

**FACULTAD 2 “TELECOMUNICACIONES”**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**



**Título:** Sistema de Gestión de Alertas para la tecnología RFID.

**Autores:**

Maylin Oña Bernabeu

Norey Fernández De la Torre

**Tutor:**

Salvador González Gómez.

**Ciudad de La Habana, junio del 2013**

**“Año 55 de la Revolución”**

## Declaración de autoría

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Maylin Oña Bernabeu

\_\_\_\_\_

*Firma del autor*

Norey Fernández De la Torre

\_\_\_\_\_

*Firma del autor*

Msc. Salvador González Gómez

\_\_\_\_\_

*Firma del tutor*

Ing.Lex Karel Zayas Hernandez

\_\_\_\_\_

*Firma del co - tutor*

## **Pensamiento**

*...“Las producciones intelectuales serán el sustento fundamental de Cuba. La idea es convertir la informática en una de las ramas más productivas y aportadoras de recurso para la nación”...*

*Fidel Castro.*



# Agradecimientos

*Agradecemos a nuestro Comandante en Jefe Fidel y a la Revolución Cubana, por poner en nuestras manos los medios para convertirnos en hombres de ciencia. A la Universidad de las Ciencias Informáticas por forjarnos como profesionales y personas de bien.*

*Al Ministerio del Interior por habernos brindado la posibilidad de estudiar en esta universidad.*

*Agradecemos a nuestro tutor Salvador González Gómez por su ayuda incondicional. A nuestro tribunal porque sin sus críticas y sugerencias no hubiésemos llegado hasta aquí. A todos los profesores que han contribuido en nuestra formación durante estos 5 años, a nuestros amigos...*

## **De Maylin**

*En este momento quiero agradecer de manera especial a mi mamá y a mi papá por darme la fuerza y el impulso, por su confianza en todo momento, por acompañarme y por creer en mí desde que inicié mis estudios, por ser un ejemplo a seguir.*

*Quiero también agradecerles por su amor y apoyo incondicional en los momentos difíciles y por guiarme siempre por el camino correcto.*

*A mi hermanita Roxi que la quiero mucho por ser parte de mi alegría.*

*A toda mi familia que de una manera u otra me han apoyado a lo largo de estos cinco años.*

*A mi compañero de tesis por su paciencia y dedicación, por estos cinco años de compartir buenos y malos momentos, por ayudarme a convertirme en Ingeniera.*

*A Adrian por su apoyo y comprensión y brindarme su ayuda en todo lo que ha podido.*

*A las amistades que se hacen a lo largo de la carrera, a todos aquellos con los que compartí buenos y malos momentos gracias por su compañía.*

## **De Norey**

*A mis padres quienes desde pequeño supieron guiarme hacia el camino correcto y que sin su ayuda y dedicación no hubiera sido posible convertirme en lo que hoy soy. A ustedes muchas gracias.*

*A mi hermana por darme su apoyo durante el transcurso de mis estudios y por estar siempre a mi lado en cada etapa de mi vida.*

*A toda mi familia por su constante preocupación y confianza en mí.*

*A mi prima Erker porque a pesar de estar lejos siempre me apoyó y estuvo pendiente de mí.*

*A Edistio y a Genry por sus consejos y ayuda infinita.*

*A los profesores Yoanny, Charchaval, Deivis y Andry por la dedicación brindada en la realización de esta tesis, por haberme transmitido sus conocimientos y ayudarme a desarrollar este trabajo.*

*A mi compañera de tesis por haber significado tanto, por ser amiga, y una gran persona. Muchas gracias por estos años de amistad y por soportarme a tu lado con tanto cariño.*

*A Reinier a Luis y a Olivio, por apoyarme y dedicarme parte de su tiempo y ayudarme cada vez que necesitaba. Estaré eternamente agradecido por su apoyo y amistad.*

*A mis compañeros de cuarto, Olivio, Reinier, Dagoberto y Arián por confiar en mí, y darme la seguridad de confiar en ustedes. Por llenar cada uno de los recuerdos que me llevo, por sus consejos, sus charlas y por tantos ratos inolvidables.*

*A todos mis compañeros de estudio por todo lo que hemos compartido juntos en este tiempo y que nunca los voy a olvidar.*

*A todos los profesores que he tenido, que han contribuido de alguna forma a mi formación, a ellos les agradezco infinitamente por el perdurable caudal de conocimientos que me dieron y mi formación como profesional a la que contribuyeron en gran medida.*

*A todo el que me conoce y que de una forma u otra ha contribuido a hacer de este sueño una realidad.*

# **Dedicatoria**

## **De Maylin**

*Les dedico este trabajo a mis padres como regalo de su constante dedicación y esfuerzo, por iluminar mi camino y darme la pauta para poder realizarme en mis estudios y en mi vida. Agradezco los consejos sabios que en el momento exacto han sabido darme para convertirme en una mejor persona y enfrentar los momentos difíciles, por ayudarme a tomar las decisiones que me ayuden a balancear mi vida y por sobre todo por el amor tan grande que me dan.*

*A mi hermanita por ser mi razón de ser.*

## **De Norey**

*Dedico este trabajo a dos personas que son lo más grande para mí en esta vida, mi mamá y mi papá. A ustedes por creer en mí y porque me sacaron adelante, dándome ejemplo digno de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzar mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera y por el orgullo que sienten por mí fue lo que me hizo ir hasta el final. Va por ustedes por lo que valen, por lo que admiro, por ser mi razón de ser y por ser los mejores padres del mundo.*

*Dedico también mis resultados a una persona que quiero con la vida, mi hermana. Gracias por estar siempre a mi lado, por tu comprensión, tu dedicación, gracias por el apoyo, gracias por seguirme a donde voy, por estar ahí cuando te necesito, por quererme y preocuparte por mí.*

*Sé cuánto les orgullece verme graduado y cuánto desearon este momento, por eso este triunfo es de los cuatro, porque no somos cuatro, somos uno.*

## **Acrónimos.**

<b>AJAX</b>	<i>Asynchronous JavaScript And XML</i>
<b>CRC</b>	<i>Class-Responsibility-Collaboration</i>
<b>CMMI</b>	<i>Capability Maturity Model Integration</i>
<b>CASE</b>	<i>Computer Aided Software Engineering</i>
<b>CSS</b>	<i>Cascading Style Sheets</i>
<b>DAO</b>	<i>Data Access Object</i>
<b>DOM</b>	<i>Document Object Model</i>
<b>EPC</b>	<i>Electronic Product Code</i>
<b>GRASP</b>	<i>General Responsibility Assignment Software Patterns</i>
<b>HQL</b>	<i>Hibernate Query Language</i>
<b>HTML</b>	<i>HyperText Markup Language</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>IDEs</b>	<i>Integrated Development Environment</i>
<b>JDBC</b>	<i>Java Database Connectivity</i>
<b>JSP</b>	<i>JavaServer Pages</i>
<b>LLRP</b>	<i>Low Level Reader Protocol</i>
<b>MVC</b>	<i>Model View Controller</i>
<b>RFID</b>	<i>Radio Frequency Identification</i>

**Acrónimos.**

**Sistema de Gestión de Alertas para la tecnología RFID**

---

<b>RUP</b>	<i>Rational Unified Process</i>
<b>SQL</b>	<i>Structured Query Language</i>
<b>SMS</b>	<i>Short Message Service</i>
<b>XML</b>	<i>Extensible Markup Language</i>
<b>XP</b>	<i>Extreme Programming</i>

## Resumen

Desde la antigüedad el hombre se percató de la importancia de la información y desde entonces se empleó en dejar un legado de todo conocimiento posible, transitando por varias formas de almacenarla, compartirla y gestionarla en sentido general, empezando por tallados en piedra, pinturas en paredes de cuevas, luego por papiros, no sólo puso en práctica la invención tecnológica de los materiales, también desarrolló otras muchas facetas inventivas como la escritura, la invención de la imprenta, la construcción de nuevas máquinas y actualmente, usando las poderosas herramientas de la era digital. En la actualidad la información se ha convertido en un enorme tesoro, por su importancia y cantidad. La gestión de esta se considera formada en dos sub-disciplinas la microinformática<sup>1</sup> y la macroinformática<sup>2</sup>. Llegando a alcanzar su gestión en general, un importante lugar en el mundo digital, lo que ha traído consigo que a medidas que crece este volumen de datos, continuar con los métodos tradicionales de tratamiento se ha vuelto algo realmente imposible.

Las Tecnologías de la Información y las Comunicaciones (TIC) representan una potente herramienta para la gestión de la información en la actualidad. El proceso de toma de decisiones y análisis de grandes empresas se sustentan en la información almacenada durante años. A lo largo del proceso revolucionario cubano se han llevado a cabo un grupo de acciones para el desarrollo del país. Una de las esferas que se ha venido perfeccionando en estos últimos años es la informática y las comunicaciones, presente en todos los ámbitos de la sociedad. El país no cuenta actualmente con un sistema que brinde la funcionalidad de gestionar la información relacionada con la gestión de alertas usando la tecnología RFID (Identificación por Radio Frecuencia). La esencia de este trabajo es la concepción de un sistema que pretende ser una solución viable a la problemática de gestión de esta información para su posterior utilización. Siendo capaz de mantener disponible y actualizada la información relacionada con las alertas, permitiendo hacer búsquedas filtradas y realizar reportes sobre la información que se gestiona.

Para la elaboración de este sistema se ha utilizado como tecnología Spring MVC y la metodología de desarrollo de software Extreme Programming (XP).

Palabras claves: gestión, informatización, tecnología, sistema, RFID, alertas.

---

<sup>1</sup> Estudia el individuo y su relación con la información, así como los sistemas diseñados, para facilitar ese uso por el individuo. Abarca el estudio de las interfaces persona-información y los sistemas de información

<sup>2</sup> Estudia la información en el contexto de las organizaciones y la sociedad. Abarca el estudio de los contextos y entornos de información.

# Índice de Contenidos

<b>Acrónimos</b> .....	7
<b>Resumen</b> .....	9
<b>Introducción</b> .....	15
<b>Capítulo 1. Fundamentación Teórica</b> .....	19
1.1    Conceptos Fundamentales: .....	19
1.1.1    Tecnología de Identificación por Radiofrecuencia (RFID). .....	19
1.1.2    Descripción de componentes.....	19
1.1.3    Principio y funcionamiento de un sistema RFID. ....	20
1.1.4    Ventajas de un sistema RFID. ....	21
1.1.5    Desventajas: .....	21
1.1.6    Aplicaciones de la tecnología RFID. ....	22
1.2    Software relacionado con la gestión de alertas usando la tecnología RFID. ....	23
1.2.1    Sistema de localización de personal hospitalario mediante RFID activa. ....	23
1.2.2    Sistema RFID para detectar fugas en estaciones de gas. ....	23
1.2.3    Tecnologías, herramientas y metodologías.....	24
1.2.4    Lenguaje de programación.....	24
1.2.5    Sistema Gestor de Base de Datos (SGBD).....	25
1.2.6    Tecnología del lado del servidor. ....	26
1.2.7    Tecnología del lado del cliente. ....	28
1.2.8    Entorno de Desarrollo Integrado (Integrated Development Environment, IDE). ....	29
1.2.9    Metodología de desarrollo de software. ....	29
1.2.10    Herramienta case. ....	30
1.3    Conclusiones. ....	30
<b>Capítulo 2. Características y Propuesta del Sistema</b> .....	32
2.1    Introducción. ....	32
2.2    Propuesta del Sistema.....	32
2.2.1    Proceso de Gestión de Alertas.....	32

2.3	Lista de Reserva del producto. ....	33
2.3.1	Funcionalidades del Sistema. ....	33
2.3.2	Requisitos no funcionales del sistema. ....	34
2.4	Planeación. ....	36
2.4.1	Historias de Usuario. ....	36
2.4.2	Campos de las historias de usuarios. ....	37
2.4.3	Campos de las historias de usuarios. ....	37
2.5	Plan de iteraciones. ....	42
2.6	Conclusiones. ....	43
<b>Capítulo 3: Diseño e Implementación del Sistema. ....</b>		<b>45</b>
3.1	Introducción. ....	45
3.2	Patrón arquitectónico: Modelo Vista Controlador (MVC). ....	45
3.3	Arquitectura del Sistema. ....	47
3.3.1	Arquitectura Cliente/Servidor. ....	47
3.3.1.1	Componentes de la arquitectura Cliente/Servidor. ....	47
3.3.1.2	Elementos principales. ....	48
3.4	Diseño de la Base de Datos. ....	49
3.5	Tareas de Ingeniería. ....	49
3.5.1	Campos de las tareas de ingeniería. ....	50
3.5.2	Tareas de Ingeniería a desarrollar. ....	50
3.6	Principios de Diseño. ....	53
3.6.1	Estándares de la interfaz de la aplicación. ....	53
3.7	Diagrama de Despliegue. ....	53
3.8	Tarjetas Clase-Responsabilidad-Control (CRC). ....	54
3.8.1	Tarjetas CRC. ....	54
3.9	Patrones. ....	54
3.9.1	Patrones de Diseño. ....	54
3.10	Conclusiones. ....	55
<b>Capítulo 4. Pruebas. ....</b>		<b>57</b>

4.1	Introducción.....	57
4.2	Pruebas de Software.....	57
4.3	Verificación y Validación.....	57
4.4	Tipos de pruebas.....	57
4.5	Tipo de prueba seleccionada.....	58
4.6	Casos de Pruebas.....	58
4.7	No Conformidades.....	60
4.8	Conclusiones.....	61
	<b>Conclusiones Generales.....</b>	<b>62</b>
	<b>Recomendaciones.....</b>	<b>63</b>
	Anexo 1: Historias de usuario.....	66
	Anexo 2: Tareas de Ingeniería.....	70
	Anexo 3: Tarjetas Clase-Responsabilidad-Control.....	70
	Anexo 4: Casos de Pruebas.....	71

## Índice de Tablas

Tabla 2.1: Historia de Usuario Generar Alarma.....	38
Tabla 2.2: Historia de Usuario Actualizar estado de la alarma.....	39
Tabla 2.3: Historia de Usuario Enviar mensaje de texto y correo electrónico. ....	40
Tabla 2.4: Historia de Usuario Mostrar reportes.....	41
Tabla 2.5: Historia de Usuario Búsquedas filtradas.....	41
Tabla 2.6: Plan de Iteraciones. ....	43
Tabla 3.1: Tarea de Ingeniería Gestionar alarma. ....	50
Tabla 3.2: Tarea de Ingeniería Actualizar estado de las alarmas. ....	51
Tabla 3.3: Tarea de Ingeniería Enviar mensaje de texto y correo electrónico. ....	51
Tabla 3.4: Tarea de Ingeniería Mostrar reportes. ....	52
Tabla 3.5: Tarea de Ingeniería Búsquedas filtradas.....	52
Tabla 3.6: Tarjeta CRC Clase IUsuarioDao. ....	54
Tabla 4.1: Caso de Prueba Gestionar Usuario.....	60
Tabla 4.2: Resumen de No Conformidades. ....	60
Tabla 2.7: Historia de Usuario Crear Usuario. ....	67
Tabla 2.8: Historia de Usuario Modificar Usuario. ....	67
Tabla 2.9: Historia de Usuario Desactivar Usuario. ....	68
Tabla 2.10: Historia de Usuario Autenticar Usuario.....	69
Tabla 3.7: Tarea de Ingeniería Crear usuario. ....	70
Tabla 3.24: Tarjeta CRC Clase ICentroDao.....	70
Tabla 4.3: Caso de Prueba Gestionar Animal. ....	72

## Índice de Figuras

Figura 1.1: Componentes de un sistema RFID.....	20
Figura 3.1: Patrón Arquitectónico: Modelo-Vista-Controlador.....	45
Figura 3.2: Diseño Arquitectónico. ....	46
Figura 3.3: Modelo Entidad Relación. ....	49
Figura 3.4: Diagrama de Despliegue.....	53

## Introducción

En los últimos años, el mundo ha sido testigo del gran auge que ha tenido el desarrollo masivo de las ciencias y las tecnologías. Las invenciones tecnológicas de una u otra manera han servido para mejorar la calidad de vida de muchas personas. Estos avances demuestran que en la sociedad compartir lo poco que se tiene es la mejor forma de obtener beneficio, conocimiento y desarrollo en forma abundante. Diferentes disciplinas científicas tuvieron un impacto directo en ello, pero sin dudas una de las que más influencia ha tenido y tiene, es la Informática.

En Cuba como en otros países, también se ha puesto en práctica la utilización de la informática para el desarrollo de la sociedad, apoyándose en las Tecnologías de la Información y las Comunicaciones (TIC), las cuales se han convertido, desde su llegada a los escenarios nacionales, en un elemento indispensable para establecer las líneas de desarrollo de la sociedad cubana.

La Universidad de las Ciencias Informáticas (UCI) como centro de estudios y encargada del desarrollo de software en el país, está organizada en varias facultades donde cada una desempeña un papel fundamental en los diferentes campos de esta rama de la ciencia. Entre ellas se encuentra la Facultad 2 donde radica el Centro de Telemática, la cual tiene la tarea de desarrollar un Sistema de Gestión de Alertas para la Tecnología RFID, la misma está pensada para brindar servicios tales como la gestión de las alertas en determinadas instituciones. En Cuba el proceso de gestión de alertas se realiza de forma manual, los encargados de llevar este proceso reciben un informe con la descripción de cada una de las alertas a las cuales se le realizarán tratamiento. Según la magnitud del problema las alertas son clasificadas por niveles donde van a ser atendidas por una persona determinada la cual es la encargada de darle seguimiento en un período de tiempo. Una vez atendidas serán archivadas como constancia de su tratamiento formando parte de un historial de archivos. Este proceso resulta muy engorroso y trae consigo que existan problemas de seguridad, organización, así como pérdida de archivos, mala redacción de la información, falta de comunicación entre el personal, desaprovechamiento de la jornada laboral por parte de los trabajadores y consumo de recursos como papel, lapiceros y escritorios.

La tecnología RFID poco a poco ha ido ganando terreno entre los sistemas de identificación automáticos. Pensado en un origen como sustituto del código de barras, ahora mismo se aplica en numerosos campos y sectores de la industria. Ejemplos de estos son: el control de la calidad, producción y distribución, localización y seguimiento de objetos, detección de falsificaciones, etc. Además hay multitud de

investigaciones orientadas al uso de esta tecnología en un futuro no muy lejano. Por estos motivos ha surgido un interés en los sistemas basados en RFID y se ha seleccionado esta tecnología para su estudio en profundidad en este proyecto.

Dada la situación antes expuesta surge el siguiente *problema a resolver*:

¿Cómo contribuir a la gestión de alertas en sistemas de RFID?

Según el problema identificado anteriormente el *objeto de estudio* se enmarca en un sistema de gestión de alertas teniendo *como campo de acción* Proceso de gestión de alertas mediante la tecnología RFID.

El *objetivo general* de la investigación es desarrollar un Sistema de Gestión de Alertas que permita visualizar los registros de actividades desarrollados por un sistema RFID.

Con el propósito de cumplir con el objetivo planteado se definen las siguientes *tareas de investigación*:

- ✓ Realizar un estudio de la tecnología RFID identificando sus principales características y sus aplicaciones para la realización del sistema.
- ✓ Determinar las tendencias actuales en la utilización de la tecnología RFID para el desarrollo de la aplicación.
- ✓ Definir el funcionamiento de la tecnología RFID, mediante un estudio de las plataformas y herramientas de desarrollo.
- ✓ Determinar el formato de la alerta mediante el estándar.
- ✓ Implementar funcionalidades de la aplicación.
- ✓ Realizar pruebas al sistema.

Con el cumplimiento de las tareas de investigación se obtendrá una aplicación para la gestión de alertas mediante la tecnología RFID, la misma permitirá a las empresas contar con una herramienta que gestione las alertas generadas.

En el transcurso de la investigación se utilizarán los siguientes Métodos de Investigación Científica:

Métodos teóricos:

- ✓ *Histórico – Lógico:* En la primera parte de la investigación se desarrollará un estudio del estado del arte de la problemática; así como se analizarán las ventajas y desventajas de cada una de las herramientas utilizadas para poder desarrollar el sistema de gestión de alertas.
- ✓ *Revisión documental:* Consiste en la selección y recopilación de información por medio de la lectura y crítica de documentos y materiales bibliográficos. Utilizada para recoger los datos más valiosos que aporta la bibliografía actual sobre el tema a tratar.
- ✓ *Analítico–Sintético:* Se utiliza durante el proceso de revisión bibliográfica para conocer las principales técnicas, tecnologías, metodologías de desarrollo de software con el fin de lograr la confesión de la aplicación. Se puede comprobar la utilización de este método en el Capítulo 1.

Métodos empíricos:

- ✓ *Entrevista:* Se utilizará la entrevista como una conversación planificada con el cliente, para obtener información acerca del problema en cuestión. Su uso constituye un medio para el conocimiento cualitativo de las características particulares de un proceso y puede influir en el posterior análisis y diseño del producto de software que se informatizará.
- ✓ *Observación:* Este método permitió conocer la realidad mediante percepción del objeto de estudio, es decir, cuando se comenzó la investigación la observación permitió saber cómo son los problemas existentes, teniendo como parte además de la misma las entrevistas.

El presente documento consta de 4 capítulos, estructurados de la siguiente forma:

Capítulo 1: Fundamentación Teórica.

Contiene todo el fundamento teórico de la investigación. El mismo abarcará los principales conceptos y términos a tratar a lo largo del documento. Se hace referencia a la fundamentación teórica del tema en el cual se explica la necesidad del presente trabajo, así como las metodologías de desarrollo de software, tecnologías, herramientas y lenguajes de programación candidatos.

Capítulo 2: Características y Propuestas del Sistema.

Se realiza un estudio del negocio para identificar los procesos fundamentales que se quieren automatizar. Se describen las historias de usuarios, así como el prototipo no funcional de la aplicación.

Capítulo 3: Diseño e Implementación del Sistema.

Se define la arquitectura del sistema y con ello el patrón arquitectónico empleado, así como las tareas de la ingeniería y tarjetas CRC. Además se realiza el modelo de datos. Se ofrece una visión general de la implementación y de los estándares de diseño.

Capítulo 4: Prueba.

Se realizan las pruebas a las principales interfaces de la aplicación, con el objetivo de dejar documentado si esta cumple con los requerimientos definidos en las primeras fases de la metodología propuesta.

## Capítulo 1. Fundamentación Teórica

### Introducción.

En el presente capítulo se realizará un estudio del estado del arte sobre los temas relacionados con la tecnología RFID haciendo énfasis en sus beneficios y aplicaciones, así como los conceptos teóricos que constituyen la base de la investigación realizada. Se analizan las características fundamentales de las tecnologías, metodologías y herramientas que se utilizarán en el desarrollo del sistema.

### 1.1 Conceptos Fundamentales:

#### 1.1.1 Tecnología de Identificación por Radiofrecuencia (RFID).

La tecnología RFID se basa en un sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas. Cuando las etiquetas entran en el área de cobertura de un lector, éste envía una señal para que las mismas le transmitan la información almacenada en su memoria. Una de las claves de la tecnología es que la recuperación de la información contenida en la etiqueta se realiza vía radiofrecuencia y sin necesidad de que exista contacto físico o visual (línea de vista) entre el dispositivo lector y las etiquetas. (1)

#### 1.1.2 Descripción de componentes.

Un sistema RFID incluye los siguientes componentes:

- ✓ La etiqueta, consiste en un pequeño circuito, integrado con una pequeña antena, capaz de transmitir un número de serie único hacia un dispositivo de lectura, como respuesta a una petición.
- ✓ El lector, (el cual puede ser de lectura o lectura/escritura) está compuesto por una antena, un módulo electrónico de radiofrecuencia y un módulo electrónico de control.
- ✓ Un controlador o un equipo anfitrión, comúnmente una PC, en la cual corre una base de datos (2)

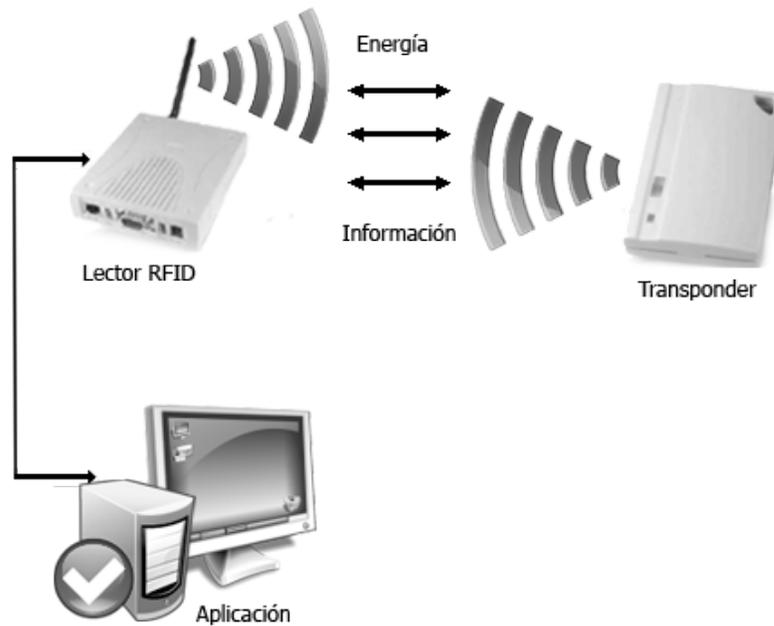


Figura 1.1: Componentes de un sistema RFID.

Todos los elementos mencionados anteriormente conforman un sistema RFID, sin embargo, a pesar de que los aspectos tecnológicos pueden variar, se basan en un principio y funcionamiento.

### 1.1.3 Principio y funcionamiento de un sistema RFID.

Como se había mencionado anteriormente para que la tecnología RFID funcione, son necesarios sus tres elementos básicos: una *etiqueta electrónica*, un *lector de etiquetas* y una *base de datos*. Las etiquetas electrónicas llevan un microchip<sup>3</sup> incorporado que almacena el código único identificativo del producto al que están adheridas. El lector envía una serie de ondas de radiofrecuencia a la etiqueta, que capta a través de una pequeña antena. Las ondas activan el microchip, que, mediante la micro - antena y la radiofrecuencia, transmite al lector cuál es el código único del artículo. En definitiva, un equipo lector envía una señal de interrogación a un conjunto de productos y estos responden enviando cada uno su número único de identificación. Una vez el lector ha recibido el código único del producto, lo almacena en una base de datos, donde se han recopilado previamente las características del artículo en cuestión: fecha de

---

<sup>3</sup>También llamado circuito integrado (CI) es una placa de silicio pequeña en la que se encuentran miles de dispositivos.

caducidad, material, peso, dimensiones, localización, etc., dependiendo también a que se aplique esta tecnología. De este modo se hace posible consultar la identidad de algo o alguien en cualquier momento, ya sea el caso de una aplicación a un producto o a una persona. (3)

Son muchos los sectores de la economía que se pueden beneficiar de las principales ventajas que ofrece esta tecnología.

#### 1.1.4 Ventajas de un sistema RFID.

- ✓ *Agilidad:* para hacer inventario es suficiente con pasar el lector portátil por las estanterías, se puede saber al instante qué materiales faltan, los que no están en su sitio, etc. y todo ello ahorrando tiempo y recursos.
- ✓ *Durabilidad ilimitada:* las etiquetas son reutilizables y retornables.
- ✓ *Eficacia:* Garantiza la identificación inequívoca de cada ítem.
- ✓ *Exactitud:* El seguimiento de accesos mediante contador es preciso y 100% fiable.
- ✓ *Amplitud:* Gracias al control geoespacial se puede conocer la ubicación de cada etiqueta dentro del edificio o recinto.
- ✓ *Flexibilidad:* Las posibilidades de la tecnología permiten su adaptación a cada negocio, agregando nuevos servicios o personalizándolos según cada situación.
- ✓ *Innovación:* capacidad de almacenamiento de datos.
- ✓ *Valor añadido:* funcionamiento en condiciones adversas (4)

Como se menciona anteriormente esta tecnología permite mejoras considerables, pero también presentan algunos inconvenientes de los cuales se le dará una breve descripción.

#### 1.1.5 Desventajas:

- ✓ *Costo:* Las etiquetas activas poseen su propia fuente autónoma de energía, que utilizan para dar corriente a sus circuitos integrados y propagar su señal al lector. Son mucho más fiables (tienen menos errores) que las etiquetas pasivos debido a su capacidad de establecer sesiones con el lector, son efectivos a distancias mayores pudiendo generar respuestas claras a partir de recepciones débiles (lo contrario que las etiquetas pasivas). Por tanto, suelen ser mayores y más caros.
- ✓ *Colisión:* Uno de los problemas con los sistemas de RFID es que la señal de uno de los lectores puede interferir con la señal de otro lector cuando se sobreponen sus áreas de cobertura. A esta

situación se la denomina colisión de lectores. Los lectores pueden usar blindajes, pero esto complica su instalación y los hace más caros. Las normas estandarizadas de la Gen 2 EPC incluyen algo que se denomina modo de "lector denso" para evitar la colisión de lectores. Lector Denso es una modalidad de operación que previene que los lectores que cumplen con las normas estandarizadas de la Gen 2 EPC<sup>4</sup>, interfieran entre sí cuando muchos de estos lectores se utilizan ubicándolos de manera cercana el uno del otro. (5)

#### 1.1.6 Aplicaciones de la tecnología RFID.

Los sistemas RFID tienen sus aplicaciones en varios sistemas industriales como:

- ✓ Control de calidad, producción y distribución.
- ✓ Localización y seguimiento de objetos.
- ✓ Control de accesos.
- ✓ Identificación de materiales.
- ✓ Control de fechas de caducidad.
- ✓ Detección de falsificaciones.
- ✓ Almacenaje de datos.
- ✓ Automatización de los procesos de fabricación.
- ✓ Información al consumidor.
- ✓ Reducción de tiempo y coste de fabricación.
- ✓ Reducción de colas a la hora de pasar por caja.
- ✓ Identificación y localización de animales perdidos.
- ✓ Elaboración de censos de animales.
- ✓ Identificación y control de equipajes en los aeropuertos.
- ✓ Inventario automático.
- ✓ Entre otras aplicaciones más. (6)

---

<sup>4</sup> Nombre abreviado de la segunda generación del protocolo de la interfaz aérea (el lenguaje que usan las etiquetas y los lectores para comunicarse).

### Software relacionado con la gestión de alertas usando la tecnología RFID.

#### 1.1.7 Sistema de localización de personal hospitalario mediante RFID activa.

El proyecto consiste en un sistema de localización para personal de un hospital psiquiátrico dentro de unas zonas delimitadas, el cual está preparado para realizar una multilectura, es decir, que cada lector RFID pueda identificar más de una etiqueta a la vez. La identificación se realiza a través de una etiqueta de pulsera RFID que la persona lleva en todo momento, el cual transmite los datos que contiene cuando detecta que está siendo interrogado por un lector RFID. La etiqueta se codifica con un identificador único (ID), lo que permite que las lecturas de las diferentes etiquetas RFID, vayan asociadas a la identificación individual. Los lectores identifican la etiqueta en el momento en que un usuario pulsa el botón de aviso. La señal de alarma es llevada al software de control que muestra en pantalla la información asignada a la etiqueta (nombre de usuario y otros datos que se consideren relevantes), así como la localización del mismo. (7)

#### Ventajas de la aplicación:

- ✓ Localización inmediata de todos los médicos.
- ✓ Pulsador de emergencia.
- ✓ No hay límite de usuarios ni zonas delimitadas.
- ✓ Sistema multilectura, permite identificar a más de una etiqueta al mismo tiempo.
- ✓ Detección de pérdida o no uso de la etiqueta pulsera RFID.

El sistema es totalmente fiable, eficaz y escalable en todos sus aspectos: usuarios, zonas de localización, registros de alertas, etc. Haciendo de él un sistema altamente seguro y flexible, ya que permite combinar otros sistemas de control de acceso y control de presencia al centro. (7)

#### 1.1.8 Sistema RFID para detectar fugas en estaciones de gas.

El proyecto consiste en un sistema para la detección de fugas de gas, el cual utiliza etiquetas RFID para enviar alertas tan pronto la filtración de combustibles sea detectada. El sistema RFID transmite los datos del sensor de los tanques subterráneos con un lector instalado en la superficie del suelo. La etiqueta se inserta en la cámara subterránea en la que está situado el depósito, en la boca de acceso. La etiqueta mide la presión entre dos paredes del tanque. Si la presión sobrepasa los límites establecidos, el dispositivo determina que puede haber ocurrido una fuga dentro de la pared interna. La etiqueta, transmite

un mensaje de alerta al lector instalado en la superficie por encima del tanque y el dispositivo alerta al centro de control de la estación de gas. La información de las lecturas puede ser almacenada con el fin de crear un registro de lecturas de los sensores de cada tanque. Los usuarios pueden compartir sus datos con el software de gestión propio, lo que permite a las empresas recibir mensajes de alerta de sus propios sistemas a través de teléfono móvil, mensaje de texto o correo electrónico. (8)

#### Desventajas de soluciones existentes:

Después del análisis de los sistemas anteriores se pueden observar varios inconvenientes. Las soluciones en la mayoría de los casos poseen limitantes de licencias de software a las que están sujetas, como por ejemplo muchos de los sistemas son propietarios o tienen un periodo en que expira esa licencia, teniendo que pagar nuevamente al pasar un tiempo para utilizar la misma. Por otra parte en muchos casos la realización de los sistemas se concibió con herramientas propietarias. Otra limitante es que no son multiplataforma, es decir, son compatibles solo con un sistema operativo, por lo que no son factibles y las soluciones existentes estudiadas no cumplen con todas las características del sistema que se propone realizar. Para esto se hace una selección de las tecnologías, herramientas y metodología de desarrollo.

#### 1.1.9 Tecnologías, herramientas y metodologías.

##### 1.1.10 Lenguaje de programación.

Un Lenguaje de Programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. Cada lenguaje posee sus propias sintaxis. (9)

Debido a las tendencias y necesidades, se han ido desarrollando diferentes lenguajes de programación para desarrollar en la web. A medida que pasó el tiempo, las tecnologías fueron evolucionando y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

Actualmente existen lenguajes de programación para desarrollar aplicaciones web, dependiendo del lado en el que se ejecutan, se pueden clasificar en lenguajes del lado del cliente o lenguajes del lado del servidor, siendo estos últimos los más importantes en el desarrollo de aplicaciones, donde se puede citar Java.

### Java.

Es un lenguaje potente que trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Sus características principales son: (10)

- ✓ Orientado a objetos
- ✓ Distribuido
- ✓ Interpretado
- ✓ Robusto
- ✓ Seguro
- ✓ Portable.
- ✓ De altas prestaciones
- ✓ Multitarea
- ✓ Dinámico

Diversas aplicaciones que son implementadas con el lenguaje de programación Java necesitan guardar grandes cantidades de datos por lo cual utilizan un sistema gestor de base datos que les permita almacenar la información en un servidor. Las bases de datos son un recurso de gran importancia para el desarrollo del conocimiento, del éxito profesional y de las organizaciones, permitiendo un almacenamiento y acceso confiable, eficiente y práctico en el uso de la información.

#### 1.1.11 Sistema Gestor de Base de Datos (SGBD).

Actualmente en el mundo existen diversas formas de almacenar la información, una de estas son las bases de datos. Una base de datos (BD) es un conjunto de datos relacionados entre sí, entendiéndose por dato los hechos conocidos, que pueden registrarse y que tienen significado implícito (11). Un Sistema Gestor o Manejador de Bases de Datos es un conjunto de programas que permite a los usuarios crear y mantener una base de datos, por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico. (11)

### PostgreSQL.

Es un potente servidor de base de datos relacional orientado a objetos de uso libre y de código abierto. Se puede ejecutar en la gran mayoría de sistemas operativos existentes en la actualidad, entre ellos Linux y Windows por lo que se considera que de los sistemas gestores de base de datos PostgreSQL es la opción ideal. Sus principales características son: (12)

- ✓ Soporta distintos tipos de datos:
- ✓ Incorpora una estructura de datos, los arreglos.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

### Ventajas:

- ✓ Instalación ilimitada.
- ✓ Mejor soporte que los proveedores comerciales.
- ✓ Ahorros considerables en costos de operación.
- ✓ Estabilidad y confiabilidad legendarias.
- ✓ Extensible.
- ✓ Multiplataforma.
- ✓ Diseñado para ambientes de alto volumen.
- ✓ Herramientas gráficas de diseño y administración de bases de datos.

Después de haber analizado el sistema gestor de base de datos se hace un estudio de las tecnologías del lado del servidor que puedan ser utilizadas para el desarrollo de la aplicación, resaltándose para el desarrollo de la misma Spring Framework e Hibernate.

#### 1.1.12 Tecnología del lado del servidor.

### Spring Framework.

Spring Framework es una plataforma que nos proporciona una infraestructura que actúa de soporte para desarrollar aplicaciones Java. Por su diseño el framework ofrece mucha libertad a los desarrolladores, soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable (13). Implementa el patrón MVC, separando la interfaz de usuario, los controladores y la capa de acceso a datos y utiliza el Objeto Módulo básico de la Plataforma de Servicios Bioinformáticos de Acceso a Datos (DAO, siglas en inglés de Data Access Object), que es un patrón de diseño para gestionar las bases de datos.

Spring proporciona:

- ✓ Una potente gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control. Esto hace que la configuración de aplicaciones sea rápida y sencilla. (13)
- ✓ Una capa genérica de abstracción para la gestión de transacciones, permitiendo gestores de transacción añadibles y haciendo sencilla la zona de transacciones sin tratarlas a bajo nivel. (13)
- ✓ Una capa de abstracción JDBC<sup>5</sup> que ofrece una significativa jerarquía de excepciones, simplifica el manejo de errores, y reduce considerablemente la cantidad de código necesario. (13)

---

<sup>5</sup> Java Database Connectivity, más conocida por sus siglas JDBC, permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

### Hibernate.

La selección del framework Hibernate está basada en que constituye una herramienta para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos que permiten establecer relaciones. Hibernate proporciona además un lenguaje para el manejo de consultas a la base de datos. El lenguaje es similar a SQL y es utilizado para obtener objetos de la base de datos según las condiciones especificadas en el HQL (Hibernate Query Language). El uso de HQL permite usar un lenguaje intermedio que según la base de datos que sea usada y el dialecto que se especifique será traducido al SQL dependiente de cada base de datos de forma automática y transparente. (14)

Para lograr un diseño atractivo de la aplicación se hace uso del framework JQuery el cual añade efectos visuales y animaciones para la creación de aplicaciones web.

#### 1.1.13 Tecnología del lado del cliente.

### JQuery.

Jquery es un framework de Java Script que permite simplificar la manera de actuar con el DOM del documento HTML con el que se esté trabando. Permite manejar eventos, desarrollar animaciones e interactuar con la tecnología AJAX. Ofrece una infraestructura con la que se tiene mayor facilidad para la creación de aplicaciones complejas del lado del cliente. (15)

### Ventajas:

- ✓ Ahorra muchas líneas de código.
- ✓ Hace transparente el soporte de una aplicación para los navegadores principales.
- ✓ Provee de un mecanismo para la captura de eventos.
- ✓ Provee un conjunto de funciones para animar el contenido de la página en forma muy sencilla.
- ✓ Integra funcionalidades para trabajar con AJAX. (15).

Para el correcto funcionamiento de la aplicación se hace un estudio de los diferentes IDEs de desarrollo seleccionando el NetBeans por ser un entorno de desarrollo completo y profesional con muchas funcionalidades para distintos tipos de aplicaciones y para facilitar al máximo la programación.

#### 1.1.14 Entorno de Desarrollo Integrado (Integrated Development Environment, IDE).

Un IDE es un programa informático compuesto por un conjunto de herramientas de programación, puede dedicarse exclusivamente a un solo lenguaje o bien puede utilizarse para varios. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicables por si solos o pueden ser parte de aplicaciones existentes.

NetBeans: Se ha seleccionado este IDE para el desarrollo de la herramienta debido a que cuenta con una gran cantidad de plugins que apoyan el desarrollo del sistema. Es un entorno de desarrollo integrado libre, presenta una arquitectura abierta y basada en plugins<sup>6</sup>, lo que le permite integrar diversos lenguajes sobre un mismo IDE e introducir otras aplicaciones accesorias. (16)

#### 1.1.15 Metodología de desarrollo de software.

Las nuevas tendencias en el mundo del desarrollo de software requieren de aplicaciones bajo un proceso bien definido y profundamente documentado para que cumpla con los parámetros que debe tener un producto informático. Para lograr el desarrollo exitoso de proyectos de software cada vez más complejos, surge la necesidad de crear métodos que permitan guiar, controlar y documentar el desarrollo de los mismos. De ahí surgen las metodologías, una metodología es aquella guía que se sigue con el fin de realizar las acciones propias de una investigación. CMMI<sup>7</sup> (Capability Maturity Model Integration, según sus siglas en inglés) plantea que su implementación aumenta la fiabilidad del software producido, la visibilidad de los procesos de producción y soporte, la reusabilidad de componentes, y como resultado de la combinación de este tipo de mejoras, disminuye los costes de producción y mantenimiento de las aplicaciones. En términos más sencillos se trata de la guía que va indicando qué hacer y cómo actuar cuando se quiere obtener algún tipo de investigación. Es posible definir una metodología como aquel enfoque que permite observar un problema de una forma total, sistemática y disciplinada. Según lo antes planteado un buen ejemplo de metodología de desarrollo de software aplicable es Programación Extrema (XP) ya que pone más énfasis en la adaptabilidad que en la previsibilidad.

En términos más sencillos se trata de la guía que va indicando qué hacer y cómo actuar cuando se quiere obtener algún tipo de investigación. Es posible definir una metodología como aquel enfoque que permite observar un problema de una forma total, sistemática y disciplinada.

---

<sup>6</sup> En español *complemento* es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

<sup>7</sup> Modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software.

### Programación Extrema (XP).

XP es la que mejor se ajusta a las condiciones del entorno y ofrece una alta capacidad organizativa para grupos de trabajo. A continuación se exponen las razones por las cual se seleccionó esta metodología:

XP tiene un desarrollo iterativo e incremental. Otras de las características es que la metodología es aplicada a proyectos pequeños y la programación se realiza en parejas, es flexible al cambio por lo que el cliente puede agregar nuevas historias de usuarios, dividir las o simplemente eliminarlas. XP permite al equipo la modificación de sus planes.

El cliente forma parte del equipo de desarrollo permitiendo una mayor retroalimentación y corrección de los errores posibles antes de añadir una nueva funcionalidad, logrando así un producto que satisfaga las necesidades del mismo.

#### 1.1.16 Herramienta case.

Las herramientas CASE (Computer Aided Software Engineering: Ingeniería de Software Asistida por Ordenador) son básicamente diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo en términos de tiempo y dinero (17) se resume en la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación.

### Visual Paradigm.

Se utiliza para la modelación de la base de datos del sistema, por las facilidades que brinda. Mejora el tiempo de construcción de aplicaciones de calidad, tiene la característica de permitir dibujar todos los tipos de diagramas de clases y la transformación del modelo entidad relación en las tablas de la base de datos. Facilita la ingeniería inversa y la codificación desde diagramas, así como el desarrollo de documentación de una aplicación. La abundante bibliografía, videos docentes y demostraciones interactivas encontradas, que explican el trabajo con la herramienta, es otra razón para usarla.

## 1.2 Conclusiones.

Como resultado del análisis realizado durante el presente capítulo, se pudo concluir que:

- ✓ Con el estudio de la situación actual del problema, plasmado en este capítulo, se muestra la utilidad y la necesidad de un sistema capaz de gestionar las alertas. Se profundizó en conceptos

relacionados con el tema y las condiciones actuales de estos tipos de sistemas, se ha recogido un resumen del estudio realizado sobre las tendencias y tecnologías actuales del desarrollo de software que son la base teórica y tecnológica de la solución propuesta.

- ✓ Los gestores de bases de datos mejoran la gestión de la información y permiten mantener un alto nivel de seguridad sobre el sistema, siendo PostgreSQL la solución por la seguridad y potencialidades que brinda.
- ✓ Para la selección de una metodología de desarrollo deben ser revisadas de forma estricta las características de la institución donde se va a llevar a cabo el proyecto. Es XP la que se ajusta al desarrollo de este sistema por las particulares del mismo.
- ✓ De una gran variedad de lenguajes de programación que existen en la actualidad, Java posee interfaces de programación que facilitan la creación de aplicaciones web y el NetBeans como IDE de desarrollo posee una plataforma de herramientas integradas.

## Capítulo 2. Características y Propuesta del Sistema.

### 2.1 Introducción.

En este capítulo se recogen las principales características del sistema a desarrollar. Se identificarán los procesos con el objetivo de entender aquellos que intervienen en el Sistema de Gestión de Alertas, para un correcto diseño e implementación. También se definen las historias de usuarios propias de la metodología seleccionada en la fase de planeación, así como los requerimientos no funcionales. Además se precisan las personas que intervienen en el proceso.

### 2.2 Propuesta del Sistema.

La solución propuesta es una aplicación web, la cual estará diseñada para aquellas instituciones que gestionen alertas asociadas a un lector RFID. El sistema contará con áreas de trabajo para cada uno de los usuarios que facilitarán la interacción de estos, dando cumplimiento a los requerimientos funcionales.

#### 2.2.1 Proceso de Gestión de Alertas.

Dentro del proceso de alertas se lleva a cabo un grupo de actividades que complementan el mismo. No solo se genera un listado de las alertas, sino que se les dará tratamiento en un período de tiempo. Además se llevará el control de trazas de los usuarios y un grupo de reportes.

#### Nuevas alertas.

El proceso de alertas comienza cuando el lector le envía una serie de ondas de radiofrecuencia a la etiqueta, los cuales capta a través de una pequeña antena, estas ondas activan el microchip que mediante la micro -antena y la radiofrecuencia le envían al lector el código identificativo de cada producto, persona o animal. Una vez que el lector recibe esta información es enviada a una base de datos donde serán almacenadas las características de cada una. A partir de este momento se genera la alerta, se actualiza el estado y se envía un SMS y correo electrónico con sus datos más significativos. Asociado al proceso se genera un conjunto de reportes, información que es muy importante para la toma de decisiones. Además se llevará el control de las estadísticas mediante gráficos. Luego se localizarán las antenas y los productos a través de un sistema de identificación geográfica donde se podrá visualizar la ubicación de cada uno de ellos.

### 2.3 Lista de Reserva del producto.

#### 2.3.1 Funcionalidades del Sistema.

##### **RF1:** Gestionar Alarmas.

- ✓ **RF1.1:** Generar Alarmas.
- ✓ **RF1.2:** Actualizar estado
- ✓ **RF1.3:** Enviar SMS

##### **RF2:** Generar Reportes.

- ✓ **RF2.1:** Crear Reportes
- ✓ **RF2.2:** Modificar Reportes

##### **RF3:** Realizar búsquedas.

- ✓ **RF3.1:** Usuarios
- ✓ **RF3.2:** Alarmas
- ✓ **RF3.3:** Trazas

##### **RF4:** Generar Estadísticas.

- ✓ **RF4.1:** Crear estadísticas relacionadas con las alarmas

##### **RF5:** Gestionar Centro.

- ✓ **RF5.1:** Adicionar centro.
- ✓ **RF5.2:** Modificar centro
- ✓ **RF5.3:** Desactivar centro

#### Módulo de Administración.

##### **RF6:** Gestionar Usuarios.

- ✓ **RF6.1:** Adicionar usuario
- ✓ **RF6.2:** Modificar usuario

- ✓ **RF6.3:** Desactivar usuarios

**RF7:** Gestionar funcionalidades.

- ✓ **RF7.1:** Adicionar
- ✓ **RF7.2:** Modificar
- ✓ **RF7.3:** Eliminar

**RF8:** Gestionar directivas de seguridad.

- ✓ **RF8.1:** Asignar privilegios
- ✓ **RF8.2:** Modificar privilegios
- ✓ **RF8.3:** Eliminar privilegios

**RF9:** Gestionar productos, personas y animales.

- ✓ **RF9.1:** Crear productos
- ✓ **RF9.2:** Modificar productos
- ✓ **RF9.3:** Mostrar productos
- ✓ **RF9.4:** Crear personas
- ✓ **RF9.5:** Modificar personas
- ✓ **RF9.6:** Mostrar personas
- ✓ **RF9.7:** Crear animal
- ✓ **RF9.8:** Modificar animal
- ✓ **RF9.9:** Mostrar animal.

2.3.2 Requisitos no funcionales del sistema.

Para lograr un correcto funcionamiento del sistema se deben tener en cuenta un conjunto de requisitos no funcionales. Los requisitos no funcionales son cualidades o propiedades que el producto debe tener. Estos se dividen en varias categorías:

Requisitos de usabilidad.

- ✓ Se necesitará un nivel bajo o medio en computación, ya que el manejo de la aplicación es sencillo, permitiendo al usuario una fácil comprensión y trabajo con el sistema.

- ✓ La aplicación garantizará una conexión segura y eficiente con la base de datos, que tendrá almacenada toda la información del sistema, mejorando la gestión de la información y los datos.

#### Requisitos de seguridad.

- ✓ El acceso a la aplicación se realizará a través del modelo RBAC (Control de Acceso Basado en Roles), teniendo solo acceso el personal autorizado y solo a las funcionalidades definidas.
- ✓ Los mecanismos de seguridad no deben impedir que los usuarios autorizados accedan a la información, debe estar disponible en todo momento.

#### Requisitos de soporte.

- ✓ El sistema permitirá la modificación o agregarle nuevos módulos cuando sea necesario, asegurando su extensibilidad y logrando mejores prestaciones.

#### Requisitos de eficiencia.

- ✓ La aplicación debe ser rápida en cuanto a la respuesta de los datos en el momento de realizar búsquedas y reportes de información.

#### Requisitos de hardware.

En las máquinas clientes:

- ✓ Memoria RAM: Mínimo 256 MB.
- ✓ Procesador: Intel Pentium 5 superior.

Para las máquinas servidor:

- ✓ Memoria RAM: Mínimo recomendada 512 MB.
- ✓ Tarjeta de Red: Ethernet a 100 Mbps o superior.
- ✓ Capacidad de almacenamiento en disco duro: Mínimo 250 GB.

#### Requisitos de software.

En las máquinas clientes es necesario:

- ✓ Navegador Internet Explorer 6 o superior.
- ✓ Recomendado Mozilla Firefox 3.5 o superior.

### Requisitos de apariencia o interfaz de usuario.

- ✓ Las interfaces de la aplicación contarán con los componentes visuales necesarios para las operaciones correspondientes, evitando la sobrecarga de imágenes.
- ✓ El sistema debe mostrar una interfaz profesional, sin información repetida, ni en exceso.
- ✓ El diseño debe permitir al usuario navegar con facilidad por la aplicación.

### Personas relacionadas con el Sistema.

Se definen como personas relacionadas con el sistema aquellas personas que administran las funcionalidades y obtienen resultados de todas, entre ellas el administrador, el cual es el encargado de gestionar toda la información, así como hacer modificaciones en el momento que lo requiera. Otras de las personas que se relacionan son los técnicos que son los encargados de llevar a cabo el tratamiento de las alertas.

## 2.4 Planeación.

La fase de planeación es la primera fase definida por la metodología XP. La actividad de planeación comienza con la creación de las historias de usuarios, las cuales describen las características y funcionalidades requeridas para el software que se construirá.

### 2.4.1 Historias de Usuario.

Las historias de usuario son similares a las descripciones de casos de usos en RUP<sup>8</sup>, con la diferencia de que son descritas por el propio cliente en dependencia de las necesidades del sistema.

El cliente es el encargado de darle una prioridad a cada historia de usuario y el equipo de desarrollo es quien evalúa las historias y le da un costo, este se traduce en las semanas que se llevará el desarrollo de la misma.

Si según lo planificado, la historia requiere más de tres semanas de desarrollo se les sugiere al cliente que las divida en historias más pequeñas, dándoles a las nuevas una prioridad y un costo. También es importante destacar que las historias de usuarios nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología.

---

<sup>8</sup> **Proceso Unificado Racional** (*Rational Unified Process* en inglés) es un proceso de desarrollo de software.

#### 2.4.2 Campos de las historias de usuarios.

A las historias de usuario se les asignan una prioridad del negocio y un riesgo de desarrollo, los cuales pueden ser clasificados como se muestra a continuación:

Prioridad en negocio: Prioridad que se le asigna a la historia de usuario en el negocio. Si es importante debe ser implementada lo antes posible, se clasifica en:

- ✓ *Alta:* Se le otorga a las historias de usuarios que resultan funcionalidades fundamentales en el desarrollo del sistema, las que el cliente define como principales para el control integral del sistema.
- ✓ *Media:* Se le otorga a las historias de usuarios que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
- ✓ *Baja:* Se le otorga a las historias de usuarios que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

Riesgo en desarrollo: Riesgo que representa para el desarrollo la historia de usuario. En dependencia del riesgo que represente se le asigna un valor que puede ser:

- ✓ *Alta:* Cuando en la implementación de las historias de usuarios se consideran la posible existencia de errores que lleven la inoperatividad del código.
- ✓ *Media:* Cuando pueden aparecer errores en la implementación de la historias de usuarios que puedan retrasar la entrega de la versión.
- ✓ *Baja:* Cuando pueden aparecer errores que serán tratados con relativa facilidad, sin que traigan perjuicios para el desarrollo del proyecto.

#### 2.4.3 Campos de las historias de usuarios.

A continuación se exponen algunas de las historias de usuarios de mayor prioridad para el desarrollo del sistema. Para ver las demás historias de usuario, [ver anexo1](#).

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Generar alarma

<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Se genera una alarma cuando el sistema detecta una anomalía.	
<b>Observaciones:</b>	
<b>Prototipo de interface</b>	

Tabla 2.1: Historia de Usuario Generar Alarma.

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre de Historia de Usuario:</b> Actualizar estado de las alarmas
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1

<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> El técnico después de haberse autenticado actualiza el estado de la alarma en un período de tiempo determinado como constancia de su tratamiento.	
<b>Observaciones:</b> Si la alarma no es tratada en el período de tiempo establecido pasa al estado de espera para poder ser atendida nuevamente	
<b>Prototipo de interface</b>	

Tabla 2.2: Historia de Usuario Actualizar estado de la alarma.

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre de Historia de Usuario:</b> Enviar SMS y correo electrónico
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1

<p><b>Descripción:</b> Se envía un SMS y un correo electrónico al administrador del sistema en caso que se genere la alarma.</p>
<p><b>Observaciones:</b></p>
<p><b>Prototipo de interface</b></p>

Tabla 2.3: Historia de Usuario Enviar mensaje de texto y correo electrónico.

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre de Historia de Usuario:</b> Mostrar reportes
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> El sistema será capaz de mostrar la información relacionada con las alarmas.	
<b>Observaciones:</b>	

<p><b>Prototipo de interface</b></p>
--------------------------------------

Tabla 2.4: Historia de Usuario Mostrar reportes.

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre de Historia de Usuario:</b> Búsquedas filtradas
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<p><b>Descripción:</b> El administrador una vez autenticado, puede realizar búsquedas filtradas introduciendo los criterios de búsquedas.</p>	
<p><b>Observaciones:</b> Si no encuentra resultados de la búsqueda se mostrará un mensaje</p>	
<p><b>Prototipo de interface</b></p>	

Tabla 2.5: Historia de Usuario Búsquedas filtradas.

### 2.5 Plan de iteraciones.

El plan de iteraciones es uno de los artefactos propios de la metodología seleccionada. La tabla del plan de iteraciones recoge un conjunto de información, donde se especifica la iteración en la cual será desarrollado la historia de usuario, la descripción de la iteración, las historias de usuarios a implementar y el tiempo de duración. En la siguiente tabla se recogen los datos, de la cual se puede concluir que la duración total del desarrollo del sistema es de 12 semanas.

Iteraciones	Descripción de la Iteración	Orden de la HU a implementar	Duración total
1	Creación de un sistema capaz de permitir la autenticación de los usuarios de forma segura.	<ul style="list-style-type: none"><li>• Autenticar usuario</li></ul>	1 semana
1	Creación de un sistema capaz de gestionar las alarmas que van llegando según su complejidad.	<ul style="list-style-type: none"><li>• Generar Alarmas</li><li>• Actualizar estado</li><li>• Enviar SMS y correo electrónico</li></ul>	1 semana
1	Creación de un sistema capaz de generar estadísticas según su tipo.	<ul style="list-style-type: none"><li>• Crear estadísticas de Alarmas</li></ul>	2 semanas
1	Creación de un sistema capaz de localizar antenas y productos de una determinada institución	<ul style="list-style-type: none"><li>• Localizar Antenas en un mapa</li></ul>	2 semanas
1	Creación de un sistema capaz de gestionar directivas de seguridad	<ul style="list-style-type: none"><li>• Asignar Privilegios</li><li>• Modificar Privilegios</li></ul>	1 semanas

1	Creación de un sistema capaz de gestionar los locales y antenas de una determinada institución para mantener el control de objetos, animales, productos y personas	<ul style="list-style-type: none"> <li>• Crear productos</li> <li>• Modificar productos</li> <li>• Desactivar productos</li> <li>• Crear persona</li> <li>• Modificar persona</li> <li>• Desactivar persona</li> </ul>	2 semanas
1	Elaboración de un sistema capaz de generar reportes de información, a partir de los datos suministrados por el administrador del sistema.	<ul style="list-style-type: none"> <li>• Crear reportes</li> <li>• Mostrar reportes</li> </ul>	1 semanas
2	Elaboración de un sistema capaz de realizar búsquedas filtradas a partir de los criterios de búsquedas definidos por el usuario.	<ul style="list-style-type: none"> <li>• Buscar Usuarios</li> <li>• Buscar Alertas</li> <li>• Buscar Nombre de usuarios</li> <li>• Buscar centro</li> </ul>	2 semanas

Tabla 2.6: Plan de Iteraciones.

## 2.6 Conclusiones.

- ✓ Luego de abordar en este capítulo la solución propuesta para la implementación del sistema de alertas se puede concluir lo siguiente: La descripción de las Historias de Usuarios serán un aspecto fundamental y de gran ayuda en el desarrollo del sistema además se espera que teniendo en cuenta la descripción anterior la implementación sea efectuada de forma eficiente permitiendo un producto final con una buena calidad.
- ✓ A partir de las historias se realizó toda la planificación de entrega del software, la misma se efectuó conjuntamente con el cliente que es el actor principal en la etapa de planificación de la metodología seleccionada.

- ✓ Se analizaron y describieron los requisitos funcionales y no funcionales, los cuales son de gran importancia para el desarrollo del sistema ya que constituyen las funcionalidades del mismo.

.

.

## Capítulo 3: Diseño e Implementación del Sistema.

### 3.1 Introducción.

Para el desarrollo del sistema propuesto se identifica el patrón arquitectónico, la arquitectura utilizada y se precisan las Tarjetas Clase - Responsabilidad - Colaborador (CRC). Además se realiza el diseño de la base de datos y se definen las tareas de la ingeniería a desarrollar. También se aborda los estándares de codificación empleados para la implementación del sistema.

### 3.2 Patrón arquitectónico: Modelo Vista Controlador (MVC).

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. El MVC separa el sistema en tres capas fundamentales, una capa de modelo, que representa los datos relacionados con la aplicación, una capa de vista o presentación, encargada de representar los datos del modelo al usuario, mediante la cual se permitirá manipular la información y una capa controladora que procesa las peticiones que vienen desde la capa de presentación y donde está encapsulada la lógica del negocio. (18)

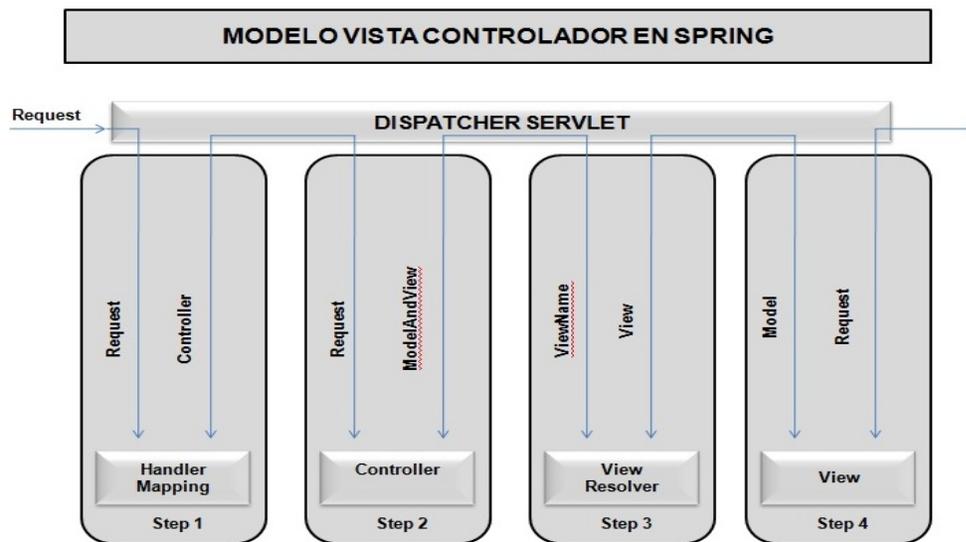


Figura 3.1: Patrón Arquitectónico: Modelo-Vista-Controlador.

Diseño arquitectónico.

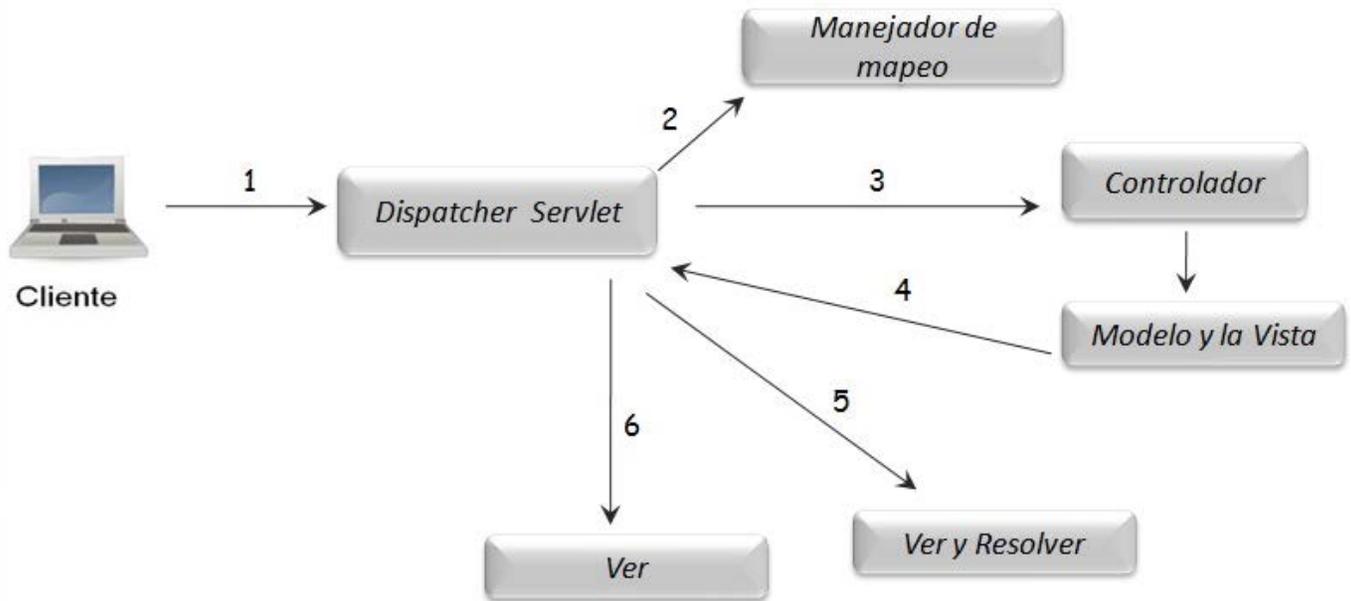


Figura 3.2: Diseño Arquitectónico.

El patrón MVC en Spring funciona de la siguiente manera:

Todas las peticiones son recibidas por el DispatcherServlet, este con el HandlerMapping(Manejador de Mapeo) identifica al controlador que procesará dicha petición, le pasa el control y obtiene como respuesta un ModelAndView(Modelo y la Vista), de este toma el nombre de la vista y utilizando el ViewResolver(Ver y Resolver) encuentra la vista física (archivo JSP), a la que le envía los datos extraídos del ModelAndView para que sea dibujada (rendered), resultado de lo cual obtiene el código HTML que es enviado como respuesta al cliente.

- ✓ *DispatcherServlet*: Las peticiones de los clientes son recibidas por el DispatcherServlet del Framework Spring, quien es el punto de entrada a la implementación del patrón MVC en Spring. En esencia lo que realiza es pasar el control de las peticiones a los Controladores (Controller).
- ✓ *HandlerMapping*: Permite mapear las peticiones HTTP, utilizando los URLs o partes de estos, a los controladores que las procesarán; también contiene de forma opcional interceptores que pueden

ser invocados antes o después de efectuarse el mapeo. Se tienen varias opciones a utilizar, independientes o combinadas y que son configuradas en archivos XML.

- ✓ *Controller*: Al recibir las peticiones HTTP, ejecutan código Java que realiza la lógica del sistema; manipula a los objetos DAO involucrados y decide que Vista usar para el despliegue de los resultados.
- ✓ *ModelAndView*: Un objeto de esta clase engloba los datos a mostrar (el modelo, Model) en un objeto de la clase Map y el nombre de la vista que mostrará dichos datos, este nombre es en general un alias, o sea, no apunta directamente a un archivo físico (JSP).
- ✓ *ViewResolver*: Esta clase es utilizada para resolver a partir del nombre de la vista, el alias contenido en el ModelAndView, la vista física que será dibujada (rendered) con los datos que le son pasados. Se tienen varias opciones a utilizar, independientes o combinadas y que son configuradas en archivos XML.
- ✓ *View*: Este término se refiere a las vistas físicas, básicamente a los archivos JSP, PDF, XLS. (19)

### 3.3 Arquitectura del Sistema.

#### 3.3.1 Arquitectura Cliente/Servidor.

La arquitectura Cliente/Servidor es el procesamiento de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales. Desde el punto de vista funcional, se puede definir la arquitectura Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma. (20)

##### 3.3.1.1 Componentes de la arquitectura Cliente/Servidor.

Los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas Cliente/Servidor son:

- ✓ El proceso cliente que es quien inicia el diálogo.
- ✓ El proceso servidor que pasivamente espera a que lleguen peticiones de servicio.

- ✓ El middleware que corresponde a la interfaz que provee la conectividad entre el cliente y el servidor para poder intercambiar mensajes. (20)

#### 3.3.1.2 Elementos principales.

Entre los elementos principales que componen la arquitectura están:

##### Cliente.

Es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos: (20)

- ✓ Administrar la interfaz de usuario.
- ✓ Interactuar con el usuario.
- ✓ Procesar la lógica de la aplicación y hacer validaciones locales.
- ✓ Generar requerimientos de base de datos.
- ✓ Recibir resultados del servidor.
- ✓ Formatear resultados.

##### Servidor.

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se enumeran a continuación: (20)

- ✓ Aceptar los requerimientos de base de datos que hacen los clientes.
- ✓ Procesar requerimientos de base de datos.
- ✓ Formatear datos para transmitirlos a los clientes.
- ✓ Procesar la lógica de la aplicación y realizar validaciones a nivel de base de datos.

### 3.4 Diseño de la Base de Datos.

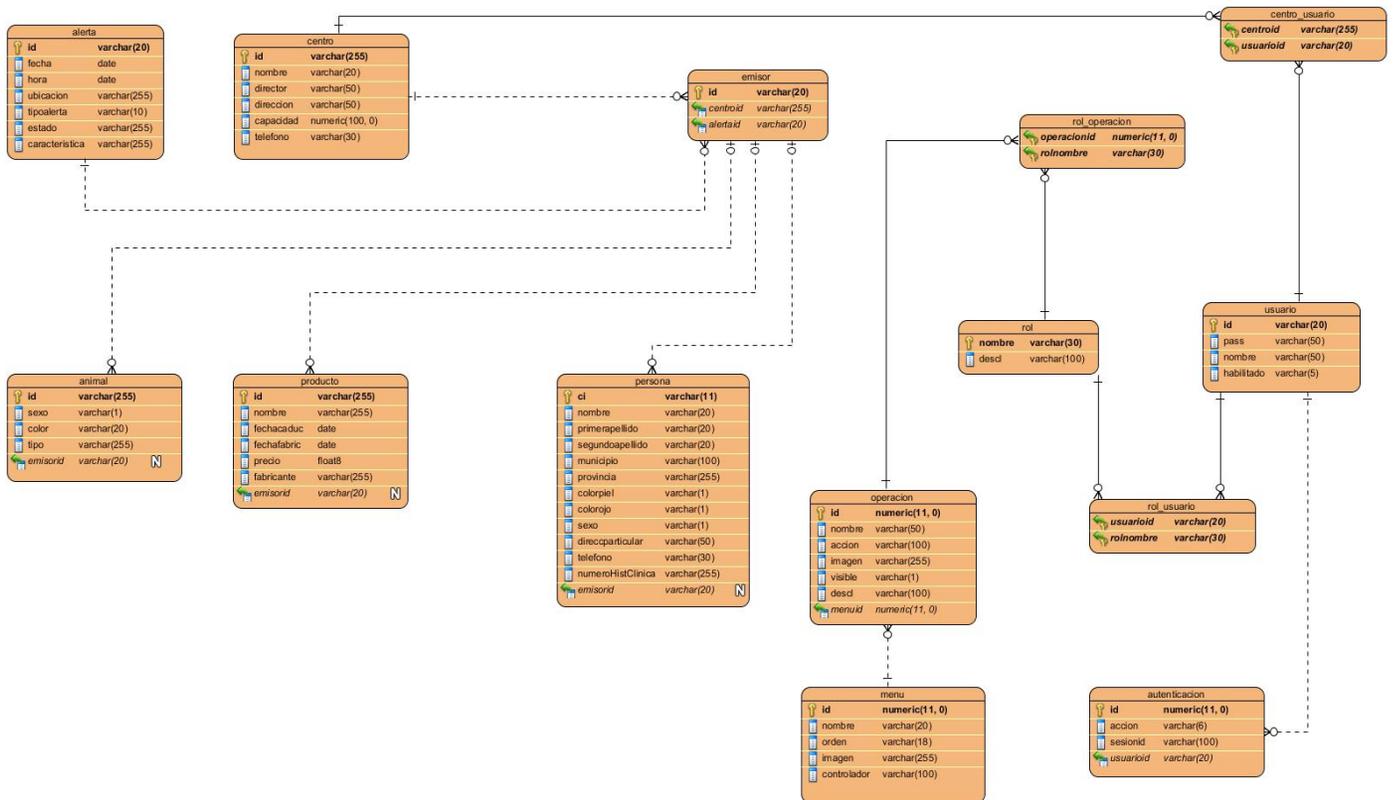


Figura 3.3: Modelo Entidad Relación.

El Diagrama Entidad Relación muestra el modelo de datos utilizado en la aplicación, representando con las entidades, los objetos y elementos principales identificados en el problema y la relación que existe entre ellos. Las entidades poseen atributos que son sus características particulares.

### 3.5 Tareas de Ingeniería.

Durante el transcurso de las iteraciones, se realiza la implementación de las historias de usuarios definidas por el cliente y descritas por el equipo de desarrollo en la etapa de Exploración. Como parte del plan, se descomponen las historias de usuarios en tareas de la ingeniería, las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente. Una historia de usuario puede tener asociada varias tareas de ingeniería para poder darle respuesta a las necesidades de su cliente.

### 3.5.1 Campos de las tareas de ingeniería.

El tipo de tarjeta que se propone adoptar está compuesta por un grupo de campos, entre estos el tipo de tarea que puede ser:

- ✓ *Desarrollo*: Tarea que se realizará por primera vez.
- ✓ *Corrección*: Tarea que se realiza a partir de una anterior que no se realizó correctamente.
- ✓ *Mejora*: Tarea que se realiza a partir de una anterior que se realizó correctamente, pero se incorporan nuevos requerimientos para la misma.
- ✓ *Otra*: Tarea que no corresponde con ninguna de las anteriormente descritas.

### 3.5.2 Tareas de Ingeniería a desarrollar.

A continuación las tareas de la ingeniería asociadas a las historias de usuario descritas anteriormente. Para ver las demás tareas de ingeniería [ver anexo 2](#).

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Implementar la funcionalidad "Generar alarma".	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 4/6
<b>Fecha Inicio:</b> 7/1/2013	<b>Fecha Fin:</b> 13/1/2013
<b>Programador Responsable:</b> Maylin Oña Bernabeu. Norey Fernandez De La Torre.	
<b>Descripción:</b> Cuando algún producto, animal o persona que tiene adherido una etiqueta RFID presenta alguna anomalía como arribo a la fecha de vencimiento, cambio de lugar o extravío la etiqueta asume que se ha emitido una alerta por lo que envía una señal de alerta al lector y este a su vez envía una señal al software de control generándose de esta manera una alerta.	

Tabla 3.1: Tarea de Ingeniería Gestionar alarma.

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 2
<b>Nombre Tarea:</b> Implementar la funcionalidad “Actualizar estado de las alarmas”.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 4/6
<b>Fecha Inicio:</b> 14/1/2013	<b>Fecha Fin:</b> 20/1/2013
<b>Programador Responsable:</b> Maylin Oña Bernabeu. Norey Fernandez De La Torre.	
<b>Descripción:</b> El técnico es el encargado de actualizar el estado de las alarmas. Una vez que la alarma se ha generado, su estado es <i>en espera</i> , mientras se le da tratamiento se encuentra en el estado <i>en proceso</i> y luego de ser tratada pasa al estado <i>de atendidas</i> .	

Tabla 3.2: Tarea de Ingeniería Actualizar estado de las alarmas.

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 3
<b>Nombre Tarea:</b> Implementar la funcionalidad “Enviar mensaje de texto y correo electrónico”.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 4/6
<b>Fecha Inicio:</b> 21/1/2013	<b>Fecha Fin:</b> 27/1/2013
<b>Programador Responsable:</b> Maylin Oña Bernabeu. Norey Fernandez De La Torre.	
<b>Descripción:</b> Cuando se genera la alarma el sistema envía un mensaje de texto y un correo electrónico al administrador como constancia de que se ha generado la alarma.	

Tabla 3.3: Tarea de Ingeniería Enviar mensaje de texto y correo electrónico.

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 4
<b>Nombre Tarea:</b> Implementar la funcionalidad “Mostrar reportes”.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 4/6
<b>Fecha Inicio:</b> 28/1/2013	<b>Fecha Fin:</b> 3/2/2013
<b>Programador Responsable:</b> Maylin Oña Bernabeu. Norey Fernandez De La Torre.	
<b>Descripción:</b> Cuando se genera una alarma el sistema crea un reporte con sus principales características.	

Tabla 3.4: Tarea de Ingeniería Mostrar reportes.

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 5
<b>Nombre Tarea:</b> Implementar la funcionalidad “Búsquedas filtradas”.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 4/6
<b>Fecha Inicio:</b> 4/2/2013	<b>Fecha Fin:</b> 10/2/2013
<b>Programador Responsable:</b> Maylin Oña Bernabeu. Norey Fernandez De La Torre.	
<b>Descripción:</b> El administrador podrá realizar búsquedas filtradas introduciendo un criterio de búsqueda.	

Tabla 3.5: Tarea de Ingeniería Búsquedas filtradas.

### 3.6 Principios de Diseño.

El diseño de la interfaz de una aplicación es uno de los elementos más importante en el desarrollo de software, ya que a través de esta es que el usuario se comunica con el sistema.

#### 3.6.1 Estándares de la interfaz de la aplicación.

Para la creación de las diferentes páginas en las que se gestiona la información se utilizó el entorno de desarrollo NetBeans, el mismo está formado por un conjunto de herramientas para la creación de aplicaciones web con JQuery, CSS, HTML, Hibernate y Spring Web permitiendo la creación de páginas uniformes. Los colores han sido utilizados dada la funcionalidad y objetivo del sistema, logrando una vista agradable a los usuarios. Los usuarios tendrán acceso a las funcionalidades teniendo en cuenta el rol que desempeñan, para esto cuando el usuario se autentique según su rol será el permiso que tiene asociado y teniendo en cuenta esto, se creará el menú.

### 3.7 Diagrama de Despliegue.

El Diagrama de Despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema propuesto. Cada nodo representa un recurso que interviene en el despliegue del sistema.

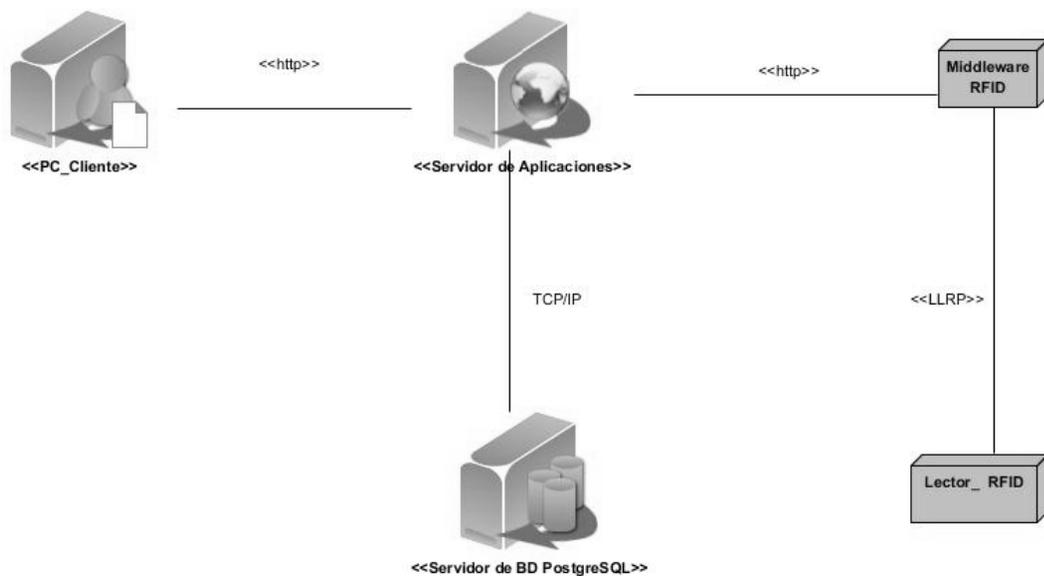


Figura 3.4: Diagrama de Despliegue.

### 3.8 Tarjetas Clase-Responsabilidad-Control (CRC).

El modelo de Clase – Responsabilidad – Colaborador proporciona un medio simple para identificar y organizar las clases relevantes para los requisitos del sistema o el producto.

Un modelo CRC es una colección de tarjetas índice estándar que representan clases. Las tarjetas se dividen en tres secciones. A lo largo del borde superior de la tarjeta se escribe el nombre de la clase, en el cuerpo de la tarjeta se listan las responsabilidades de la clase a la izquierda y los colaboradores a la derecha. (21)

#### 3.8.1 Tarjetas CRC.

El sistema está compuesto por varias clases y cada una tiene implementada las funcionalidades de las que son responsables. A continuación se muestra una de las clases que contribuye a realizar el proceso de gestionar usuarios. Para ver las demás clases [ver anexo 3.](#)

Clase :IUsuarioDao	
Responsabilidad	Colaboración
adicionarUsuario(usuario u)	UsuarioDaoImpl,usuario,UsuarioManagerImpl, IUsuarioManager, UpdateUserController, rol_usuario
eliminarUsuario(usuario u)	UsuarioDaoImpl,usuario,UsuarioManagerImpl, IUsuarioManager, UpdateUserController, rol_usuario
listarUsuario()	UsuarioDaoImpl,usuario,UsuarioManagerImpl, IUsuarioManager, UpdateUserController, rol_usuario

Tabla 3.6: Tarjeta CRC Clase IUsuarioDao.

### 3.9 Patrones.

Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos. (22)

#### 3.9.1 Patrones de Diseño.

Para evitar algunos problemas en el desarrollo de software se determinaron los patrones a utilizar con el fin de darle una solución a los mismos. Para la implementación del sistema se tuvo en cuenta el uso de los

patrones GRASP (General Responsibility Assignment Software Patterns), también conocidos como patrones para la asignación de responsabilidades.

Los patrones GRASP son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades, constituyendo un apoyo que ayuda a entender el diseño de objetos. Para el desarrollo de la aplicación se tuvo en cuenta los siguientes patrones GRASP:(22)

- ✓ *Patrón Experto*: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada como se evidencia en la clase Clase AnimalDaoImpl.
- ✓ *Patrón Creador*: Asigna a la clase B la responsabilidad de crear una instancia de la clase A bajo la condición de que B agrega, contiene, registra o usa objetos de la clase A, o tiene la información inicial que será pasada a A cuando es creada tal como se demuestra en la clase AddAnimalController.
- ✓ *Alta Cohesión*: Hace referencia a que la información que almacena una clase debe ser coherente y debe estar (en la medida de lo posible) relacionada con la clase.
- ✓ *Bajo Acoplamiento*: Expone las ideas 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 las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. (23)

Para la comunicación de los componentes del software se definió la Clase Controladora (CC) llamada SlavarController.java encargada de gestionar todas las peticiones del usuario y de determinar el controlador que le dará respuesta a dicha petición. De esta manera se aplican patrones Bajo acoplamiento y Alta cohesión.

### 3.10 Conclusiones.

- ✓ En este capítulo se trataron temas importantes para la construcción del sistema propuesto como el patrón arquitectónico, los patrones de diseño y la arquitectura a utilizar los cuales contribuyen a lograr una arquitectura estable y sólida y a crear un plano del modelo de implementación.

- ✓ Esta etapa de desarrollo se caracteriza por resultados ya visibles para los clientes y gratificantes para los desarrolladores, ya que queda implementada la aplicación con las principales funcionalidades que se definieron para la iteración del producto.
- ✓ Se referencia todo lo concerniente a la fase de Diseño e Implementación, como la metodología utilizada señala. Para ello se realizó la descripción de las tarjetas CRC, logrando un enfoque en la programación orientada a objetos, y de esta forma garantizando una mejor calidad del código en la fase de implementación, y se descompusieron las historias de usuarios en tareas de ingeniería a las cuales se le asignaron tiempos de implementación que fueron cumplidos cabalmente garantizando el objetivo principal de su realización.

## Capítulo 4. Pruebas.

### 4.1 Introducción.

En las diferentes metodologías la fase de prueba es la última y no por esto deja de ser la menos importante, al contrario las pruebas son la forma de medir si se obtuvieron buenos resultados y si las funcionalidades cumplen con lo que se definió en un inicio. En la mayoría de los casos la última etapa es la denominada Prueba, aunque durante todo el ciclo de desarrollo de software, los desarrolladores realizan un grupo de pruebas, para ir comprobando que no existan errores en el código. En el presente capítulo se abordan temas relacionados con la fase de prueba, así como algunos tipos de pruebas existentes con el objetivo de que el sistema cumpla con los requerimientos establecidos.

### 4.2 Pruebas de Software.

En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. Las pruebas de software involucran las operaciones del sistema bajo condiciones controladas y evaluando los resultados. Las condiciones controladas pueden ser normales o anormales. La prueba puede intencionalmente esforzar el programa y producir errores en las respuestas para determinar si los sucesos ocurren cuando no tendrían que ocurrir o cuando los hechos no suceden cuando deberían. (24)

### 4.3 Verificación y Validación.

La verificación y la validación son conceptos fundamentales asociados a las pruebas, logrando con esto una revisión exhaustiva. La verificación incluye por parte de los desarrolladores la revisión del código, de los requerimientos, de la documentación, las especificaciones y posteriormente una reunión con los usuarios para evaluar dichos documentos y en el segundo caso incluye las pruebas del software y comienza después que la verificación este completa.

### 4.4 Tipos de pruebas.

Existen varios tipos de pruebas que se utilizan para la realización de un software, entre ellas se encuentran:

- ✓ *Pruebas unitarias:* Se encargan de probar una clase en concreto, testeando cada uno de sus métodos y viendo si dado unos parámetros de entrada, la salida es la esperada.

- ✓ *Pruebas funcionales:* Estas permiten probar una funcionalidad completa, donde pueden estar implicadas una o varias clases y la propia interfaz del usuario.
- ✓ *Pruebas de regresión:* Son aquellas pruebas cuyo objetivo es comprobar por que ha dejado de funcionar algo que anteriormente estaba en funcionamiento.
- ✓ *Pruebas de aceptación:* Son pruebas funcionales, pero vistas directamente desde el cliente, son aquellas pruebas que demuestran al cliente que la funcionalidad está terminada y funciona de forma correcta. (25)

Los tipos de pruebas mencionados examinan el software de diferentes formas, pero en todos los casos con un mismo objetivo, comprender el funcionamiento de la aplicación ya sea revisando sus métodos o la interfaz.

#### 4.5 Tipo de prueba seleccionada.

Después de haber analizado los tipos de pruebas mencionados anteriormente, se decidió utilizar las pruebas de aceptación, ya que son las que más se ajustan a lo que se quiere comprobar de la aplicación. El objetivo de las pruebas es validar que un sistema cumple con el funcionamiento esperado, permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. Son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario. (26)

#### 4.6 Casos de Pruebas.

Después de haber seleccionado el tipo de prueba a realizar, se decidió aplicar dichas pruebas solo a la funcionalidad relacionada con el proceso de los usuarios, en este caso adicionar, modificar, buscar, desactivar y mostrar. Para ver las demás pruebas consulte el [anexo 4](#).

Caso de Prueba de Aceptación	
<b>Código caso de prueba:</b>	<b>Nombre HU:</b> Gestionar Usuarios
<b>Nombre de la persona que realiza la prueba:</b> Maylin Oña Bernabeu.	
<b>Descripción de la prueba:</b> Se debe comprobar que se adicione un usuario, así como modificar sus datos, desactivarlo y mostrarlos. Si los datos introducidos tienen errores o los campos que son obligatorios se dejan en blanco, el sistema muestra un mensaje con el tipo de información que se requiere para corregir el error. En el caso de estar correcto los datos, se guardan en la base de datos	

satisfactoriamente.

**Condiciones de ejecución:**

- La aplicación no puede tener errores de ningún tipo en su compilación.
- El usuario que tiene el rol de administrador es quien tiene acceso a la funcionalidad y al centro al que pertenece.

**Entrada / Pasos de ejecución:**

**“Escenario Adicionar Usuarios”**

Se selecciona la opción adicionar usuario, mostrándose un formulario con los campos correspondientes a este: usuario, nombre, contraseña, confirmar contraseña, rol y habilitado, posteriormente se selecciona el botón Adicionar, quedando añadido el nuevo usuario.

**“Escenario Modificar Usuarios”**

Se selecciona la opción Modificar Usuario mostrándose un formulario con los campos correspondientes, debe permitirse modificar cualquiera de los datos pertenecientes al usuario seleccionado y posteriormente se selecciona el botón Modificar.

**“Escenario Desactivar Usuarios”**

En el listado de usuarios, con sus datos creados anteriormente, con la opción clic derecho se selecciona el ícono desactivar. Este desactiva el usuario y actualiza los datos, permitiendo volver a la lista.

**“Escenario Mostrar Usuarios”**

Se selecciona la opción Mostrar Usuario, mostrándose una tabla con los datos de los usuarios existentes en el sistema.

**“Escenario Buscar Usuarios”**

Se muestra un buscador, el usuario introduce el nombre que quiere buscar o el id, así como la contraseña y si está habilitado o no en el sistema. También se puede realizar búsquedas filtradas por diferentes criterios.

**\\\\Resultado esperado:**

**“Escenario Adicionar Usuarios”**

En el listado de usuarios se muestra el usuario añadido.

**“Escenario Modificar Usuarios”**

Se modifica correctamente los datos del usuario en la base de datos y se muestra en el listado de

usuarios, se validan las entradas de datos en los campos restringidos y obligatorios.

**“Escenario Desactivar Usuarios”**

Se desactiva el usuario seleccionado, se muestra un mensaje de confirmación y después de haberse desactivado, se muestra el listado de los usuarios.

**“Escenario Mostrar Usuarios”**

Se muestra un listado con todos los usuarios.

**“Escenario Buscar Usuarios”**

El usuario introduce en el buscador un criterio de búsqueda y el sistema muestra el usuario existente.

**Evaluación de la Prueba:** Satisfactoria

*Tabla 4.1: Caso de Prueba Gestionar Usuario.*

4.7 No Conformidades.

La plantilla de No Conformidades (NC) recoge los errores que son detectados durante la revisión de la documentación del sistema. Se elabora un documento por cada revisión que se haga y se controlan a través de versiones según se vayan eliminando los errores, hasta que finalmente se hayan erradicado todos los defectos que posea el elemento que se prueba. En la siguiente tabla se expone un resumen de los resultados de las NC obtenidos distinguiendo de ellas cuáles proceden, cuáles no proceden y las resueltas, resaltando que las NC encontradas no constituyen un riesgo de desarrollo alto para el funcionamiento del sistema.

<b>Iteraciones</b>	<b>Total de NC</b>	<b>#NC No Proceden</b>	<b>#NC Resueltas</b>
<b>1</b>	15	8	7
<b>2</b>	8	2	6
<b>3</b>	2	0	2

*Tabla 4.2: Resumen de No Conformidades.*

#### 4.8 Conclusiones.

- ✓ Al culminar el capítulo se le ha dado cumplimiento a las últimas tareas trazadas para el desarrollo de la aplicación, quedando totalmente implementada la aplicación web.
- ✓ Se hizo referencia a todo lo concerniente a la fase de Prueba de la metodología XP. Se realizó la descripción de las pruebas de aceptación a la que fue sometida la aplicación con la intención de verificar el funcionamiento de la misma, probando individualmente la implementación de cada historia de usuario en su correspondiente iteración.
- ✓ Todas las pruebas realizadas fueron aceptadas por el cliente cumpliendo entonces el sistema con las historias de usuarios definidas inicialmente en el capítulo 2. Las pruebas se realizaron a través de la interfaz del sistema y se detectaron inconformidades como campos no validados y errores no tratados; todas ellas solventadas rápidamente mediante la corrección de los objetos correspondientes.

## Conclusiones Generales.

Una vez concluida la investigación se puede afirmar que se desarrolló el sistema de gestión de alertas mediante el uso de la tecnología RFID dando cumplimiento a los objetivos trazados en la investigación.

Para ello:

- ✓ Mediante la realización de un estudio sobre la ingeniería de software más acertada para ser utilizada se determinó utilizar XP, esta metodología ayudó a guiar el proceso de diseño de la propuesta del sistema, además de permitir que el trabajo se desarrollara con mayor eficacia.
- ✓ Con el desarrollo del trabajo se cumplieron las tareas de investigación satisfactoriamente, realizando un estudio de la tecnología RFID, así como de sus componentes y las principales características para su uso en el sistema.
- ✓ Durante el desarrollo de los capítulos se hizo un seguimiento de las etapas por las que pasó el software, haciendo uso de la metodología XP. Mediante el uso de la herramienta case Visual Paradigm se generó el diagrama de despliegue y con el uso del NetBeans se desarrolló el sistema en lenguaje Java, empleando las herramientas y tecnologías que se seleccionaron en el capítulo 1 siendo óptimas para la aplicación.
- ✓ Se validaron las funcionalidades implementadas a través de la interfaz del sistema, mediante el diseño de pruebas de aceptación. Las pruebas permitieron encontrar inconformidades en la validación de datos y el tratamiento de errores que contribuyeron a aumentar la calidad del sistema.

De manera general siguiendo la idea a defender planteada en la introducción del trabajo, se realizó el ciclo de vida del software hasta la fase de prueba, tributando a la construcción de un sistema capaz de gestionar alertas mediante el uso de la tecnología RFID como una nueva guía para el seguimiento y control de objetos en diferentes instituciones del país.

## **Recomendaciones.**

Luego de haber implementado la propuesta para el sistema de gestión de las alertas se recomienda:

- ✓ Utilizar este trabajo de diploma como bibliografía para posteriores investigaciones.
- ✓ Continuar con el estudio de nuevas tecnologías para mejorar las potencialidades del sistema.
- ✓ Estudiar las funcionalidades que se puedan añadir al sistema con vistas a lograr un sistema más completo y con mayores niveles de usabilidad para cada uno de los roles que interactúan.

Por último se incita a todos al análisis de este trabajo, para que de esta forma, sirva de material de consulta en la realización de posteriores trabajos relacionados con el tema. Que todos los interesados en el trabajo, una vez analizado y discutido, contribuyeran al perfeccionamiento e integración de nuevas tareas, que posibilitarán una mayor integridad y eficiencia del mismo.

## Referencia Bibliográfica.

1. **Javier I. Portillo García, Ana Belén Bermejo Nieto, Ana M. Bernardos Barbolla.** Tecnología de Identificación por Radio Frecuencia: Aplicaciones en el ámbito de la salud. Madrid: Fundación madri+d para el Conocimiento. [En línea] Diciembre de 2007.
2. **Sánchez, Jorge Alberto Alvarado.** Sistema de Control de Acceso con RFID. México. [En línea] Enero de 2008.
3. **Smith, Roger.** *RFID: A Brief Technology Analysis.* s.l. : CTO Network Library, 2005.
4. **Dargan, Gaurav, y otros.** *The Use of Radio Frequency Identification as a Replacement for Traditional Barcoding.* 2004.
5. **Media, McGraw-Hill Osborne.** *Hacking Exposed Linux: Linux Security Secrets & Solutions (3ª edición).* 2008. pp. 298. ISBN 978-0-07-226257-5..
6. Kimaldi. Área de conocimiento RFID. [En línea]  
[http://www.kimaldi.com/kimaldi/area\\_de\\_conocimiento/rfid/usos\\_y\\_aplicaciones..](http://www.kimaldi.com/kimaldi/area_de_conocimiento/rfid/usos_y_aplicaciones..)
7. Kimaldi Electronics. Kimaldi. [En línea]  
[http://www.kimaldi.com/sectores/geriatricos\\_y\\_centros\\_sanitarios/sistema\\_de\\_localizacion\\_de\\_personal\\_hospitalario\\_mediante\\_rfid\\_activa..](http://www.kimaldi.com/sectores/geriatricos_y_centros_sanitarios/sistema_de_localizacion_de_personal_hospitalario_mediante_rfid_activa..)
8. ATON Managin mobility. [En línea] <http://www.aton.eu/riflessioni/sistema-rfid-para-detectar-fugas-en-estaciones-de-gas/es/..>
9. **Microsoft.** Lenguaje de programación | Microsoft. *Lenguaje de programación | Microsoft.* [En línea] 2013.  
<http://msdn.microsoft.com/es-es/library/xk24xdbe%28VS.80%29.aspx>.
10. **Byous, Jon.** *Java technology: The early years.* s.l. : Sun Developer Network, sin fecha[ca. 1998]., Recuperado 21 de abril de 2005. .
11. **BERTINO, E. A. y MARTINO.** *Sistemas de bases de datos orientadas a objetos.* s.l. : s.l. : Ediciones Díaz de Santos, 1995.
12. **The PostgreSQL Global Development Group .** Sitio oficial de Postgres. [En línea] Copyright © 1996-2013.  
<http://www.postgresql.org/>.

13. **Rod Johnson, Juergen Hoeller, Alef Arendsen, Colin Sampaleanu, Rob Harrop, Thomas Risberg, Darren Davison, Dmitriy Kopylenko, Mark Pollack, Thierry Templier.** *Spring java/j2ee Application Framework 1.2.3.* Reference Documentation : s.n., Copyright (c) 2004-2005.
14. **CHRISTIAN BAUER, GAVIN KING.** *Hibernate in Action.* s.l. : Manning Publications Co. All rights reserved., ©2005.
15. **Jquery, Sitio web oficial de.** [En línea] The jQuery Foundation. Web hosting by Media Temple, Copyright 2013. <http://jquery.com/>.
16. **Maldonado, Daniel.** El CoDiGo K. Qué son los IDE de programación. [En línea] 3 de septiembre de 2007. [Citado el: 24 de noviembre de 2012.] <http://www.webmaster-mexico.com/book/export/html/30...>
17. **Fernández, Cárdenas, et al., et al.** Slidhare BORLAND TOGETHER. *Slidhare BORLAND TOGETHER.* [En línea] <http://www.slideshare.net/unimauro/borland-together-administracion-bases-de-datos..>
18. **Gutiérrez, Javier J.** *Qué es un framework web*[Documento].
19. **Dominguez, Carlos Manuel Gutierrez.** Sistema de Gestión de Información de las Coordinaciones Regionales de Prevención del Delito de la República Bolivariana de Venezuela. Módulos: Administración y Mapas. Habana : s.n., 2011.
20. Temario TIC. [En línea] <http://temariotic.wikidot.com/la-arquitectura-cliente-servidor..>
21. **Pressman, Roger S.** *Ingeniería de software. Un enfoque práctico.* [Documento]
22. **Visconti, Marcello, Hernán y Astudillo.** *Fundamentos de Ingeniería de Software.* [Documento] s.l. : Departamento de Informática Universidad Técnica Federico Santa María.
23. **Canales, Roberto.** Adictos al trabajo. [En línea] 1 de Enero de 2006. [Citado el: 15 de Abril de 2013.] <http://www.adictosaltrabajo.com/tutoriales/pdfs/grasp.pdf>.
24. pruebasdesoftware. [En línea] 2005. <http://www.pruebasdesoftware.com/pruebadeacceptacion.htm>.
25. **Esteves, M.C. Gabino, Ochoa, Ing. Eduardo.** Coordinación de innovación educativa. *Coordinación de innovación educativa.* [En línea] [Citado el: 29 de Abril de 2013.] <http://dieumsnh.qfb.umich.mx/gesinfo/>.
26. **Cornella, Alfons.** Libros electrónicos. *Libros electrónicos.* [En línea] 2002. [Citado el: 02 de Mayo de 2013.] <http://www.google.com/books.84-234-1957-6>.

## Anexos.

### Anexo 1: Historias de usuario.

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre de Historia de Usuario:</b> Crear usuario
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite adicionar los usuarios que tendrán acceso al sistema	
<b>Observaciones</b>	
<b>Prototipo de interfaz.</b>	
<p>El prototipo de interfaz es una ventana modal titulada 'Adicionar Usuario'. Incluye un ícono de usuario con un signo más. Los campos de entrada son: 'Usuario' (valor: usuario), 'Nombre' (valor: Nombre), 'Contraseña' (valor: Ejemplo: Xbdth90EO), 'Confirmar' (valor: Ejemplo: Xbdth90EO), 'Rol' (menú desplegable con 'Seleccionar'), y 'Habilitado' (menú desplegable con 'Seleccionar'). En la parte inferior hay tres botones: 'Cerrar', 'Limpiar' y 'Adicionar'.</p>	

Tabla 2.7: Historia de Usuario Crear Usuario.

Historia de Usuario	
<b>Número:</b> 7	<b>Nombre de Historia de Usuario:</b> Modificar usuario
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite modificar la información asociada a un usuario que exista en la lista	
<b>Observaciones</b>	
<b>Prototipo de interface</b>	
 The image shows a software prototype for a 'Modificar Usuario' (Modify User) dialog box. The dialog has a title bar with the text 'Modificar Usuario' and a close button. On the left side, there is a circular icon of a person. The main area contains several form fields: 'Usuario' (a dropdown menu with 'Seleccionar' selected), 'Nombre' (a text input field), 'Contraseña' (a text input field with the placeholder 'Ejemplo: X8d!f90EO'), 'Confirmar' (a text input field with the placeholder 'Ejemplo: X8d!f90EO'), 'Rol' (a dropdown menu with 'Seleccionar' selected), and 'Habilitado' (a dropdown menu with 'Seleccionar' selected). At the bottom, there are three buttons: 'Cerrar' (Close), 'Limpiar' (Clear), and 'Modificar' (Modify).	

Tabla 2.8: Historia de Usuario Modificar Usuario.

Historia de Usuario	
<b>Número:</b> 8	<b>Nombre de Historia de Usuario:</b> Desactivar usuario
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite desactivar un usuario de la lista	
<b>Observaciones</b>	
<b>Prototipo de interface</b>	

Tabla 2.9: Historia de Usuario Desactivar Usuario.

Historia de Usuario	
<b>Número:</b> 9	<b>Nombre de Historia de Usuario:</b> Autenticar usuario
<b>Modificación de Historia de Usuario Número:</b> Ninguna	

<b>Usuario:</b> Jorge Oña Fernández	<b>Iteración Asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite a los diferentes usuarios autenticarse en el sistema	
<b>Observaciones:</b> Si los datos son incorrectos, muestra un mensaje de Datos Incorrectos	
<b>Prototipo de interface</b>  <p>The image shows a login form for 'Alert RFID Sistema de Gestión de Alertas'. It includes a logo at the top, followed by labels for 'Usuario' and 'Contraseña'. The 'Usuario' field contains the text 'usuario'. The 'Contraseña' field contains the text 'Ejemplo. X8df!90EO'. Below the password field is a checkbox labeled 'Recordar contraseña' which is currently unchecked. At the bottom of the form is a blue button labeled 'Iniciar Sesión'.</p>	

Tabla 2.10: Historia de Usuario Autenticar Usuario.

Anexo 2: Tareas de Ingeniería.

Tarea de Ingeniería	
<b>Número Tarea:</b> 6	<b>Número Historia de Usuario:</b> 6
<b>Nombre Tarea:</b> Implementar la funcionalidad "Crear usuario".	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 4/6
<b>Fecha Inicio:</b> 11/2/2013	<b>Fecha Fin:</b> 17/2/2013
<b>Programador Responsable:</b> Maylin Oña Bernabeu. Norey Fernández De La Torre.	
<b>Descripción:</b> Se crean los usuarios que tendrán acceso al sistema.	

Tabla 3.7: Tarea de Ingeniería Crear usuario.

Anexo 3: Tarjetas Clase-Responsabilidad-Control.

Clase :ICentroDao	
Responsabilidad	Colaboración
adicionarCentro(centro c)	CentroDaoImpl,centro,centro_usuario, CentroManagerImpl,ICentroManager, AddCentroController
eliminarCentro(centro c)	CentroDaoImpl,centro,centro_usuario, CentroManagerImpl,ICentroManager, AddCentroController
listarCentro()	CentroDaoImpl,centro,centro_usuario, CentroManagerImpl,ICentroManager, AddCentroController

Tabla 3.24: Tarjeta CRC Clase ICentroDao.

Anexo 4: Casos de Pruebas.

<b>Caso de Prueba de Aceptación</b>	
<b>Código caso de prueba:</b>	<b>Nombre HU:</b> Gestionar Animal
<b>Nombre de la persona que realiza la prueba:</b> Maylin Oña Bernabeu.	
<b>Descripción de la prueba:</b> Se debe comprobar que se adicione un animal, así como modificar sus datos y mostrarlos. Si los datos introducidos tienen errores o los campos que son obligatorios se dejan en blanco, el sistema muestra un mensaje con el tipo de información que se requiere para corregir el error. En el caso de estar correcto estos datos, se guardan en la base de datos satisfactoriamente.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>➤ La aplicación no puede tener errores de ningún tipo en su compilación.</li> <li>➤ El usuario que tiene el rol de administrador es quien tiene acceso a esta funcionalidad y al centro al que pertenece.</li> </ul>	
<b>Entrada / Pasos de ejecución:</b>	
<b>“Escenario Adicionar Animal”</b>	
Se selecciona la opción adicionar animal, mostrándose un formulario con los campos correspondientes a este: identificador del animal, sexo, color y tipo, posteriormente se selecciona el botón Adicionar, quedando añadido el nuevo animal.	
<b>“Escenario Modificar Animal”</b>	
Se selecciona la opción Modificar Animal mostrándose un formulario con los campos correspondientes, debe permitirse modificar cualquiera de los datos pertenecientes al animal seleccionado y posteriormente se selecciona el botón Modificar.	
<b>“Escenario Desactivar Animal”</b>	
En el listado de animales, con sus datos creados anteriormente, con la opción clic derecho se selecciona el ícono desactivar. Este desactiva el animal y actualiza los datos, permitiendo volver a la lista.	
<b>“Escenario Mostrar Animal”</b>	
Se selecciona la opción Mostrar Animal, mostrándose una tabla con los datos de los animales existentes en el sistema.	
<b>“Escenario Buscar Animal”</b>	

Se muestra un buscador, el usuario introduce el identificador, sexo, color o tipo del animal que quiere buscar.

**\\\\Resultado esperado:**

**“Escenario Adicionar Animal”**

En el listado de animales se muestra el animal añadido.

**“Escenario Modificar Animal”**

Se modifica correctamente los datos del animal en la base de datos y se muestra en el listado de animales, se validan las entradas de datos en los campos restringidos y obligatorios.

**“Escenario Desactivar Animal”**

Se desactiva el animal seleccionado, se muestra un mensaje de confirmación y después de haberse desactivado, se muestra el listado de los animales.

**“Escenario Mostrar Animal”**

Se muestra un listado con todos los animales.

**“Escenario Buscar Animal”**

El usuario introduce en el buscador un criterio de búsqueda y el sistema muestra el animal existente.

**Evaluación de la Prueba:** Satisfactoria

*Tabla 4.3: Caso de Prueba Gestionar Animal.*