario** | |
|---|---|
| **Número**: 2 | **Nombre de historia de usuario:** Establecer comunicación con el servidor del servicio |
| **Modificación de historia de usuario número:** Ninguna | |
| **Usuario:** Dianet Díaz Oduardo, Alejandro García Núñez | **Iteración asignada:** 1 |
| **Prioridad en negocio**: Alta | **Puntos estimados**: 3 |
| **Riesgo en desarrollo:** Alto | **Puntos reales:** 3 |
| **Descripción**: El sistema solicita conectarse al servidor del servicio, que necesita la notificación enviada por la aplicación cliente. | |
| **Observaciones:** El sistema debe autenticarse correctamente en el servidor del servicio correspondiente. | |
| **Prototipo de interfaz:** No aplica | |

### 2.2.6 Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. (20)

#### 2.2.6.1 Estimación de esfuerzo por historias de usuario

A continuación se muestra la estimación del esfuerzo por cada HU propuesta para el desarrollo de la aplicación:

**Tabla 6: Estimación por historia de usuario.**

| No | Historias de Usuario | Punto de Estimación |
|---|---|---|
| 1 | Aceptar comunicación de aplicaciones clientes. | 3 |
| 2 | Establecer comunicación con el servidor del servicio. | 3 |
| 3 | Enviar notificación. | 1 |
| 4 | Iniciar, detener y reiniciar comunicación | 1 |
| 5 | Gestionar servicios. | 3 |
| 6 | Gestionar usuarios. | 3 |
| 7 | Gestionar listas de control de acceso. | 3 |
| 8 | Gestionar trazas. | 1 |
| 9 | Gestionar archivos log. | 1 |
| 10 | Gestionar configuración. | 1 |

### 2.2.6.2 Plan de iteraciones

Luego de ser identificadas y descritas las Historias de Usuarios, además de estimar su esfuerzo, se procede a la planificación de la fase de implementación, donde se realizarán dos iteraciones, las cuales se describen a continuación:

**Iteración 1**

En la iteración 1 se realizará el desarrollo de las historias de usuarios de prioridad alta (1, 2, 3).

**Iteración 2**

En la iteración 2 se realizará el desarrollo de las historias de usuarios de prioridad media (4, 5, 6, 7, 8, 9,10).

#### 2.2.6.3 Plan de duración de las iteraciones

Este plan se encarga de mostrar las historias de usuarios en el orden en que se implementarán en cada una de las iteraciones, así como la duración estimada.

**Tabla 7: Plan de duración estimada de las iteraciones.**

| Iteración | Historia de usuario | Duración total |
|---|---|---|
| **1** | Aceptar comunicación de aplicaciones clientes.<br>Establecer comunicación con el servidor del servicio.<br>Enviar notificación. | 7 semanas |
| **2** | Iniciar, detener y reiniciar comunicación.<br>Gestionar servicios.<br>Gestionar usuarios.<br>Gestionar listas de control de acceso.<br>Gestionar trazas.<br>Gestionar archivos log.<br>Gestionar configuración. | 13 semanas |

#### 2.2.6.4 Plan de entregas

En este plan se detalla la fecha fin de cada iteración, mostrando una versión desarrollada del producto en ese momento hasta lograr el producto final en la fecha establecida. A continuación se muestra el plan de entrega para cada iteración:

**SEN**: Servicio de Envío de Notificaciones.

**GC**: Gestión y Configuración.

**Tabla 8: Plan de entregas.**

| Sistema | Final de la Iteración 1. 30/3/2013 | Final de la Iteración 2. 12/5/2013 |
|---|---|---|
| Sistema | SEN v0.1 | GC v0.1 |

## 2.3 Conclusiones

En este capítulo se describió la propuesta del sistema a desarrollar. Se realizó una descripción general de los procesos de negocio y de las historias de usuarios propuestas por el cliente. Se definieron las funcionalidades a implementar. Se especificó la lista de reserva del producto. Se definieron las iteraciones y la planificación del esfuerzo dedicado a la realización de cada una de las historias de usuario.

# Capítulo 3 Diseño e implementación del sistema.

## 3.1 Introducción

En este capítulo se estudiará y definirá la arquitectura del sistema. Se realizará un análisis de los patrones de diseño que utilizará el sistema para la correcta implementación del mismo. Se mostrarán las clases y sus correspondientes responsabilidades. Se presentarán las clases persistentes y el modelo físico de la base de datos. Luego se describirán las tareas de la ingeniería necesarias para el desarrollo del sistema. Por último y no menos importante se realizarán pruebas al software para comprobar que el producto funciona según lo diseñado y que las funcionalidades se han implementado de forma adecuada.

## 3.2 Desarrollo

### 3.2.1 Arquitectura del sistema

Una arquitectura software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. La arquitectura del sistema permite la comunicación entre las partes interesadas en el desarrollo del sistema. Además resalta las primeras decisiones que tendrán un efecto profundo en todo el proceso de desarrollo. Constituye un modelo de cómo está estructurado el sistema y la forma en que interactúan sus componentes. (21)

En el desarrollo del sistema se utilizará para cada subsistema una arquitectura en dependencia de sus características. El subsistema del servicio de envío de notificaciones presentará una arquitectura cliente-servidor y el subsistema de gestión y configuración la arquitectura n capas.

#### 3.2.1.1 Arquitectura cliente servidor

La arquitectura cliente servidor es un modelo para el desarrollo de sistemas informáticos donde dos sistemas independientes pueden interactuar a través de mensajes, acceso a recursos o servicios. Estos sistemas, no tienen que estar físicamente en el mismo lugar, pueden estar separados incluso a cientos de kilómetros. El cliente es quien inicia la comunicación con el servidor. El servidor contiene la información que debe ser compartida y es el encargado de dar respuesta a las solicitudes de los clientes.

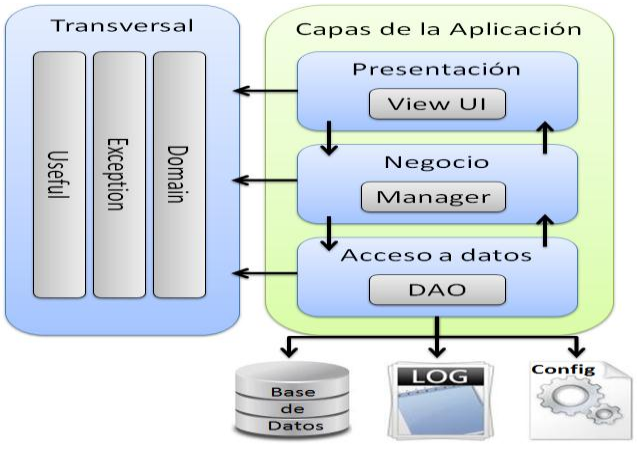
**Figura 1: Arquitectura cliente servidor.**


El servicio de envío de notificaciones utiliza la arquitectura cliente servidor para permitir la comunicación de aplicaciones clientes con el sistema. Posibilitando dar respuesta a las solicitudes de envío de notificaciones.

### 3.2.1.2 Arquitectura n capas

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada. (22) Esta arquitectura reduce las dependencias, de forma que, las capas inferiores desconocen el funcionamiento de las superiores. Esto posibilita la reutilización del código. El objetivo principal de la utilización de la arquitectura n capas en el subsistema de gestión y configuración es separar la presentación, el negocio y el acceso a datos.

**Figura 2: Arquitectura n capas**



**Capas:**

1. **Capa de presentación**

   Es la capa encargada de mostrar las interfaces de usuario. Los administradores pueden introducir datos para la gestión y configuración del sistema y este mostrará los resultados de las operaciones realizadas.

2. **Capa de negocio**

   Es la capa encargada de recibir las peticiones enviadas por la capa de presentación. Se comunica con la capa de acceso a datos para que envíe o guarde la información necesaria. Tiene como responsabilidad cargar y utilizar los plugins para la conexión con los servidores de los servicios de comunicación.

3. **Capa de acceso a datos**

   Es la capa encargada de consultar, manipular, controlar y almacenar los datos en la base de datos, en los archivos log y en los ficheros de configuración.

### 3.2.2 Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. (23)

#### 3.2.2.1 Patrones para Asignar Responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. (23)

##### 3.2.2.1.1 Experto

Problema: ¿De qué forma se puede saber qué responsabilidad delegar a cada objeto?

Solución: Asignar una responsabilidad al experto en información, la clase que tiene la información necesaria para llevar a cabo la responsabilidad. (24) Este patrón se verá reflejado por ejemplo en las clases User_Dao, Trace_Dao, Service_Dao y Acl_Dao, ya que serán las que manejen la información referente a usuarios, trazas, servicios y listas de control de acceso respectivamente.

#### 3.2.2.1.2 Creador

Problema: ¿Quién debería ser responsable de crear una nueva instancia?
Solución: Crear una nueva instancia por la clase que:

- Tiene la información necesaria para realizar la creación del objeto.
- Usa directamente las instancias creadas del objeto.
- Almacena o maneja varias instancias de la clase.
- Contiene o agrega la clase. (24)

Este patrón se verá reflejado por ejemplo en las clases User_Dao, Trace_Dao, Service_Dao y Acl_Dao, las cuales son las encargadas de insertar, actualizar, mostrar, eliminar y buscar la información de los usuarios, trazas, servicios y listas de control de acceso respectivamente.

#### 3.2.2.1.3 Controlador

Problema: ¿Quién gestiona un evento del sistema?
Solución: Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. (25) Este patrón se pone de manifiesto en la clase Server_Socket_Manager que será la encargada de gestionar los eventos del servicio de envío de notificaciones.

#### 3.2.2.1.4 Alta cohesión

Problema: ¿Cómo mantener manejable la complejidad?
Solución: Asignar responsabilidades de manera que la información que almacena una clase sea coherente y esté relacionada con la clase. (24)Este patrón se verá reflejado en las clases de acceso a datos, por ejemplo la clase User_Dao solo se encargará del acceso a los datos del usuario.

#### 3.2.2.1.5 Bajo acoplamiento

Problema: ¿Cómo dar soporte a las bajas dependencias y al incremento de la reutilización?
Solución: Diseñar con el objetivo de tener las clases lo menos ligadas entre sí que se pueda. De tal forma, que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. (23) Este patrón se verá reflejado en las clases de acceso a datos, por ejemplo la clase User_Dao solo se encargará del acceso a los datos del usuario.

### 3.2.3 Tarjetas Clase – Responsabilidad – Colaborador

Las tarjetas CRC son tarjetas de índices, una por cada clase, sobre las cuales se abrevian las responsabilidades de la clase y se anota una lista de los objetos con los que colaboran para desempeñarlas. Suelen elaborarse en una sesión de grupos pequeños donde los participantes representan papeles de las diversas clases. Cada grupo tiene las tarjetas CRC correspondientes a las clases cuyo papel está desempeñando. (26)

A continuación se muestran las tarjetas CRC de dos de las clases más importantes, las otras se encuentran en el Anexo 2.

**Tabla 9: Tarjeta CRC. Clase: Server_Socket_Manager.**

| **Clase:** Server_Socket_Manager | |
|---|---|
| **Descripción:** clase encargada de la comunicación con las aplicaciones clientes. | |
| **Responsabilidad** | **Colaborador** |
| Establecer comunicación con las aplicaciones clientes. | Intermediate_Manager, QTcpSocket, QHostAddress, Consumer, User, QTcpServer, LogFiles |
| Iniciar comunicación. | QHostAddress, QTcpServer, LogFiles |
| Detener comunicación. | QTcpServer, LogFiles, Consumer |
| Reiniciar comunicación. | QHostAddress, QTcpServer, LogFiles, Consumer |
| Recibir mensajes de las aplicaciones clientes. | Intermediate_Manager, QTcpSocket, Consumer, Encode_Manager, Useful |
| Escribir mensajes a las aplicaciones clientes. | QTcpSocket |
| Desconectar aplicaciones clientes. | QTcpSocket, Consumer, LogFiles |

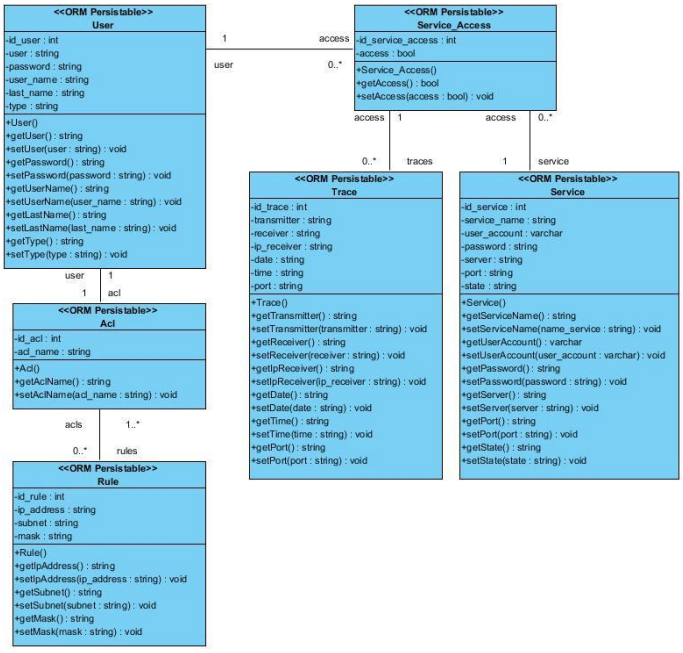
**Tabla 10: Tarjeta CRC. Clase: Intermediate_Manager.**

| **Clase:** Intermediate_Manager | |
|---|---|
| **Descripción:** clase encargada de procesar los mensajes recibidos por el servidor. | |
| **Responsabilidad** | **Colaborador** |
| Enviar mensaje. | Service_Interface, LogFiles, Service, Service_Dao, User_Dao, Consumer, QDateTime, QHostAddress, Trace, LogFiles |
| Procesar mensaje. | User, QHostAddress, Consumer, LogFiles |
| Autenticar aplicación cliente. | User, User_Dao, Consumer |

### 3.2.4 Diagrama de clases persistentes

El diagrama de clases persistentes muestra las clases que perduran en el tiempo, así como la relación que existe entre cada una de las clases.

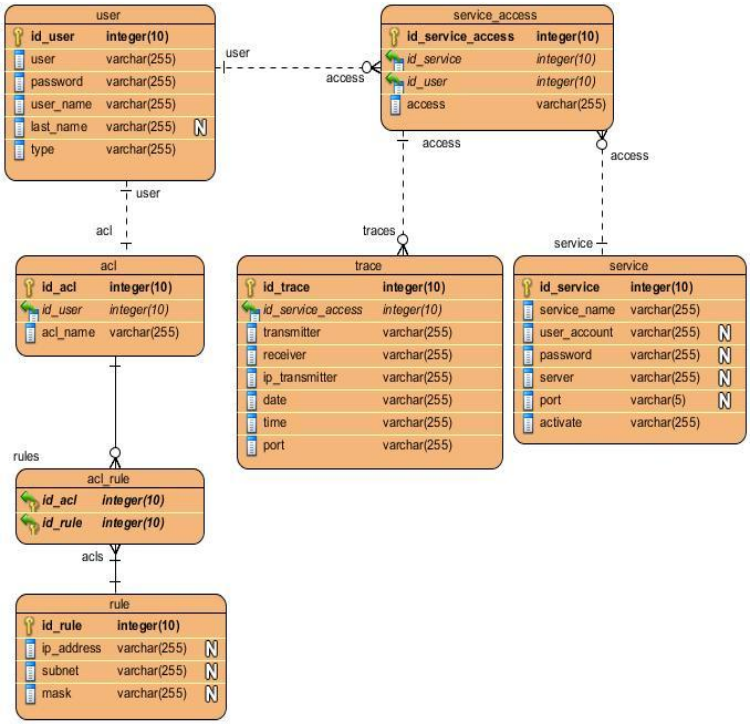
**Figura 3: Diagrama de clases persistentes**



### 3.2.5 Modelo físico de la base de datos

El modelo físico de la base de datos que a continuación se presenta, es obtenido del diagrama de clases persistentes. Este modelo muestra la relación que existe entre las entidades de las base de datos.

**Figura 4: Modelo de datos**




### 3.2.6 Tareas de Ingeniería

Las historias de usuario son descompuestas en tareas de la Ingeniería y asignadas a los programadores para ser implementadas durante una iteración.

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

- ✓ **Número tarea:** los números deben ser consecutivos.
- ✓ **Número historia de usuario:** número de la historia de usuario a la que pertenece la tarea.
- ✓ **Nombre tarea:** nombre que identifica a la tarea.
- ✓ **Tipo de tarea:** las tareas pueden ser de: Desarrollo, Corrección, Mejora, Otra (Especificar)
- ✓ **Puntos estimados:** tiempo estimado en semanas que se le asignará a su desarrollo.
- ✓ **Fecha inicio:** fecha en que inicia el desarrollo de la tarea.

- ✓ **Fecha fin**: fecha en que finaliza el desarrollo de la tarea.
- ✓ **Programador responsable:** nombre y apellidos del programador.
- ✓ **Descripción:** breve descripción de la tarea.

A continuación se presentan algunas de las tareas de Ingeniería más importantes y las otras se encuentran en el Anexo 2.

**Tabla 11: Tarea de Ingeniería #3: Escuchar y responder a la aplicación cliente.**

| **Tarea de Ingeniería** | |
|---|---|
| **Número tarea:** 3 | **Número historia de usuario:** 1 |
| **Nombre tarea:** Escuchar y responder a la aplicación cliente. | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 5/5 |
| **Fecha inicio:** 21/01/2013 | **Fecha fin:** 25/01/2013 |
| **Programador Responsable:** Dianet Díaz Oduardo | |
| **Descripción:** El sistema recibe los mensajes para la autenticación de aplicaciones clientes y sus solicitudes de envío de notificaciones. El sistema envía a las aplicaciones clientes el resultado de procesar sus mensajes. | |

**Tabla 12: Tarea de Ingeniería #8: Enviar notificación.**

| **Tarea de Ingeniería** | |
|---|---|
| **Número tarea:** 8 | **Número historia de usuario:** 3 |
| **Nombre tarea:** Enviar notificación | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**10/10 |
| **Fecha inicio:** 18/02/2013 | **Fecha fin:** 22/02/2013 |
| **Programador Responsable:** Alejandro García Núñez | |
| **Descripción:** La aplicación cliente le envía al sistema la notificación que desea enviar, si la aplicación está autenticada el sistema verifica que los datos de la notificación sean válidos y se los envía al servidor del servicio. Si la aplicación no está autenticada el sistema le notifica que no puede enviar notificaciones. Si la notificación pudo ser enviada correctamente al servidor, el sistema se lo comunica a la aplicación cliente. | |

### 3.3 Conclusiones

En este capítulo se estudió y se definió la arquitectura cliente servidor para el subsistema de envío de notificaciones y la arquitectura n capas para el subsistema de gestión y configuración. Se realizó un análisis de diferentes patrones y posteriormente se seleccionaron los patrones de diseño que se utilizarán para el desarrollo del sistema. Se describieron las tareas de Ingeniería para cada historia de usuario, necesarias para la implementación del sistema.

# Capítulo 4 Prueba.

## 4.1 Introducción

Se realizarán pruebas de aceptación y pruebas unitarias al sistema, para comprobar que el producto funciona según lo diseñado. También se efectuarán pruebas de carga y estrés para observar el comportamiento del sistema ante un número determinado de conexiones.

## 4.2 Desarrollo

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

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida, diseñada por el cliente final. (27)

### 4.2.1 Pruebas unitarias

Las pruebas unitarias son desarrolladas por los programadores para verificar que el sistema se comporta de la manera esperada. (27) Las pruebas se realizaron una vez culminada cada una de las funcionalidades.

### 4.2.2 Pruebas de aceptación

Las pruebas de aceptación son una parte integral del desarrollo incremental practicado por la metodología XP. Todas las Historias de Usuarios son soportadas por las pruebas de aceptación, las cuales son definidas por el cliente. (27)

Las pruebas de aceptación correspondiente a cada una de las funcionalidades del sistema serán representadas mediante tablas divididas por las siguientes secciones:

- ✓ **Clases Válidas**: se definirán cada uno de los pasos necesarios durante el desarrollo de la prueba, teniendo en cuenta las entradas válidas.

- ✓ **Clases Inválidas**: se definirán cada uno de los pasos necesarios durante el desarrollo de la prueba, teniendo en cuenta las entradas inválidas.
  **Resultado Esperado**: se hará una breve descripción de cómo debe de responder el sistema ante las entradas válidas o entradas inválidas.
- ✓ **Resultado de la Prueba**: se hará una breve descripción del resultado que se obtiene.
- ✓ **Observaciones**: algún señalamiento o advertencia que sea necesario hacerle a la sección que se está probando.

A continuación se presentan las pruebas de aceptación más importantes, las otras están en el Anexo 4:

**Tabla 13: Prueba #1: Aceptar comunicación de aplicaciones clientes.**

| Clases Válidas | Clases Inválidas | Resultado Esperado | Resultado de la Prueba | Observaciones |
|---|---|---|---|---|
| La aplicación cliente se conecta con el sistema para enviar una notificación. La dirección IP desde donde la aplicación cliente está tratando de conectarse debe ser de las permitidas en las listas de control de acceso. Luego envía su usuario y contraseña al sistema. La autenticación debe ser correcta y la aplicación cliente debe tener permitido enviar notificaciones desde esa dirección IP. | | El sistema verifica que la dirección IP desde donde la aplicación cliente está tratando de conectarse es de las permitidas en las listas de control de acceso. Luego envía su usuario y contraseña al sistema. Si la autenticación es correcta y la aplicación cliente tiene permitido enviar notificaciones desde esa dirección | El sistema acepta la comunicación con la aplicación cliente. | |

| | | | | |
|---|---|---|---|---|
| | | IP. El sistema acepta la comunicación. | | |
| | La aplicación cliente se conecta con el sistema para enviar una notificación y la dirección IP desde donde la aplicación cliente está tratando de conectarse no es de las permitidas en las listas de control de acceso. | El sistema le notifica a la aplicación cliente que la dirección IP no es permitida y rechaza la conexión. | El sistema no acepta la comunicación con la aplicación cliente. | |
| | La aplicación cliente se conecta con el sistema para enviar una notificación. La dirección IP desde donde la aplicación cliente está tratando de conectarse debe ser de las permitidas en las listas de control de acceso. Luego envía usuario y contraseña no válidos. | El sistema le notifica a la aplicación cliente que la autenticación es incorrecta. | El sistema no acepta la comunicación con la aplicación cliente. | |
| | La aplicación cliente se conecta con el sistema para enviar una notificación. La | El sistema le notifica a la aplicación cliente que la dirección IP no es | El sistema no acepta la comunicación con la | |

| | | | | |
|---|---|---|---|---|
| | dirección IP desde donde la aplicación cliente está tratando de conectarse debe ser de las permitidas en las listas de control de acceso. Luego envía su usuario y contraseña al sistema. La autenticación debe ser correcta pero la aplicación cliente no tiene permitido enviar notificaciones desde esa dirección IP. | permitida y rechaza la conexión. | aplicación cliente. | |

**Tabla 14: Prueba #2: Establecer comunicación con el servidor del servicio.**

| Clases Válidas | Clases Inválidas | Resultado Esperado | Resultado de la Prueba | Observaciones |
|---|---|---|---|---|
| El sistema se conecta con el servidor del servicio, utilizando la dirección y el puerto definidos por el servidor. Luego el sistema se autentica con el servidor con su usuario y contraseña. | | El sistema se comunica con el servidor del servicio satisfactoriamente. | El sistema se comunica con el servidor del servicio. | |
| | El sistema intenta conectarse con el | El sistema le notifica a la aplicación | El sistema no se comunica con el | |

| | | | | |
|---|---|---|---|---|
| | servidor del servicio, utilizando una dirección no definida por el servidor. | cliente que no pudo establecer la comunicación con el servidor del servicio satisfactoriamente. | servidor del servicio. | |
| | El sistema intenta conectarse con el servidor del servicio, utilizando un puerto no definido por el servidor. | El sistema le notifica a la aplicación cliente que no pudo establecer la comunicación con el servidor del servicio satisfactoriamente. | El sistema no se comunica con el servidor del servicio. | |
| | El sistema se conecta con el servidor del servicio, utilizando la dirección y el puerto definidos por el servidor. Luego el sistema intenta autenticarse con un usuario incorrecto. | El sistema le notifica a la aplicación cliente que no pudo establecer la comunicación con el servidor del servicio satisfactoriamente. | El sistema no se comunica con el servidor del servicio. | |
| | El sistema se conecta con el servidor del servicio, utilizando la dirección y el puerto definidos por el servidor. Luego el sistema intenta autenticarse con una contraseña | El sistema le notifica a la aplicación cliente que no pudo establecer la comunicación con el servidor del servicio satisfactoriamente. | El sistema no se comunica con el servidor del servicio. | |

| | | | | |
|---|---|---|---|---|
| | incorrecta. | | | |

**Tabla 15: Prueba #3: Enviar notificación.**

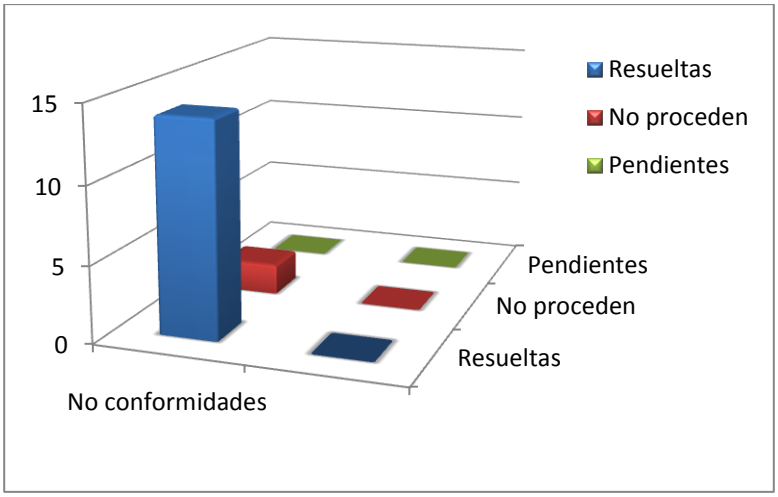
| Clases Válidas | Clases Inválidas | Resultado Esperado | Resultado de la Prueba | Observaciones |
|---|---|---|---|---|
| La aplicación cliente le envía al sistema la notificación que desea enviar, la aplicación debe estar autenticada, el servicio debe de existir, la aplicación cliente debe de tener permitido el uso del servicio que solicita, también la notificación tiene que tener destinatario. | | El sistema verifica que la aplicación esté autenticada, que la aplicación cliente tiene permitido el uso del servicio que solicita, que el servicio exista y que la notificación tenga un destinatario. Luego el sistema establece la comunicación con el servidor del servicio y envía la notificación. | El sistema envía la notificación. | |
| | La aplicación cliente le envía al sistema la notificación que desea enviar pero la aplicación no está autenticada. | El sistema le notifica a la aplicación cliente que no está autenticada. | El sistema no envía la notificación. | |
| | La aplicación cliente le envía al sistema la | El sistema le notifica a la aplicación | El sistema no envía la | |

| | | | | |
|---|---|---|---|---|
| | notificación que desea enviar, la aplicación está autenticada pero el servicio que solicita no existe. | cliente que el servicio solicitado no existe. | notificación. | |
| | La aplicación cliente le envía al sistema la notificación que desea enviar, la aplicación está autenticada, el servicio existe pero la aplicación cliente no tiene permitido el uso del servicio que solicita. | El sistema le notifica a la aplicación cliente que no tiene permitido el uso del servicio que solicita. | El sistema no envía la notificación. | |
| | La aplicación cliente le envía al sistema la notificación que desea enviar, la aplicación está autenticada, el servicio existe, la aplicación cliente tiene permitido el uso del servicio que solicita pero la notificación no tiene destinatario | El sistema le notifica a la aplicación cliente que la notificación no tiene destinatario. | El sistema no envía la notificación. | |

Luego de realizar las pruebas de aceptación para la primera iteración, se obtuvieron un total de 16 no conformidades, de las cuales 14 fueron resueltas, 2 no proceden y no quedó ninguna pendiente. En el

caso de la segunda iteración, no se detectaron no conformidades. Los resultados mencionados se puede observar en la siguiente gráfica.

**Figura 5: No conformidades**




### 4.2.3 Pruebas de carga y estrés

Debido a que el sistema informático permite la conexión de múltiples aplicaciones clientes de forma concurrente, es necesaria la realización de pruebas de carga y estrés para comprobar el comportamiento del software ante disimiles números de conexiones.

Para una mayor comprensión de los resultados de las pruebas, se tienen en cuenta las siguientes medidas:

Medidas estadísticas

- Mínimo:
  - Mínimo tiempo (mili segundos) de conexión entre todas las solicitudes realizadas.
- Máximo:
  - Máximo tiempo (mili segundos) de conexión entre todas las solicitudes realizadas.

Medidas de rendimiento

- Cantidad de conexiones por segundo:
  - Cantidad de conexiones por segundo que el sistema puede aceptar.
- Cantidad de conexiones por minuto:
  - Cantidad de conexiones por segundo que el sistema puede aceptar.
- Errores:
  - Porciento de errores respecto al número total de peticiones

#### 4.2.3.1 Pruebas de carga

Una prueba de carga se ejecuta para comprender el comportamiento de una aplicación ante una carga determinada. Esta carga puede ser el número de usuarios esperado en producción o un número de transacciones durante un tiempo determinado.

A continuación se establecen los elementos a tener en cuenta para la realización de la prueba:

- Se analizará el comportamiento del sistema cuando 100 aplicaciones clientes intentan conectarse concurrentemente al servidor.
- Recursos necesarios:

**Tabla 16: Recursos necesarios para realizar la prueba de carga.**

| Tipo de Prueba | Software | Hardware |
|---|---|---|
| **Carga** | JMeter[18] 2.3.1 | • **Procesador:** Intel Pentium 4<br>• **Motherboard:** Asus P5LD2<br>• **RAM:** 1 GB.<br>• **Disco duro:** 160 GB.<br>• **CPU:** 3.0 GHz.<br>• **Red:** 100,0 Mbps. |

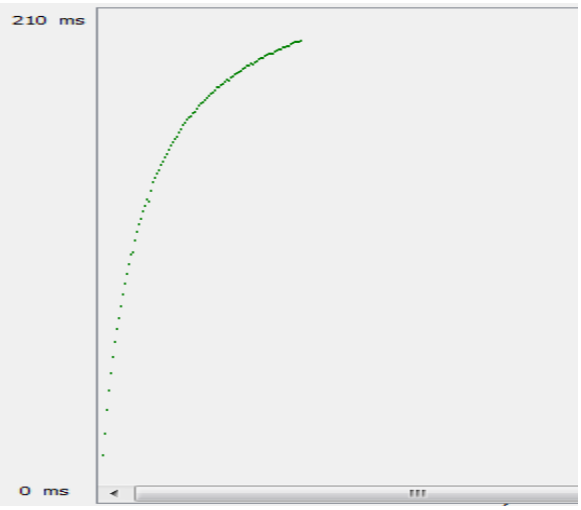
Resultados de la prueba:

[18] JMeter: Herramienta para la realización de pruebas de carga y estrés.

- Mínimo: 205
- Máximo: 208
- Cantidad de conexiones por segundo: 77
- Cantidad de conexiones por minuto: 4651
- Errores: 0%

Gráfico de resultados:

**Figura 6: Gráfico de resultados de la prueba de carga.**




### 4.2.3.2 Pruebas de estrés

Las pruebas de estrés son utilizadas para someter a la aplicación al límite de su funcionamiento mediante la ejecución de un número de usuarios muy superior al esperado. Tiene como finalidad determinar la robustez de una aplicación cuando la carga es extrema. Permite determinar el límite real de la aplicación en cuanto a número de usuarios concurrentes y el número de transacciones por segundo.

Prueba #1: Esta prueba consiste en analizar el sistema teniendo en cuenta la conexión de un número de aplicaciones clientes muy superior a la esperada.

A continuación se establecen los elementos a tener en cuenta para la realización de la prueba:

- Se analizará el comportamiento del sistema cuando 2300 aplicaciones clientes intentan conectarse concurrentemente al servidor.
- Recursos necesarios:

**Tabla 17: Recursos necesarios para realizar la prueba de estrés número 1.**

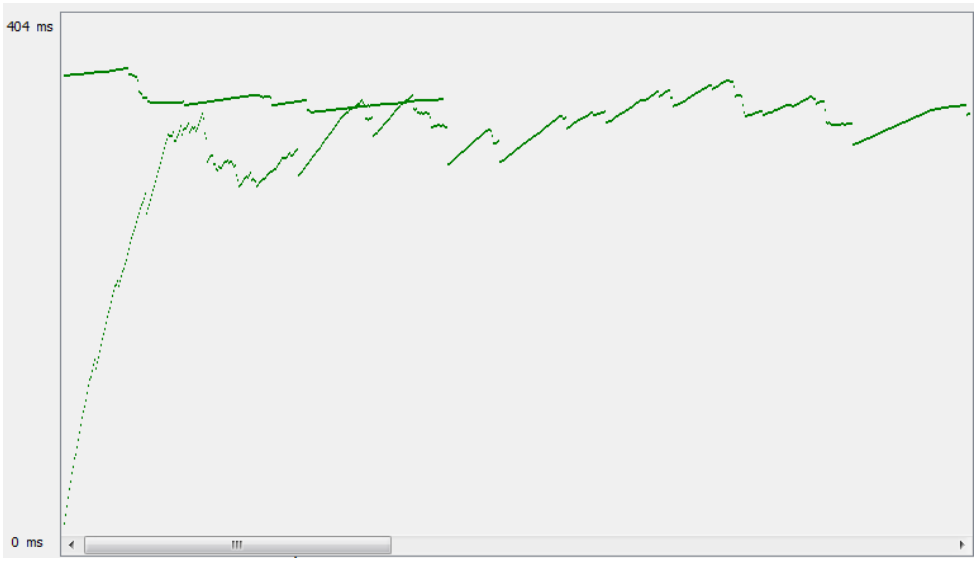
| Tipo de Prueba | Software | Hardware |
|---|---|---|
| **Estrés** | JMeter 2.3.1 | • **Procesador:** Intel Pentium 4<br>• **Motherboard:** Asus P5LD2<br>• **RAM:** 1 GB.<br>• **Disco duro:** 160 GB.<br>• **CPU:** 3.0 GHz.<br>• **Red:** 100,0 Mbps. |

Resultados de la prueba:

- Mínimo: 203
- Máximo: 1571
- Cantidad de conexiones por segundo: 222
- Cantidad de conexiones por minuto: 13147
- Errores: 0%

Gráficos de resultados:

**Figura 7: Gráfico de resultados de la prueba de estrés número 1.**




Prueba #2: Esta prueba consiste en analizar el sistema teniendo en cuenta la conexión de un número de aplicaciones clientes a las que el sistema no puede dar respuesta en su totalidad.

A continuación se establecen los elementos a tener en cuenta para la realización de la prueba:

- Se analizará el comportamiento del sistema cuando 2500 aplicaciones clientes intentan conectarse concurrentemente al servidor.
- Recursos necesarios:

**Tabla 18: Recursos necesarios para realizar la prueba de estrés número 2.**

| Tipo de Prueba | Software | Hardware |
|---|---|---|
| **Estrés** | JMeter 2.3.1 | • **Procesador:** Intel Pentium 4<br>• **Motherboard:** Asus P5LD2<br>• **RAM:** 1 GB.<br>• **Disco duro:** 160 GB. |

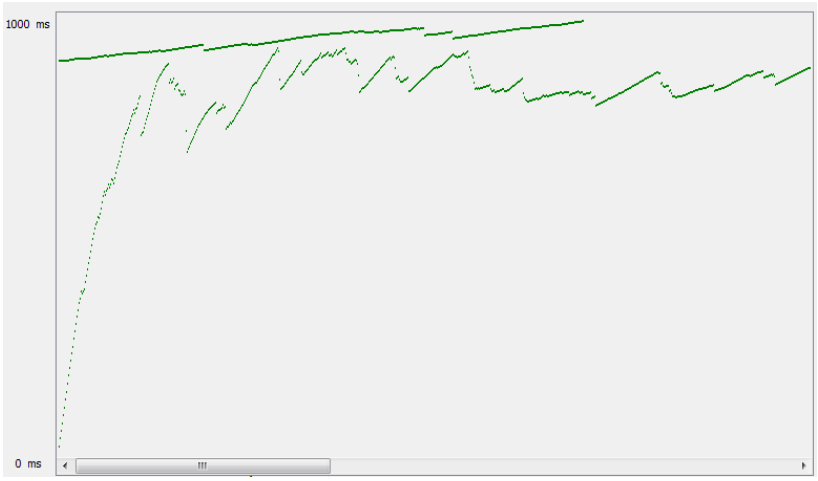
| | | |
|---|---|---|
| | | • **CPU:** 3.0 GHz.<br>• **Red:** 100,0 Mbps. |

Resultados de la prueba:

- Mínimo: 203
- Máximo: 3111
- Cantidad de conexiones por segundo: 223
- Cantidad de conexiones por minuto: 12873
- Errores: 0,2%

Gráfico de resultados:

**Figura 8: Gráfico de resultados de la prueba de estrés número 2.**




### 4.3 Conclusiones

Se realizaron pruebas unitarias y de aceptación. Las pruebas de aceptación proporcionaron como resultado un total de 16 no conformidades en la primera iteración y ninguna no conformidad para la segunda iteración. Se efectuaron pruebas de carga y estrés con la herramienta JMeter las cuales dieron como resultados que el sistema instalado en una computadora con 1 GB de RAM, 160 GB de disco duro, 3.0 GHz de CPU y una velocidad de red de 100,0 Mbps, **a**dmite 2500 aplicaciones conectadas concurrentemente.

## Conclusiones

Una vez finalizada la investigación se arribaron a las siguientes conclusiones:

- Se integró el proceso de comunicación asociado al envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP en la UCI, independientemente de las tecnologías que se utilicen en el desarrollo de las aplicaciones.
- Se diseñó e implementó la gestión y configuración del sistema, permitiendo a los administradores, tener un control de las aplicaciones clientes que hagan uso del envío de notificaciones.
- Se realizaron pruebas de aceptación y unitarias con el objetivo de comprobar la correcta implementación de cada una de las funcionalidades y así evidenciar la calidad del sistema desarrollado.
- Se realizaron pruebas de carga y estrés para analizar el comportamiento del sistema ante diferentes cantidades de aplicaciones conectadas concurrentes.

## Recomendaciones

Introducir al sistema otros servicios de comunicación para el envío de notificaciones.

Introducir al sistema la posibilidad de generar reportes bien detallados para analizar de manera más efectiva todos los aspectos relacionados con el envío de notificaciones.

## Referencias bibliográficas


1. **Mena, Alex Hans Hunter.** *Diseño e Implementación de Experiencias Docentes para el Servicio de Voz sobre IP, Mediante la utilización de la Plataforma Asterisk IPBX.* Valdivia : s.n., 2007. pág. 100.

2. **Claudia Rodríguez Crespo, Odaimys Rodríguez García.** Sitio de la Biblioteca de la Universidad de las Ciencias Informáticas! [En línea] [Citado el: 14 de 1 de 2013.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_04068_11/1/TD_04068_11.pdf.

3. Slideshare. [En línea] [Citado el: 29 de 10 de 2012.] http://www.slideshare.net/000kmi000/conceptos-bsicos-de-internet-y-sus-aplicaciones.

4. Universidad ICESI. [En línea] [Citado el: 1 de 15 de 2013.] http://www.icesi.edu.co/servicios_apoyo/acl.php.

5. Comunicación online para todos los públicos. [En línea] [Citado el: 29 de 10 de 2012.] http://comunicacionparatodos.wordpress.com/category/glosario/.

6. Scribd. [En línea] [Citado el: 2 de 3 de 2013.] http://es.scribd.com/doc/124201439/Informacion-de-Mantenimineto.

7. Pergamino Virtual. [En línea] [Citado el: 2 de 3 de 2013.] http://www.pergaminovirtual.com.ar/definicion/Log_Files.html.

8. Dell. [En línea] [Citado el: 14 de 1 de 2013.] http://www.dell.com/downloads/global/services/AlertFind-Spec-ES-XL-v2.pdf.

9. Entorno Virtual de Aprendizaje. [En línea] [Citado el: 25 de 11 de 2012.] http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/Evolucion_de_Metodologias.pdf.

10. **Autores, Colectivo de.** *Metodologías Ágiles en el Desarrollo de Software.* [ed.] Emilio A. Sánchez López Patricio Letelier Torres. Alicante, España : s.n., 12 de 11 de 2003.

11. Digia. [En línea] [Citado el: 2 de 3 de 2013.] http://qt.digia.com/product/.

12. Zona Qt. [En línea] [Citado el: 5 de mayo de 2013.] http://www.zonaqt.com/tutoriales/tutorial-b%C3%A1sico-de-qt-4.

13. Qt Project. [En línea] [Citado el: 11 de 1 de 2013.] http://qt-project.org/wiki/QtCreatorWhitepaper .

14. Free Download Manager. [En línea] [Citado el: 4 de 12 de 2012.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

15. software.com.ar. [En línea] [Citado el: 2 de 3 de 2013.] http://www.software.com.ar/visual-paradigm-para-uml.html.

16. SQLite. [En línea] [Citado el: 2 de 3 de 2013.] http://www.sqlite.org/about.html.

17. Zator Systems. [En línea] [Citado el: 11 de 1 de 2013.] http://www.zator.com/Cpp/E1_2.htm.

18. **Yoandri Quintana Rondón, Ivelin Ibarra Pérez, Jean Michael Suárez Pérez.** Biblioteca de la Universidad de las Ciencias Informáticas. [En línea] 6 de 2011. [Citado el: 2 de 3 de 2013.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_04329_11/1/TD_04329_11.pdf.

19. Biblioteca de la Universidad de las Ciencias Informáticas. [En línea] julio de 2008. [Citado el: 10 de mayo de 2013.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_1546_08/1/TD_1546_08.pdf.

20. **Canós, José H., Letelier, Patricio y Penadés, Mª Carmen.** *Métodologías Ágiles en el Desarrollo de Software.* [Documento] Valencia : Universidad Politécnica de Valencia.

21. Slideshare. [En línea] [Citado el: 1 de 4 de 2013.] http://www.slideshare.net/lilyPacheco7/arquitectura-de-software-13925226.

22. Universidad Politécnica Salesiana, Repositorio Digital. [En línea] [Citado el: 22 de 4 de 2013.] http://dspace.ups.edu.ec/bitstream/123456789/1663/13/UPS-GT000298.pdf.

23. MSDN. [En línea] [Citado el: 07 de 04 de 2013.] http://msdn.microsoft.com/es-es/library/bb972240.aspx.

24. Prácticas de software. [En línea] [Citado el: 05 de 04 de 2013.] http://www.practicasdesoftware.com.ar/2011/03/patrones-grasp/.

25. **Rainer Sugura Peña, Danae Pérez Arias.** Repositorio institucional. [En línea] [Citado el: 07 de 04 de 2013.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_04059_11/1/TD_04059_11.pdf.

26. Entorno Virtual de Aprendizaje. [En línea] [Citado el: 01 de 04 de 2013.] http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_3/UML_y_Patrones/04_Parte_IV_Fase_del_Diseno_1_.pdf.

27. **J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres.** *Pruebas del sistema en programación.* Sevilla : s.n.

28. Biblioteca Virtual en Salud. [En línea] [Citado el: 21 de 1 de 2013.] http://bvs.sld.cu/revistas/aci/vol13_5_05/aci03505.htm.

## Bibliografía


29. **Date, J. C.** *Introduction to database systems.* s.l. : Addison Wesley Longman, Inc., Reading Massachusetts., 2001. p. 960.

30. **Eva Gómez Ballester, Patricio Martínez Barco, Paloma Moreda Pozo, Armando Suárez Cueto, Andrés Montoyo Guijarro, Estela Saquete Boro.** *Apuntes Bases de Datos 1.* Alicante : s.n. p. 172.

31. **Landívar, Edgar.** *Comunicaciones Unificadas en Elastix.* [Documento] 2009.

32. **Villar, Malay Rodríguez.** Introducción de procedimientos Ágiles en la producción de software en la Facultad 7 de la Universidad de las Ciencias Informáticas. [Online] Junio 2007. [Cited: Noviembre 3, 2010.] http://bibliodoc.uci.cu/TD/TD_0693_07.pdf.

33. **Grau Abalo, Ricardo, Correa Valdés, Cecilia and Rojas Betancur, Mauricio.** *METODOLOGÍA DE LA INVESTIGACIÓN.* [Documento] Ibagué : s.n.

34. **Fernández Escribano, Gerardo.** *Introducción a Extreme Programming.* [Documento]

35. **Sommerville, Ian.** *Ingeniería de Software. Séptima Edición.* [Documento] Madrid : Pearson educación, 2005.

36. **Pressman, Roger S.** *Ingeniería de software. Un enfoque práctico. Capítulo 8 Modelado de Análisis.* [Documento]

37. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* [Documento] México : Pearson Education, 1999.

38. **Van Meggelen, Jim, Madsen, Leif and Smith, Jared.** *Asterisk. The Future of de Telephony.* [Documento] 2007.

39. **Prieto, Félix.** *Patrones de Diseño.* [Documento] s.l. : Universidad de Valladolid, 2008/09.

40. **Microsoft.** TechNet. [Online] [Cited: febrero 25, 2013.] http://technet.microsoft.com/es-es/library/aa996704%28v=exchg.65%29.aspx.

Anexos

## Anexo 1

**Tabla 19: HU #3: Enviar notificación.**

| **Historia de usuario** | |
|---|---|
| **Número**: 3 | **Nombre de historia de usuario:** Enviar notificación. |
| **Modificación de historia de usuario número:** Ninguna | |
| **Usuario:** Dianet Díaz Oduardo, Alejandro García Núñez | **Iteración asignada:** 1 |
| **Prioridad en negocio**: Alta | **Puntos estimados**: 2 |
| **Riesgo en desarrollo:** Alto | **Puntos reales:** 2 |
| **Descripción**: El sistema envía al servidor del servicio solicitado, la notificación enviada desde la aplicación cliente. | |
| **Prototipo de interfaz:** No aplica | |

**Tabla 20: HU #4: Iniciar, detener y reiniciar comunicación.**

| **Historia de usuario** | |
|---|---|
| **Número**: 4 | **Nombre de historia de usuario:** Iniciar, detener y reiniciar comunicación. |
| **Modificación de historia de usuario número:** Ninguna | |
| **Usuario:** Dianet Díaz Oduardo, Alejandro García Núñez | **Iteración asignada:** 2 |
| **Prioridad en negocio**: Media | **Puntos estimados**: 1 |
| **Riesgo en desarrollo:** Medio | **Puntos reales:** 1 |
| **Descripción**: El administrador puede iniciar, detener o reiniciar el servidor del sistema. | |

**Prototipo de interfaz:**


## Anexo 2

**Tabla 21: Tarjeta CRC. Clase: Acl_Dao.**

| **Clase:** Acl_Dao | |
|---|---|
| **Descripción:** clase encargada de la gestión de las listas de control de acceso. | |
| **Responsabilidad** | **Colaborador** |
| Insertar ACL. | Acl, QSqlQuery, User |
| Mostrar ACL | Acl, QSqlQuery, User |

**Tabla 22: Tarjeta CRC. Clase: Trace_Dao.**

| **Clase:** Trace_Dao | |
|---|---|
| **Descripción:** clase encargada de la gestión de trazas. | |
| **Responsabilidad** | **Colaborador** |
| Insertar traza. | Trace, QSqlQuery, User, Service |
| Mostrar trazas | Trace, QSqlQuery, Service |
| Buscar trazas | Trace, QSqlQuery |

**Tabla 23: Tarjeta CRC. Clase: Rule_Dao.**

| **Clase:** Rule_Dao | |
|---|---|
| **Descripción:** clase encargada de la gestión de reglas. | |
| **Responsabilidad** | **Colaborador** |

| | |
|---|---|
| Insertar regla. | Rule, QSqlQuery, User |
| Eliminar regla. | Rule, QSqlQuery, User |
| Buscar reglas. | Rule, QSqlQuery, User |
| Modificar regla | Rule, QSqlQuery, User |

## Anexo 3

**Tabla 24: Tarea de Ingeniería #1: Verificar dirección IP de la aplicación cliente.**

| **Tarea de Ingeniería** | |
|---|---|
| **Número tarea:** 1 | **Número historia de usuario:** 1 |
| **Nombre tarea:** Verificar dirección IP de la aplicación cliente. | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 5/5 |
| **Fecha inicio:** 7/01/2013 | **Fecha fin:** 11/01/2013 |
| **Programador Responsable:** Dianet Díaz Oduardo | |
| **Descripción:** La aplicación cliente se conecta con el sistema para enviar una notificación. El sistema verifica que la dirección IP desde donde la aplicación cliente está tratando de conectarse es permitida, si no lo es, el sistema le notifica al cliente que no puede conectarse desde ese IP y rechaza la conexión. Antes de autenticar una aplicación cliente, el sistema verifica si esta tiene permitido enviar notificaciones desde esa dirección IP. Si no tiene permiso, el sistema le informa a la aplicación cliente que no puede enviar notificaciones y rechaza la conexión. | |

**Tabla 25: Tarea de Ingeniería #2: Autenticar aplicación cliente.**

| **Tarea de Ingeniería** | |
|---|---|
| **Número tarea:** 2 | **Número historia de usuario:** 1 |
| **Nombre tarea:** Autenticar aplicación cliente. | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 5/5 |
| **Fecha inicio:** 14/01/2013 | **Fecha fin:** 18/01/2013 |

| |
|---|
| **Programador Responsable:** Alejandro García Núñez |
| **Descripción:** La aplicación cliente le envía al sistema un mensaje con su usuario y contraseña, si son válidos el sistema autentica la aplicación y le envía un mensaje de confirmación. En caso contrario el sistema le notifica a la aplicación cliente que los datos no son válidos. |

**Anexo 4**

**Tabla 26: Prueba #4: Iniciar comunicación.**

| Clases Válidas | Clases Inválidas | Resultado Esperado | Resultado de la Prueba | Observaciones |
|---|---|---|---|---|
| El administrador presiona el click derecho sobre el icono del sistema y escoge la opción ¨Iniciar¨. | | El sistema está en escucha esperando las solicitudes de conexión o de envío de notificaciones de las aplicaciones clientes. | El sistema está en escucha esperando las solicitudes de conexión o de envío de notificaciones de las aplicaciones clientes. | |