



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 7

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

**“Módulo de Gestión de Notificaciones para el Sistema de Gestión para Ingeniería Clínica  
y Electromedicina”**

**Autores:** Yanisbel Díaz Pérez  
Carlos Ernesto Echemendía Ávila

**Tutores:** Ing. Luismel del Valle Román  
Ing. Dennys Javier Hernández Peña

*La Habana, junio de 2013  
“Año 55 de la Revolución”*



La ciencia y la tecnología, en la sociedad revolucionaria, deben estar al servicio de la liberación permanente de la HUMANIZACIÓN del hombre.

**PAULO FREIRE**

### DATOS DE CONTACTO

**Ing. Luismel del Valle:** Graduado de Ingeniería en Ciencias Informáticas en el año 2011 en la Universidad de la Ciencias Informáticas y actualmente desempeña el rol de desarrollador en el proyecto Electromedicina del Departamento. Sistemas Especializados de la Salud (SES).

Correo electrónico: **ldroman@uci.cu**

**Ing. Dennys Hernández Peña:** Ingeniero Informático graduado en el curso 2004-2005. CUJAE. Profesor Asistente. Ha sido líder del grupo de desarrollo para la gestión de equipos médicos (GDEM) durante cuatro años. Ha impartido las asignaturas de Seguridad Informática, Sistema Operativo e Ingeniería de Software. Fue tribunal de tesis cinco años consecutivos y tutor de más de 20 trabajos de tesis de grado. Actualmente arquitecto del proyecto SIGICEM del Departamento. Sistemas Especializados y profesor adjunto a la Facultad 7.

Correo electrónico: **dhernandezp@uci.cu**

## AGRADECIMIENTOS

**Yanisbel**

- A mi familia y a mis padres por el apoyo incondicional que siempre me han brindado y por confiar en mí.
- A mi primo Heney y Daira que en esta trayectoria siempre me han ayudado cuando siempre lo he necesitado, gracias por estar siempre ahí.
- A Maricela que la considero como mi hermana, siempre me ha brindado su apoyo en los buenos y malos momentos que hemos pasado juntas y por sus sabios consejos que siempre me han ayudado.
- A mis tutores por haber confiado en nosotros para la realización de este trabajo.
- En especial a Luismel que se ha convertido para mí, más que mi tutor, en un amigo aconsejándome siempre y por escuchar mis lamentos, gracias por estar en esos momentos y decir siempre Sí no te preocupes, gracias a ti se ha hecho realidad uno de mis sueños.
- A mis compañeras de aula de apartamento, en especial a yaima, la neny, yeny, lisbel, mairim, ity, mily, yanet, lisandra, nené, lili que siempre han estado ahí preocupándose por mis cosas gracias por hacer que la convivencia de todo este tiempo haya sido una satisfacción.
- A mis compañeros de grupo que me acogieron desde el primer día como una compañera más, en especial a nane que se ha convertido en una excelente compañera y amiga, en estos momentos tan difíciles de estrés y preocupaciones, a edney, adrián, alexis, jj, hansel, nelson, rey, andrés, que siempre me hacen reír con sus ocurrencias.
- A mi compañero de tesis Carlos, por su esfuerzo en la realización de este trabajo gracias.
- A mi niña Sony (laptop) que sin ella este trabajo no hubiera sido posible.
- A mis amigos y vecinos en Ciego de Ávila por preocuparse por mis estudios.

- A la profesora Vel por su ayuda incondicional, ha sido de gran apoyo en este momento tan difícil.

### **Carlos**

- A mi madre, por ser mi amiga y estar para mí en los momentos en que la necesito, por complacerme en todos mis caprichos y principalmente, por quererme como nadie.
- A mis abuelos Sofía y Moisés, por darme una buena educación, por apoyarme siempre que lo he necesitado, por escucharme hablar y por sus acertados consejos.
- A Noel, el marido de mi mamá, por ser un padre para mí en todos los sentidos y por cuidar de mi familia.
- A mi tío Carlos y a Enaibis, por tenerme siempre presente.
- A mi tía Magalys, Tata y Piqui grande, por darme abrigo y techo en momentos difíciles y por quererme de forma sincera.
- A Yanisbel, mi compañera de tesis, pues sin ella no hubiese sido posible realizar este trabajo. A ella le agradezco y le aprecio su esfuerzo y enorme dedicación.
- A todas las personas que de una forma u otra me han ayudado a crecer personal y profesionalmente en estos cinco años (en especial a Yosvany (El colega) por haberme ayudado con las CSS de la aplicación).
- A la escuela de mis sueños, la Universidad de las Ciencias Informáticas, que me ha brindado los mejores años de mi vida, en la que me he formado como hombre y de la que nunca me voy a olvidar.
- A mis compañeros de grupo, con los cuales he compartido cinco años llenos de experiencias inolvidables (en especial a JJ, por ser mi mejor amigo y por haberme perdonado cuando traicioné su confianza).
- A mi hermana Claudia, por su cariño sincero.

- A mi laptop (Princesa) por acompañarme en momentos difíciles y permitirme trabajar con mucha tranquilidad.
- A mi Susy, por sus recibimientos inolvidables, su increíble fidelidad y su ilimitado cariño

## DEDICATORIA

**Yanisbel**

Les dedico este trabajo a mis padres, mi hermano y mi sobrino. A mis padres por haberme dado la vida y estar siempre conmigo en los buenos y malos momentos, por enseñarme que por alcanzar una meta hay que hacer grandes sacrificios, por el ejemplo de entrega, esfuerzo y dedicación que son ellos para mí, son los mejores papás del mundo ya que me complacen en todo, son la luz que me guían en todo momento los amo con todo mi corazón, son mi razón de ser. A mi hermano por sus momentos de alegría que me hace olvidar cualquier preocupación que tenga, por ser mi único hermano te quiero mucho y te adoro aunque nunca te lo he dicho espero ser la hermana que siempre has querido, te quiero mucho. A mi sobrinito roland por ser la alegría que me llena día a día por ser la lucecita que billa en la casa y por sus travesuras que a todos les entretiene y nos llena de satisfacción, que este trabajo le sirva de ejemplo y guía los quiero mucho.

**Carlos**

Le dedico este trabajo a mi Mamita por haber llenado el vacío que constituye la falta de un padre, por siempre confiar en mí, por respetar mis decisiones en todo momento y sobre todo, por darme mucho amor. A mis abuelos por haberme criado y educado como si fuera su propio hijo y por haberme consentido en todo.

## RESUMEN

El Centro de Ingeniería Clínica y Electromedicina (CICEM), ha promovido el desarrollo de una solución informática para llevar el control y mantenimiento de equipos médicos a nivel nacional. Por el alcance y desarrollo del centro, se evidenció la necesidad de crear un sistema que contenga la mayor cantidad de funcionalidades, el cual se nombra Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM) y tiene como objetivo cubrir todo el proceso de gestión tecnológica llevada a cabo por los especialistas del Centro Nacional de Ingeniería Clínica y Electromedicina, garantizando la fiabilidad y actualización de la información referente al equipamiento médico instalado en el país. El sistema permite llevar el control de los equipos médicos, pero no cuenta con un mecanismo para realizar la gestión de notificaciones, por lo que se propuso el desarrollo del Módulo de Gestión de Notificaciones para el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM).

Para el desarrollo del módulo se utiliza Visual Paradigm como herramienta CASE, UML como lenguaje de modelado, el lenguaje de programación PHP y Apache como servidor web se utilizan los frameworks Ext JS y Symfony y el sistema gestor de base de datos MySQL.

Con el desarrollo de este módulo se obtendrán beneficios para los diferentes centros de electromedicina que existen en el país, permitiendo la optimización del trabajo, se espera un aumento en la velocidad y precisión de la gestión de la información, además se podrá llevar a cabo todo el proceso de notificaciones, ocurrido cualquier evento en los diferentes módulos del sistema.

**Palabras claves:** equipos médicos, eventos, gestión de notificaciones.



ÍNDICE

INTRODUCCIÓN ..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 6

    1.1 Términos relacionados con la investigación ..... 6

        1.1.1 Emisor:..... 6

        1.1.2 Receptor:..... 6

        1.1.3 Evento:..... 7

    1.2 Sistemas automatizados existentes en el ámbito internacional y nacional..... 7

        1.2.1 Sistemas internacionales existentes ..... 7

        1.2.2 Sistemas nacionales existentes..... 10

    1.3 Metodología ..... 12

        1.3.1 Proceso Unificado de Desarrollo ..... 12

    1.4 Herramientas y tecnologías utilizadas en el proceso de desarrollo ..... 13

        1.4.1 Lenguaje Unificado de Modelado ..... 13

        1.4.2 Visual Paradigm para UML..... 14

        1.4.3 DB Designer Fork 1.0 ..... 15

        1.4.4 NetBeans ..... 16

        1.4.5 Framework ..... 16

        1.4.6 ExtJS 2.2..... 17

        1.4.7 PHP..... 17

        1.4.8 Sistemas Gestor de Base de Datos ..... 17

        1.4.9 PhpMyAdmin como aplicación cliente para el manejo de la base de datos ..... 18

        1.4.10 Sistema Operativo ..... 18

        1.4.11 Servidor Web Apache 2.2..... 19

        1.4.12 Plataforma LAMP ..... 19

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA ..... 21

    2.1 Modelo de dominio ..... 21

        2.1.1 Definición de los conceptos del modelo de dominio..... 21

2.2	Propuesta de solución .....	22
2.3	Especificación de los requerimientos del software.....	23
2.2.1	Requerimientos Funcionales .....	23
2.2.2	Requerimientos No Funcionales.....	25
2.4	Actores del sistema .....	28
2.3.1	Diagrama de Caso de Uso del sistema.....	29
2.4	Descripción de los casos de uso del sistema .....	30
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....		35
3.1	Modelo de análisis.....	35
3.1.1	Diagramas de clases del análisis.....	36
3.1.2	Diagramas de interacción del análisis.....	37
3.2	Modelo de diseño .....	38
3.2.2	Descripción de las clases del diseño .....	41
3.3	Patrón arquitectónico.....	42
3.4	Patrones de diseño .....	43
3.5	Diagrama de despliegue.....	46
3.6	Modelo de datos.....	47
3.6.1	Descripción de las tablas del modelo de datos .....	48
CAPITULO 4: IMPLEMENTACIÓN .....		54
3.7	Modelo de implementación.....	54
3.7.1	Diagrama de componentes.....	54
3.8	Tratamiento de errores .....	55
3.9	Seguridad.....	56
3.10	Estándares .....	57
CONCLUSIONES .....		60

RECOMENDACIONES .....61

REFERENCIAS BIBLIOGRÁFICAS .....62

BIBLIOGRAFÍA .....66

GLOSARIO DE TÉRMINOS .....72

ANEXOS .....75

## INTRODUCCIÓN

Los medios de comunicación impactan de forma profunda en la sociedad actual y proponen nuevos procesos de interacción informativa para conocer los acontecimientos del mundo. Los seres humanos comienzan a desarrollarse en un nuevo tipo de sociedad, llamada Sociedad de la Información o Sociedad de Conocimiento, que viene a reemplazar a los dos modelos socioeconómicos precedentes: la Sociedad Agraria y la Sociedad Industrial **(1)**.

La forma en que los seres humanos se relacionaban e intercambiaban información cambió con el surgimiento del correo electrónico, la mensajería instantánea de texto, teléfonos celulares y las redes sociales. El uso de las Tecnologías de la Información y las Comunicaciones (TIC) trae consigo el surgimiento de nuevas necesidades y deseos en los usuarios, aspectos que conducen al desarrollo de aplicaciones informáticas de forma acelerada especialmente en los últimos años, las que a su vez crean nuevas posibilidades de intercambio de información.

Las organizaciones y empresas han implementado el uso de sistemas de comunicación con el fin de mejorar la efectividad y eficiencia de sus productos. Un ejemplo de estos lo constituyen las aplicaciones web, caracterizadas por atraer a grupos de usuarios, propiciar la interacción entre las empresas y dichos usuarios, así como las relaciones entre ellos. Las aplicaciones web cuentan con un mecanismo de notificación formal y oficial de noticias, hechos y eventos que están por suceder, conocido como sistemas de notificación, los cuales brindan la posibilidad de coordinar la información de manera rápida y segura entre diferentes áreas, personas u organismos que interactúen. Su objetivo fundamental es mantener informados en todo momento a los clientes de un sitio determinado, de una aplicación u otras entidades informáticas en relación a los eventos que se producen en los mismos; además ayudan a la generación y envío de información a través de una confirmación **(2)**.

El sector de la salud en nuestro país trabaja en aprovechar las potencialidades que le brindan los sistemas de comunicación. En este sector se ha logrado desarrollar diversos productos, algunos de los cuales se centran en facilitar el intercambio entre los especialistas, desarrollando mecanismos que permiten que se cumplan las premisas básicas de la comunicación: que sea rápida, eficaz, educativa, informativa y valorativa **(3)**.

En el desarrollo de aplicaciones informáticas en Cuba, con importantes productos y aplicaciones para la medicina, se destaca la Universidad de las Ciencias Informáticas (UCI). Fomentando una amplia red de centros de desarrollo, donde se producen aplicaciones y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación **(4)**, la Universidad apostó, con la creación del Centro de Informática Médica (CESIM), al desarrollo de productos, servicios y soluciones informáticas para la optimización del trabajo y mejoramiento de la calidad de la atención médica. De esta manera se estaría contribuyendo a la formación integral de profesionales de la salud, permitiendo su posicionamiento en el mercado nacional e internacional. En el departamento Sistemas Especializados en Salud (SES) que pertenece a este centro, se desarrolla el proyecto: Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM), el cual tiene como objetivo desarrollar un sistema informatizado, que permita cubrir todo el proceso de gestión tecnológica llevado a cabo por los especialistas del Centro Nacional de Electromedicina (CNE), garantizando la fiabilidad y actualización de la información referente al equipamiento médico instalado en el país.

El SIGICEM se encuentra distribuido en diferentes niveles y cuenta con un conjunto de especialistas organizados en grupos, según el tipo de tarea que realizan. En este sistema se evidencian retrasos en la comunicación entre los especialistas sobre los acontecimientos que suceden diariamente, trayendo consigo demoras e insuficiencias en la realización de las tareas.

En los diferentes centros de electromedicina ocurren eventos y hechos de vital importancia para los especialistas que laboran en los mismos. De su correcta y oportuna notificación depende en gran medida la eficiencia de los servicios de gestión. Por tal razón, se hace necesario crear un mecanismo que permita a dichos especialistas tener acceso a la información, en forma de eventos, para ejecutar las acciones pertinentes de acuerdo a los datos recibidos, con la mayor inmediatez posible.

Actualmente el SIGICEM no cuenta con un mecanismo automatizado que notifique la ocurrencia de determinados eventos. A continuación se describen cómo pudieran ser algunos de estos.

- El mecanismo existente para el manejo de las órdenes de servicio no es el más eficiente, debido a que el proceso no se realiza de forma automatizada. Esto provoca que las reparaciones y los mantenimientos a los equipos puedan retrasarse, afectando la calidad y rapidez de la prestación de servicio a los pacientes en las unidades de salud a las que pertenecen dichos equipos.

- Al ocurrir un cambio en la definición de un nomenclador en el sistema, los especialistas deben tener conocimiento de este evento instantáneamente ya que puede incurrir en la introducción de errores en el mismo, que en casos extremos pudieran provocar pérdidas económicas de un monto considerable.
- Cuando se ofertan nuevos cursos para los especialistas como parte de su capacitación, los mismos deben ser notificados rápidamente, logrando así que puedan realizar una planificación adecuada de su superación sin interferir con sus actividades cotidianas y garantizando que adquieran los conocimientos que les permitan tener un mayor dominio sobre el equipamiento para brindar un servicio de calidad.
- Cuando se incorporan nuevas piezas al almacén, en muchas ocasiones se producen demoras en informar a los especialistas de la entrada de las mismas, por lo que sería conveniente que se notificara este hecho rápidamente y de esa manera darle solución inmediata a las órdenes de servicio y mantenimientos que podrían encontrarse a la espera de la entrada de dichos componentes.
- No se puede establecer la comunicación a través del sistema a los diferentes niveles en los que se encuentra desplegado, por lo que los especialistas no cuentan con un mecanismo mediante el cual puedan intercambiar opiniones, conferenciar y ventilar diversos sucesos con el fin de llegar a una solución en conjunto.

Teniendo en cuenta los elementos mencionados anteriormente se plantea como **problema a resolver**: ¿Cómo gestionar las notificaciones en el Sistema de Gestión para Ingeniería Clínica y Electromedicina contribuyendo a una mayor inmediatez en la transmisión de la información?

Para dar solución a dicho problema se define como **objeto de estudio**: Proceso de gestión de notificaciones en aplicaciones web, enmarcándose en el **campo de acción**: Gestión de notificaciones en el Sistema de Gestión para Ingeniería Clínica y Electromedicina.

Se propone como **objetivo general**: Desarrollar el Módulo Gestión de Notificaciones para el Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM).

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **tareas de investigación**:

1. Realización del estudio del arte referente al proceso de gestión de notificaciones a nivel internacional y nacional, estableciendo similitudes con la investigación en curso.
2. Realización de la definición de los procesos de SIGICEM con los cuales interactuará el proceso de gestión de notificaciones.
3. Asimilación de la metodología, plataforma, tecnologías, librerías, herramientas y pautas definidas por el Sistema de Gestión para Ingeniería Clínica y Electromedicina en el desarrollo de sus módulos.
4. Establecimiento de una clasificación para las notificaciones que va a emitir el sistema, teniendo en cuenta sus características en relación al tipo de evento que la originó y a quién o quiénes van dirigidas.
5. Diseño de los artefactos de los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño e Implementación para las funcionalidades definidas para el Módulo de Gestión de Notificaciones para el Sistema de Gestión para la Ingeniería Clínica y Electromedicina.
6. Implementación de las funcionalidades identificadas en el proceso.

Una vez desarrollado el Módulo de gestión de notificaciones para el SIGICEM, se obtendrán los siguientes beneficios.

- Los especialistas recibirán una mayor cantidad de información de una manera rápida y eficiente.
- Con el despliegue de la solución propuesta, se garantizará la comunicación de todos los especialistas a través del Sistema de Gestión para Ingeniería Clínica y Electromedicina.
- Proveerá a los especialistas de los diferentes centros de electromedicina, de una herramienta de apoyo que les permitirá la toma de decisiones de una manera rápida y precisa.

El presente documento está estructurado en cuatro capítulos, los cuales se relacionan a continuación.

**Capítulo 1 “Fundamentación Teórica”:** se aborda el estado del arte del tema a tratar a nivel internacional y nacional. Además, se explican y justifican los principales conceptos que se abordan, e incluye una explicación de las técnicas, tecnologías, metodologías, herramientas y software empleados en la investigación para darle solución al problema.

**Capítulo 2 “Características del sistema”:** se realiza una breve descripción del problema a resolver y un estudio de los principales procesos que se llevan a cabo en el CICEM. Se define la relación de requisitos funcionales, no funcionales y los casos de uso del sistema descritos a través de los artefactos definidos.

**Capítulo 3: “Análisis y Diseño del sistema”:** se ofrece una representación de la realización de los casos de uso del sistema a través de diagramas de clases del análisis, del diseño y los de interacción, específicamente los de comunicación, así como los patrones de diseños y arquitectónicos utilizados, además del modelo de despliegue que será utilizado para la puesta en marcha del sistema.

**Capítulo 4: “Implementación”:** se expone una descripción del desarrollo de la solución inicialmente propuesta mediante la representación del modelo de implementación. Se asimilan los estándares y estilos de codificación a utilizar y se presentan algunos de los criterios tomados en cuenta para llevar a cabo el tratamiento de errores y la seguridad en el sistema.



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo está dedicado a realizar un análisis detallado sobre el estado del arte de los sistemas de Gestión de Notificaciones a nivel nacional e internacional. Se abarcan una serie de conceptos, que permiten una mejor comprensión sobre la investigación en cuestión. Se describe una caracterización de las principales metodologías, tecnologías y herramientas definidas por la dirección del proyecto Sistema de gestión para Ingeniería Clínica y Electromedicina (SIGICEM).

### 1.1 Términos relacionados con la investigación

La noción de notificación está estrechamente vinculada con una comunicación o un aviso. El envío de una notificación, tiene como objetivo, dejar asentada determinada resolución o información que se ha tomado o que se tomará en un futuro y que es de vital importancia para esa persona, tener conocimiento de la misma.

En el ámbito de la informática, las notificaciones aparecen relacionadas con alertas que emiten ciertos programas o servicios, para advertir o actualizar a un determinado usuario interesado en algo específico. Con el auge de las tecnologías de la información y las comunicaciones, la utilización de las notificaciones en sistemas informáticos se ha convertido en algo imprescindible para el éxito de una aplicación, por lo cual han surgido diferentes maneras de obtener una notificación desde una aplicación web **(5)**.

**1.1.1 Emisor:** es una persona que enuncia un mensaje en un acto comunicativo. Esto quiere decir que el emisor envía un mensaje al receptor, quien está en condiciones de procesarlo e interpretarlo. Este acto de comunicación es posible cuando tanto el emisor como el receptor comparten un mismo código, un idioma en común, un lenguaje de señas. Un emisor también puede ser un aparato u objeto que codifica un mensaje y lo transmite a través de un cierto canal hacia el receptor. En este sentido, una antena de radiofonía puede ser emisora y transmitir un mensaje a través de ondas **(6)**.

**1.1.2 Receptor:** es aquella persona a quien va dirigida la comunicación. El receptor realiza un proceso inverso al del emisor, ya que descifra e interpreta los signos utilizados; es decir, descodifica el mensaje que recibe. Receptor es todo sujeto con capacidad de recibir (de ahí receptor) e interpretar un mensaje,

sea de la índole que sea este mensaje, y sea cual fuera el código empleado para su materialización y los canales elegidos para su transporte. El emisor y el receptor deben utilizar el mismo código **(7)**.

**1.1.3 Evento:** acontecimiento, suceso, eventualidad, o hecho imprevisto. Es una información que interesa a los usuarios y que reciben de la aplicación de notificaciones **(8)**.

### 1.2 Sistemas automatizados existentes en el ámbito internacional y nacional

Se realizó una investigación de los sistemas existentes a nivel internacional y nacional que facilitarán la gestión de notificaciones en aplicaciones web arrojando como resultados.

#### 1.2.1 Sistemas internacionales existentes

##### ❖ Openbravo

Es una aplicación de código abierto de gestión empresarial del tipo ERP destinada a empresas de pequeño y mediano tamaño. Openbravo es una aplicación con arquitectura cliente/servidor escrita en Java. Se ejecuta sobre Apache y Tomcat y con soporte para bases de datos PostgreSQL y Oracle. Es una aplicación completamente web que ha sido desarrollada siguiendo el patrón Modelo Vista Controlador, lo que facilita el desacoplamiento de las áreas de desarrollo, permitiendo el crecimiento sostenible de la aplicación y una mayor facilidad en el mantenimiento del código.

Openbravo ERP incorpora ingeniosas características a continuación se explican brevemente algunas de estas características **(9)**.

- **Alertas:** se pueden programar notificaciones para avisar al usuario cuando ocurre una determinada condición (por ejemplo cuando el stock de un producto es demasiado bajo, un cliente tiene facturas vencidas sin pagar, etc.).
- **Escalabilidad:** Openbravo ERP puede implantarse en un solo servidor o en un clúster de servidores, prestando servicio a miles de usuarios.

Este sistema tiene como inconveniente que está basado en la plataforma de Java, la que es diferente a la que utiliza el sistema SIGICEM; por lo que hacer una posible integración de ambos sistemas sería engorroso. Además Openbravo utiliza como Gestor de Base de Datos a PostgreSQL u Oracle, los cuales

no coinciden con las exigencias del Sistema Nacional de Salud. Por estas razones se hace imposible la utilización de dicho sistema.

### ❖ Facebook

Facebook es un servicio gratuito que permite conectar a las personas en Internet, a los usuarios registrados en su página web, se pueden gestionar los espacios personales: crear álbumes de fotos, compartir vídeos, escribir notas, crear eventos o compartir el estado de ánimo con otros usuarios de la red. Además, tiene un componente importante de interactividad. Posee una serie de mini aplicaciones disponibles, como por ejemplo juegos que permiten interactuar con otros usuarios. También dispone de una aplicación para escritorio llamada Facebook Notifications donde le notifica a cada usuario de los eventos ocurrido en la página web **(10)**.

### Facebook Notifications

Es un pequeño programa gratuito que permite tener las notificaciones del mismo directamente al escritorio del usuario. Actualmente en caso de que no se tenga abierta la cuenta de Facebook, ni el correo para que notifique que se tiene una nueva notificación, no constituye un problema, gracias a este sistema. Indica cuando se tienen nuevos mensajes, eventos e invitaciones. Además permite actualizar el estado desde la aplicación. Con este programa ya no se tendrá problema para enterarse al momento de las notificaciones que tengan los perfiles en Facebook **(11)**.

Facebook Notify, como también es conocido, posee un guión escrito en Python que se conecta a Facebook y utiliza las notificaciones del escritorio para avisar de cualquier cambio o actividad en tu cuenta.

Actualmente permite notificar los eventos que se relacionan a continuación.

- Algún amigo cambia su estado o su perfil.
- Te envían una petición de amigo, evento o grupo.
- Alguien escribe en el muro de alguno de tus amigos.
- Una foto es etiquetada con el nombre de uno de tus amigos.

Al ejecutarlo, aparecerá un icono en el área de notificación. Haciendo clic sobre él mostrará la opción de conexión utilizando los datos que el usuario le introduzca. Por ahora no hay un paquete instalador, pero es sencillo hacerlo funcionar **(12)**.

### ❖ MySpace

MySpace es un sitio web de interacción social, con sede central en Beverly Hills, California. Se convirtió en la red social más popular en USA en junio del 2006. A lo largo de 2005 y 2008, MySpace ha remodelado muchas de las características de su sitio, tanto en diseño como en función. Una de las primeras funciones en ser remodelada fue la página de inicio del usuario, con características tales como actualizaciones de estado, las aplicaciones, y las suscripciones que se añadieron a fin de competir con Facebook. Una de las aplicaciones con que cuenta MySpace hoy en día es con MySpace Notifier 1.3, es una aplicación de escritorio donde se le notifica al usuario de los eventos que ocurren en la página web **(13)**.

### ❖ Notifier 1.3

MySpace Notifier es una pequeña aplicación que se aloja en la barra de sistema del escritorio y notifica cuando se reciben nuevos mensajes, comentarios y pedidos de amigos. El programa muestra una ventana emergente tipo balón y su icono cambia para cada notificación. Una vez recibida la notificación, se da clic en la ventana de notificación y se trasladará automáticamente a la cuenta del usuario que lo está solicitando. Requiere para su uso, utilizar como navegador el Internet Explorer 4.0 o superior **(14)**.

#### Características

- Notificación de nuevos mensajes, comentarios, etcétera.
- Notificación por ventana emergente.
- Se aloja en la barra de sistema.
- Permite acceder a la cuenta directamente al pinchar sobre la ventana de notificación.
- Tiene una interfaz de fácil uso.

Estas aplicaciones, tanto Facebook Notifications como MySpace Notifier, comprenden algunas de las funcionalidades necesarias para la gestión de notificaciones, pero presenta algunos inconvenientes. Entre estos se pueden relacionar los siguientes

- Los usuarios tienen que estar registrados en la página web de la red social.
- Ambas son aplicaciones de escritorio y en el caso del Facebook Notifications posee un guión escrito en Python, razones suficientes para no ser utilizados, pues no son compatibles con la arquitectura definida por el proyecto Sistema de Ingeniería Clínica y Electromedicina (SIGICEM).

### 1.2.2 Sistemas nacionales existentes

#### ❖ Gestión Universitaria

Sistema de Gestión Universitaria (SGU); producto de software, el cual tiene como objetivo, alinear los procesos sustantivos de la Universidad de las Ciencias Informáticas. El módulo de Notificaciones y Alertas, implementa funciones que se comportan como manejadores de notificaciones y manejadores de alertas los cuales se encargan de organizar las acciones a realizarse después de la ocurrencia de un evento o el análisis de una variable; en ambos casos se arma el mensaje o notificación a enviarse, se buscan las relaciones con los usuarios y se ejecuta el envío de acuerdo al dispositivo registrado. Los dispositivos que hoy se manejan como destino de la información son: dirección de correo electrónico y buzón de notificaciones y alertas del SGU. Fueron identificados eventos que no pueden pasar inadvertidos en el Sistema de Gestión Universitaria (SGU), por citar algunos ejemplos.

- Los estudiantes que son dados de baja de la universidad, deben ser informadas a áreas como: residencia, comedores y seguridad.
- El personal de la universidad que se encuentra de misión en el extranjero y está próximo a vencer su permiso de viaje temporal, deben ser informados a los jefes administrativos.
- El personal que opta por cursar una actividad de postgrado (diplomado, maestría, doctorado), debe ser informado del cambio de estado de su solicitud (aprobada y no aprobada).

Fue utilizado CodeIgniter 1.7.2 como marco de trabajo, Sistema Gestor de Base de Datos (SGBD), PostgreSQL versión 8.4, Scrum y Programación Extrema (XP por sus siglas en inglés) como metodología de desarrollo, Visual Paradigm para UML versión 6.4, Lenguaje de programación, Hypertext Preprocessor (PHP) versión 5.2, Ambiente Integrado de Desarrollo (IDE por sus siglas en inglés), NetBeans versión 6.9.1, entre otras tecnologías **(15)**.

Dentro de las razones por las cuales no es conveniente emplear dicho sistema se destaca el hecho de que las tecnologías que utiliza no son compatibles con las que emplea el Sistema de Gestión para

Ingeniería Clínica y Electromedicina. Sería un trabajo engorroso realizar una integración de ambos sistemas.

### ❖ **Mundicampus**

Desarrollado por la empresa española Mundicampus y el Centro de Estudios de Ingeniería de Sistemas (CEIS) del Instituto Superior Politécnico José Antonio Echeverría (CUJAE). Plataforma de formación a distancia, que no solo permite publicar los contenidos de un curso, sino que pone a disposición del alumno, además de textos, imágenes, simulaciones, animaciones, videos y audio. El alumno puede ser examinado a medida que contesta las interrogantes, sus notificaciones se guardan dentro del sistema, ello le posibilita conocer, entre otros elementos, dónde se hallan sus principales dificultades. La información de la herramienta se agrupa en más de 100 tablas que son gestionadas usando Microsoft Access. La programación del sistema fue realizada en el entorno de desarrollo de software Delphi. El sistema puede ejecutarse en Win9X, Windows NT 4.00, Windows 2000 y Windows XP **(16)**.

Este sistema tiene como inconveniente que es dependiente a plataformas de trabajo ya que solo puede ejecutarse en Win 9X, Windows NT 4.0, Windows 2000 y Windows XP. Además, utiliza Microsoft Access para gestionar su información y su programación está realizada en el entorno de desarrollo de software Delphi. Por estas razones se hace imposible el empleo de dicho sistema.

### ❖ **Sistema de Gestión de Tareas Empresariales**

Sistema de Gestión de Tareas Empresariales (GesPro). Es un paquete para la gestión de proyectos desarrollado por el CDAE - Centro de Consultoría y Desarrollo de Arquitecturas Empresariales de la (UCI) y comercializable desde las empresas comercializadoras asociadas a la Universidad. Producto registrado en el Centro Nacional de Derecho de Autor (CENDA) con No. Registro 1540-2010. Desarrollada sobre el framework Ruby on Rails v2.3.5, Apache 2, TomCat 6.0, Máquina virtual Java 6.0, gestor de bases de datos PostgreSQL 9.01.1, presenta un conjunto de características que se relacionan a continuación.

- Tiene una administración e interfaz web amigable y fácil de usar.
- Es adaptable a varios entornos, procesos, empresas, áreas y proyectos.
- Permite la asignación, seguimiento, control y evaluación de tareas según la estructura y permisos del área.
- Soporta múltiples proyectos o áreas simultáneamente y seguimiento de tiempos.

- En cuanto a gestión de usuarios, el control es basado en roles y permite el auto-registro de los mismos.
- Tiene campos ajustados a cada proyecto y a cada usuario así como a las peticiones (tareas).

Esta herramienta soporta autenticación a LDAP, puede usar cualquier gestor de bases de datos relacional (principalmente MySQL), envía notificaciones vía correo a los usuarios cada vez que se les asigna una tarea o ante cualquier evento relacionado con el proyecto. También ofrece la posibilidad de gestionar noticias, ficheros y documentos al proyecto, así como adjuntos a las tareas creadas. Se pueden definir nuevos tipos de tareas con sus campos personalizados **(17)**.

Dentro de las razones por las cuales no es conveniente emplear dicho sistema, se encuentra el hecho de que no utiliza la misma arquitectura del Sistema de Ingeniería Clínica y Electromedicina, pues de utilizarse sería un proceso muy complicado a la hora de realizar la integración con este sistema.

### 1.3 Metodología

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevas aplicaciones de software. En un proyecto de desarrollo, define quién debe hacer qué, cuándo y cómo hacerlo. Es un proceso, puede seguir uno o varios modelos de ciclo de vida, indica cómo hay que obtener los distintos productos parciales y finales en el desarrollo de un software.

Está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo. Se basan en una combinación de los modelos de procesos genéricos (cascada, incremental y evolutivo) **(18)**.

#### 1.3.1 Proceso Unificado de Desarrollo

Es una metodología de desarrollo de software, que está basada en componentes e interfaces bien definidas, y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Es un proceso que puede adaptarse, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. RUP no es un sistema con pasos

firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización **(19)**.

El ciclo de vida de RUP se caracteriza por ser

- Dirigido por casos de uso: los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.
- Iterativo e Incremental: una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

### **1.4 Herramientas y tecnologías utilizadas en el proceso de desarrollo**

#### **1.4.1 Lenguaje Unificado de Modelado**

UML es un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, se centra en la representación gráfica de un sistema. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

Los elementos de UML se clasifican en estructurales (clases, interfaces, colaboraciones, casos de uso, clases activas, componentes y nodos), de comportamiento (interacciones y máquinas de estado), de



agrupación (paquetes) y de anotación (notas). A su vez, hay cuatro tipos de relaciones: de dependencia, de asociación, de agrupación y de realización. Para construir un plano de software que tenga sentido, lo que se hace es combinar los elementos estructurales con sus respectivas relaciones, según sea el caso, obteniendo como resultado uno de los nueve diagramas que existen en UML: de clases, de objetos, de casos de uso, de secuencia, de colaboración, de estados, de actividades, de componentes y de despliegue.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones

- Visualizar: permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (*workflow*) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware **(20)**.

Un modelo UML está compuesto por tres clases de bloques de construcción

- Elementos: los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: son colecciones de elementos con sus relaciones.

### 1.4.2 Visual Paradigm para UML

Es una herramienta CASE que utiliza el UML. Está orientada a la creación de diseños usando el paradigma de Programación Orientada a Objetos. Es poderosa para visualizar y diseñar elementos de software. Incluye una herramienta llamada Visual Architect que permite la generación de código para el manejo de la base de datos.

Con esta herramienta se puede generar código para los lenguajes PHP, JAVA y C# y para los gestores de base de datos DB2, Informix, SQL Server, MySQL, Oracle y PostgreSQL.

Beneficios de Visual Paradigm para UML

- Persistencia de forma fácil

Los desarrolladores emplean mucho esfuerzo en salvar y cargar objetos entre la memoria y la base de datos lo que hace que el programa sea complicado y difícil de mantener. Simplifica estas tareas mediante la generación de una capa de persistencia entre objeto y modelos de datos.

- Generador de mapeo objeto-relacional sofisticado

La capa de mapeo objeto-relacional que se genera incorpora características como soporte de transacciones, agrupación de conexiones y personalización de sentencias SQL.

- Amplia cobertura para bases de datos

Soporta una amplia gama de base de datos, incluidos Oracle, DB2, Cloudscape / Derby, Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Microsoft SQL Server, PostgreSQL, MySQL y otros.

- Integración con IDE

Es una aplicación independiente, se puede integrar a los principales (IDEs) por sus siglas en inglés: Eclipse/WebSphere®, Borland **(21)**.

### 1.4.3 DB Designer Fork 1.0

DB Designer Fork es un sistema de diseño visual de base de datos que integra el diseño, el modelado, la creación y el mantenimiento de bases de datos, en un simple ambiente de ventanas. Posibilita la generación del esquema de la base de datos definida por el usuario, permite la sincronización del modelo con la base de datos, soporta índices y todos los tipos de campos de MySQL, al igual los que defina el usuario.

Este es capaz de generar código para SQL Server, MySQL, Oracle y PostgreSQL (PGSQL). Emplea Licencia Pública General (GPL) dando al usuario la posibilidad de compartir el software licenciado bajo ella, así como realizar cambios en él. Puede ser ejecutado en distintos sistemas operativos debido a su condición de ser multiplataforma **(22)**.

### 1.4.4 NetBeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans es un producto libre y gratuito sin restricciones de uso. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

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

La integración de NetBeans con Symfony permite desarrollar aplicaciones de forma más sencilla y productiva. En primer lugar, es posible crear nuevos proyectos y aplicaciones directamente desde el IDE. También se pueden ejecutar todas las tareas de Symfony, incluso pasándole argumentos y opciones, visualizando el resultado sin necesidad de utilizar una consola de comandos externa. Además, al editar el archivo de una vista se tiene acceso al autocompletado de variables, incluso de los objetos del núcleo de Symfony. NetBeans también permite saltar de una acción al archivo de su vista asociada y viceversa **(23)**.

### 1.4.5 Framework

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas **(24)**.

### ❖ **Symfony 1.4**

Framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación **(23)**.

#### **1.4.6 ExtJS 2.2**

Framework de presentación JavaScript completamente orientado a objeto (OO), es multiplataforma y hace uso del conjunto de tecnologías AJAX. Es una herramienta que brinda múltiples posibilidades para el trabajo con las validaciones y manejo de errores en el cliente. La personalización de temas de estilos es posible en su utilización y provee el trabajo con una amplia configuración e intenso trabajo con las hojas de estilo CSS. Basa toda su funcionalidad en JavaScript a través de librerías YUI, JQuery, o haciendo uso de la librería nativa, así en tiempo de ejecución, carga y crea todos los objetos HTML a través del uso intenso del Modelo de objetos para la representación de documentos (DOM por sus siglas en inglés). Cuenta con dos licencias una comercial y otra Open Source **(25)**.

#### **1.4.7 PHP**

Hypertext Pre-processores (PHP), es un lenguaje interpretado de alto nivel y ejecutado en lado del servidor. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Este es un lenguaje multiplataforma con capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL. Posee una amplia documentación en su página oficial ([www.php.net](http://www.php.net)). Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Permite las técnicas de Programación Orientada a Objetos. Contiene una biblioteca nativa de funciones sumamente amplia e incluida **(26)**.

#### **1.4.8 Sistemas Gestor de Base de Datos**

Los SGBD están constituidos por un paquete de software cuya función es la gestión del acceso a la base de datos (BD), las operaciones fundamentales son: crear, modificar, eliminar y obtener la estructura

asociada al esquema lógico. El objetivo principal es proporcionar un entorno práctico y eficiente a la hora de almacenar y recuperar la información lo que facilita la gestión de grandes volúmenes de datos.

### ❖ MySQL

MySQL es un gestor de base de datos sencillo de usar y rápido. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo, bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad, hace que MySQL sea altamente conveniente para acceder a bases de datos en Internet **(27)**.

Las características principales de MySQL son

- Es un gestor de base de datos capaz de manejar conjuntos de datos de manera eficiente y cómoda.
- Es una base de datos relacional, la cual se establece una relación para manejar los datos de una forma eficiente y segura.
- Existe una gran cantidad de software que la usa.

### 1.4.9 PhpMyAdmin como aplicación cliente para el manejo de la base de datos

PhpMyAdmin es una herramienta de software libre programada en PHP, destinada a la administración de base de datos MySQL **(28)**.

Características

- Interfaz web intuitiva.
- Multiplataforma.
- Permite la administración de varios servidores.
- Admite la creación de gráficos en PDF de su diseño de base de datos.

### 1.4.10 Sistema Operativo

Linux es un sistema operativo con dos características muy peculiares lo diferencian del resto de sistemas que podemos encontrar en el mercado, la primera, es que es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del código fuente. Linux se distribuye bajo la GNU General Public

License por lo tanto, el código fuente tiene que estar siempre accesible y cualquier modificación o trabajo derivado tiene que tener esta licencia, es una de las más importantes distribuciones de GNU/Linux a nivel mundial. Se basa en la distribución Debian GNU/Linux y concentra su objetivo en la facilidad y libertad de uso, la fluida instalación y los lanzamientos regulares **(29)**.

### 1.4.11 Servidor Web Apache 2.2

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El servidor Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Este actualmente es uno de los servidores más usados en la creación de sitios web **(30)**.

### 1.4.12 Plataforma LAMP

En el mundo del código libre, la plataforma más extendida para el desarrollo es la conocida como LAMP (por Linux, Apache, MySQL y PHP). Complementación que resulta utilizada en gran escala para las soluciones basadas en la web. Para su utilización se ha conformado mediante la unión de Ubuntu 10.04, Apache 2.2, MySQL 5.1 y PHP 5.1 **(31)**.

En este capítulo se han evaluado las tendencias actuales de los sistemas de gestión de notificaciones a nivel internacional y nacional, y se determinó que no son una solución viable para aplicarlos en el SIGICEM, ya que no presentan la arquitectura adecuada para su utilización.

- Otras son dependientes a plataformas de trabajo predeterminadas o presentan características propias para una institución determinada.
- Algunas resultaron de ayuda en la implementación del Módulo Gestión de Notificaciones, pues de ellos se obtuvieron elementos que fueron aplicados en la presente investigación. Un ejemplo de ello es el diseño del panel de notificaciones, que se asemeja al elemento que contiene el chat de

la aplicación Facebook. Por lo que se evidenció la necesidad de desarrollar una solución nacional que facilite la gestión de las notificaciones.

- Para el ambiente de desarrollo se aplican las pautas establecidas por el Centro de Informática Médica CESIM y por la dirección del proyecto SIGICEM para el desarrollo de sus aplicaciones, por ser la más idónea y posibilitar el desarrollo de un sistema flexible.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se describe la propuesta de solución para el módulo Gestión de Notificaciones para el Sistema de Gestión para Ingeniería Clínica y Electromedicina. Para ello se detallan los procesos del dominio mediante diagramas de clases del dominio, se identifican los actores que interactúan con el módulo, se exponen los requisitos funcionales y no funcionales, además se presentan los diagramas de casos de uso del sistema con sus descripciones, permitiendo un mejor entendimiento del sistema.

2.1 Modelo de dominio

El modelo de dominio es una representación visual de los conceptos, objetos y términos importantes del dominio y sus acciones. Mecanismo fundamental para comprender el dominio del problema y para establecer conceptos comunes. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema de software (32).

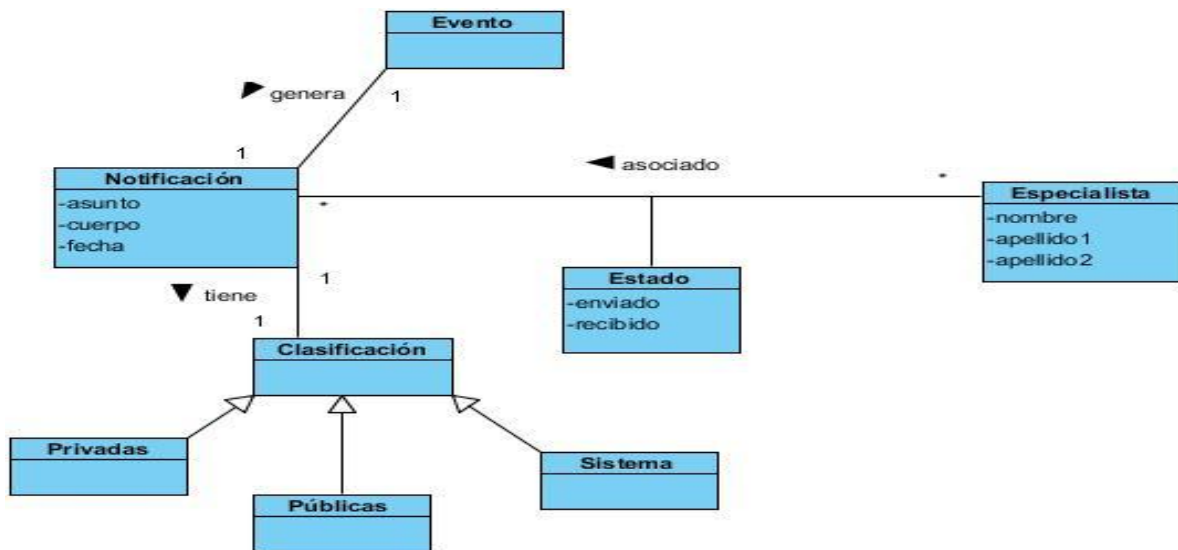


Figura 1: Modelo de dominio.

2.1.1 Definición de los conceptos del modelo de dominio

- Notificación

Notificación representa la información que se genera diariamente, debido a los eventos que suceden en el



Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM).

➤ Evento

Evento representa los diferentes sucesos o acontecimientos que suceden en el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM) y que deben ser notificados.

➤ Clasificación

Clasificación representa los tipos de notificaciones existentes en el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM) como son: Privadas, Públicas y del Sistemas.

- Notificaciones Privadas

Las notificaciones privadas se refieren a las notificaciones que emite un especialista hacia otro especialista solamente.

- Notificaciones Públicas

Las notificaciones públicas se refieren a las notificaciones que emite un especialista hacia un grupo de especialistas.

- Notificaciones del Sistemas

Las notificaciones del sistema se generan debido a los eventos que ocurren en el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM) y que los especialistas deben conocer con la mayor inmediatez.

➤ Especialista

Especialista se refiere a los datos que tienen los especialistas en el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM).

➤ Estado

Estado representa la información tanto enviada como recibida de las notificaciones en el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM) de los especialistas.

### 2.2 Propuesta de solución

Como solución a la situación problemática descrita en el inicio de este trabajo se propone desarrollar un módulo en el SIGICEM, para la gestión de notificaciones, el cual garantizará que los especialistas del

sistema reciban las informaciones con la mayor inmediatez posible. Con dicho módulo interactuarán tres actores del sistema: especialista, administrador y el sistema. Éstos tendrán la posibilidad de gestionar las notificaciones, ocurrido cualquier evento dentro o fuera del sistema, también les ofrece la posibilidad de generar reportes en formato PDF. Se pretende que el sistema sea confiable, el mismo deberá implementar diversas políticas de seguridad para garantizar la confidencialidad, integridad y disponibilidad de la información que se maneje. En aras de lograr este objetivo serán otorgados a cada usuario los derechos que le corresponden, permitiéndoles acceder a los datos a los que tienen permiso, en dependencia del rol.

### **2.3 Especificación de los requerimientos del software**

La especificación de requerimientos describe el comportamiento que se espera del software una vez desarrollado. Debe identificar las necesidades de la institución, así como la interacción con los usuarios funcionales de éste, para recolectar, clasificar, identificar, priorizar y especificar los requerimientos del software, de manera que estos no queden incompletos, o sean ambiguos o contradictorios. Lo anterior, exige a los profesionales dedicados al desarrollo de sistemas informáticos, poner especial atención y cuidado al diseño de la arquitectura, bajo la cual estará soportado el funcionamiento de sus sistemas **(33)**.

#### **2.2.1 Requerimientos Funcionales**

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, indican qué es lo que el software debe hacer, especifican cómo debe comportarse el sistema en situaciones particulares y cómo debe ser el comportamiento de entrada y salida del sistema. A continuación se listan los requerimientos funcionales para la realización del Módulo de Gestión de Notificaciones para el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM) **(34)**

<b>Lista de requisitos funcionales</b>	
<p>RF 1 – Enviar notificación</p> <p>(Información a entrar: fecha_activación, asunto, cuerpo, recibir confirmación de lectura, especialistas )</p> <p>RF 2 –Reenviar notificación</p> <p>( Información a entrar: fecha_activación, asunto, cuerpo, recibir confirmación de lectura, especialistas )</p> <p>RF 3 –Responder notificación</p> <p>( Información a entrar: fecha_activación, asunto, cuerpo, recibir confirmación de lectura)</p> <p>RF 4–Listar notificaciones recibidas</p> <p>(El sistema muestra los datos: asunto, fecha_activación, detalles, remitentes, estado)</p> <p>RF 5–Listar notificaciones enviadas</p> <p>(El sistema muestra los datos: clasificación, asunto, fecha_creación, fecha_activación, cuerpo, editar)</p> <p>RF 6 – Exportar a PDF</p> <p>(Restricción: debe seleccionar la notificación a exportar a pdf)</p>	<p>RF 8 – Eliminar notificación</p> <p>(Restricción: debe seleccionar la notificación a eliminar)</p> <p>RF 9–Visualizar notificaciones por especialistas</p> <p>( Restricción: debe seleccionar el especialista y la fecha en que se desea visualizar sus notificaciones, al igual que se debe seleccionar si se quiere visualizar las notificaciones recibidas o enviadas del especialista)</p> <p>RF 10–Modificar estado de los usuarios dentro del módulo</p> <p>( Restricción: se debe seleccionar el especialista a modificar su estado)</p> <p>RF11 – Visualizar log de notificaciones</p> <p>( Restricción: se debe seleccionar la fecha en que se desea visualizar los log)</p> <p>RF12 – Generar notificación del sistema</p> <p>(El sistema genera estas notificaciones internamente, después de ocurrido un evento y le envía la notificación al especialista)</p> <p>RF13 – Confirmar recibo de notificación</p> <p>(El sistema genera esta confirmación)</p>

<p>RF 7 – Editar notificación</p> <p>(Información a editar: fecha_activación, asunto, cuerpo, especialistas )</p>	<p>internamente, después de ocurrido un evento)</p> <p>RF14 – Generar log de notificación</p> <p>(El sistema genera esta información en la clase classAuxNotificaciones y se guardan en la carpeta sigicem\web\uploads\Logs de notificaciones)</p>
---	--

Tabla 1: Lista de requisitos funcionales.

### 2.2.2 Requerimientos No Funcionales

Un requisito no funcional (NFR) especifica los criterios que se deben usar para juzgar el funcionamiento de un sistema, son a menudo llamados las “cualidades de un sistema”. A continuación se presentan los requisitos funcionales para la realización del Módulo de Gestión de Notificaciones para el Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM) **(35)**.

#### ➤ RNF Usabilidad

La Organización Internacional de Normalización (ISO por sus siglas en inglés) define usabilidad como:

"Grado de eficacia, eficiencia y satisfacción con la que usuarios específicos pueden lograr objetivos específicos, en contextos de uso específicos " **(36)**.

El módulo de notificaciones tiene que ser capaz de ofrecer facilidades de uso para un buen entendimiento y aceptación del producto por los usuarios finales. Debe ser sencillo a la vista de los usuarios, con el objetivo de implementar una herramienta que gane aceptación entre sus usuarios.

#### ➤ RNF Fiabilidad

“Probabilidad de que un sistema se comporte tal y como se espera de él” **(37)**.

- La aplicación tendrá un sistema de logs que registran el flujo constante de los datos y los responsables de sus cambios.
- Los usuarios tendrán acceso solo a los niveles establecidos de acuerdo a la función que realizan,

es decir, según su rol.

### ➤ **RNF Apariencia o interfaz externa**

La aplicación será sencilla, amigable e intuitiva, de fácil navegación por parte del usuario. Estará diseñada para una óptima visualización siendo adaptable a cualquier resolución. La interfaz de la aplicación será similar a la interfaz de la plataforma de Windows para garantizar que el personal que trabaje con el sistema esté lo más familiarizado posible, de manera que agilice y facilite el trabajo con el software. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán tanto en idioma español como en inglés, en dependencia de la opción que seleccione el usuario.

### ➤ **RNF Rendimiento**

El módulo de notificaciones debe ser de rápida respuesta frente a una petición del usuario. Para lograr un buen rendimiento, debe tener un rápido procesamiento de los datos pues los usuarios estarán en constante intercambio de información. Está concebido para un ambiente cliente/servidor, el cual debe garantizar la rapidez de respuesta ante las solicitudes de los usuarios, así como la velocidad de procesamiento de la información.

### ➤ **RNF Soporte**

Una vez puesto en marcha el módulo y usado por los usuarios finales, se recogerá toda la información referente a los defectos, para incorporar las mejoras sugeridas al mismo. La implementación del sistema se regirá por las normas de codificación definidas en el documento de arquitectura de software del Centro de Informática Médica para obtener un producto legible y homogéneo. Se utilizarán como normas para el nombrado la notación CamellCasing para identificar las variables y la PascalCasing para los métodos, lo cual hace el código más legible y facilita el soporte y mantenimiento de la aplicación.

### ➤ **RNF Portabilidad**

El módulo será multiplataforma lo que permitirá ejecutarse sobre diferentes sistemas operativos sin importar sus versiones, y sin necesidad de modificar su código fuente.

### ➤ RNF Seguridad

La seguridad del módulo será responsabilidad del sistema en general (SIGICEM) el cual implementará una jerarquía de accesos para los diferentes usuarios del sistema ya que todos no tendrán los mismos privilegios sobre las posibles acciones a realizar.

### ➤ RNF Confiabilidad

El módulo debe ser confiable y preciso en la información que le suministra al usuario para evitar cualquier tipo de error. Estará disponible todo el tiempo, permitiendo el trabajo a los usuarios y las acciones de mantenimiento. Este debe ser estable, fiable y la velocidad de respuesta debe ser rápida durante la utilización del mismo. La información almacenada debe ser confiable en cuanto a su veracidad e integridad desde su recopilación y durante toda su manipulación.

### ➤ RNF Legales

El módulo estará desarrollado en base a las políticas del software libre, que fueron ajustadas al Sistema Nacional de Salud como: uso de servidores GNU/Linux, gestor de base de datos MySQL, lenguaje del lado del servidor PHP, que son herramientas libres además del uso de Visual Paradigm, que no es libre pero la Universidad de las Ciencias Informáticas posee licencia para su uso.

### ➤ Estándares aplicables

El desarrollo de la aplicación estará basado en Integración de Modelos de Madurez de Capacidades (CMMI por sus siglas en inglés), un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

### ➤ RNF Software

Se requiere del navegador Mozilla 3.6 o superior para la visualización del módulo. Se usará como gestor de bases de datos MySQL Server, ya que da la posibilidad que el sistema sea multiplataforma y que se ajuste a las políticas que plantea el Sistema Nacional de Salud en cuanto al uso del software libre para sus aplicaciones. Para su implementación se usará Symfony 1.4.

### ➤ RNF Hardware

Requerimientos mínimos para el servidor:

- Computadora Pentium a 2.8 GHz o superior.

- 512 MB RAM o superior.
- 40 GB de espacio libre en disco duro como mínimo.

Requerimientos mínimos para la conexión del cliente:

- Computadora Pentium a 333 MHz o superior.
- 128 MB RAM o superior.
- MODEM o red con TCP-IP para conexión al servidor.

### 2.4 Actores del sistema

Representa a un grupo de usuarios del sistema. Un usuario es cualquier ente externo que interactúa con el sistema. No tiene por qué ser un ser humano, puede ser otro sistema informático o unidades organizativas o empresas **(38)**.

Actor	Descripción
Especialista	El especialista es el encargado de gestionar todo lo referente a las notificaciones ya sea crearlas, modificarlas, eliminarlas, visualizarlas, editarlas, responderlas, reenviarlas y listarlas, tanto en notificaciones privadas y públicas.
Sistema	El sistema es el encargado de generar todo lo referente a las notificaciones del sistema, como son: generar notificaciones del sistema, generar recibo de notificaciones y generar logs de las notificaciones.
Administrador	El administrador del sistema es el encargado de controlar y auditar el correcto funcionamiento del módulo de notificaciones.

Tabla 2: Actores del sistema

2.3.1 Diagrama de Caso de Uso del sistema

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema (39).

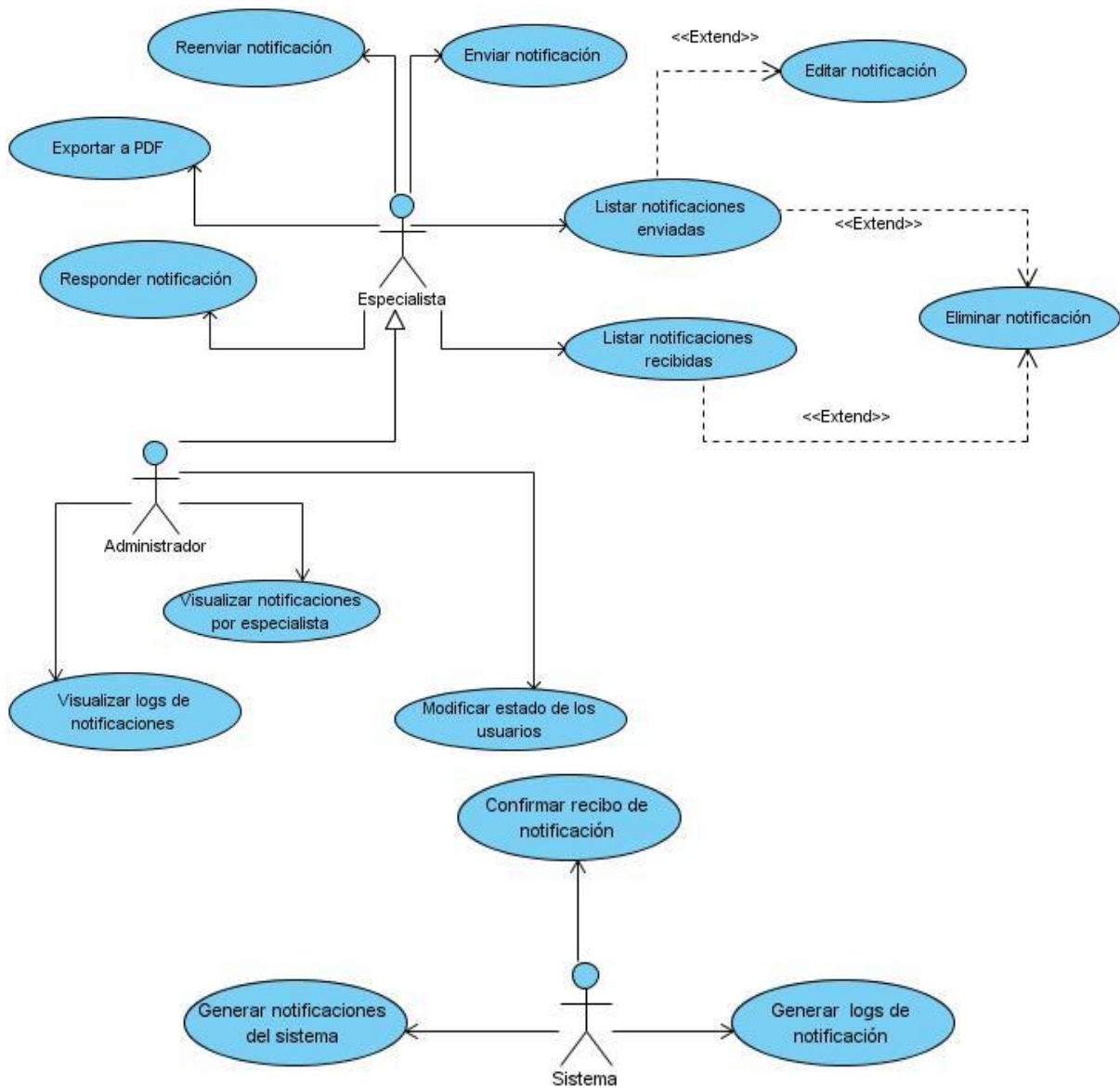


Figura 2: Diagrama de CUS.



2.4 Descripción de los casos de uso del sistema

Tabla 3: CU Enviar notificación

Objetivo	El objetivo de este caso de uso es enviar la notificación correctamente a sus receptores.	
Actores	Especialista	
Resumen	El caso de uso inicia cuando el actor accede a la pestaña de notificaciones que se encuentra en la barra de tareas del sistema SIGICEM, donde se activa el menú de notificaciones y acceden a la opción Enviar notificación. El sistema brinda la posibilidad de introducir y/o seleccionar los datos para enviar la notificación, el actor introduce los datos de la misma, el sistema crea la notificación y la envía, finalizando así el caso de uso.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones		
Postcondiciones	La notificación ha sido enviada.	
Flujo de eventos	Flujo básico Enviar Notificación	
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Enviar notificación.	
2.		<p>Muestra los datos:</p> <ul style="list-style-type: none"> <li>• Fecha activación</li> <li>• Asunto</li> <li>• Cuerpo</li> <li>• Receptores de la notificación</li> </ul> <p>Brinda la posibilidad de introducir los datos de la notificación:</p> <ul style="list-style-type: none"> <li>• Fecha activación</li> </ul>

		<ul style="list-style-type: none"> <li>• Asunto</li> <li>• Cuerpo</li> </ul> Seleccionar: <ul style="list-style-type: none"> <li>• Receptores de la notificación</li> </ul> y permite: <ul style="list-style-type: none"> <li>• Enviar la notificación</li> <li>• Recibir confirmación de lectura</li> <li>• Cerrar operación. Ver Alternativa 1.</li> </ul>
3.	Introduce los datos de la notificación.	
4.		Valida los datos. Si hay datos incompletos, ver Alternativa 2: "Este campo es obligatorio". Si hay datos incorrectos, ver Alternativa 3: "No es un formato válido para ese campo."
5.		Envía la notificación
6.		Muestra el mensaje de información "La notificación ha sido enviada correctamente"
7.		El caso de uso termina.
Flujos alternos		
"Existen campos incorrectos."		
	Actor	Sistema
1.		El sistema señala el campo incorrecto en rojo.
2.		Muestra un indicador sobre los campos incorrectos. "No es un formato válido para ese campo "
"Existen campos vacíos"		
	Actor	Sistema

1.		El sistema señala el campo en blanco en rojo.
2.		Muestra un indicador sobre los campos incorrectos. "Error en el campo"
Flujos alternos		
"Cerrar operación."		
	Actor	Sistema
1.	Selecciona la opción de Cerrar.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 4: CU Listar notificaciones recibidas

Objetivo	El objetivo de este caso de uso es listar todas las notificaciones recibidas.
Actores	Especialista.
Resumen	El caso de uso inicia cuando los actores acceden a la pestaña de notificaciones que está en la barra de tareas del sistema SIGICEM, donde se activa el menú de notificaciones el actor selecciona una de las tres opciones que brinda el menú que son Privadas, Públicas y De Sistema. Cuando se despliegan esas pestañas aparece un mensaje que dice: Ver todas las notificaciones Privadas, Públicas o de Sistema según sea el caso. El sistema Lista las notificaciones recibidas, permitiendo responder, ampliar, reenviar, eliminar, exportar a pdf y seleccionar, el caso de uso termina.

Complejidad	Media	
Prioridad	Secundario	
Precondiciones		
Postcondiciones	Muestra los datos de las notificaciones.	
Flujo de eventos		
Flujo básico Listar notificaciones recibidas		
	Actor	Sistema
1	El caso de uso inicia cuando el actor accede a la opción Listar notificaciones recibidas.	
2		<p>Muestra los datos:</p> <ul style="list-style-type: none"> <li>• Opciones de notificación. Dentro de estas opciones se encuentran las opciones: Responder, Reenviar, Marcar como leída y no leída.</li> <li>• Eliminar.</li> <li>• Exportar a PDF.</li> <li>• Seleccionar.</li> <li>• Asunto.</li> <li>• Detalles.</li> <li>• Remitente.</li> <li>• Estado.</li> </ul> <p>y permite:</p> <ul style="list-style-type: none"> <li>• Cerrar operación. Ver Alternativa 4:</li> </ul>
3.	Selecciona la opción que quiera realizar.	
4.		El caso de uso termina.
Flujos alternos		
"No existen notificaciones del sistema."		

	Actor	Sistema
1		Muestra interfaz vacía.
Flujos alternos		
"Cerrar operación."		
	Actor	Sistema
1.	Selecciona la opción de Cerrar.	
2.		El caso de uso termina.
Relaciones	CU Incluidos	
	CU Extendidos	Eliminar notificación.
Requisitos no funcionales		
Asuntos pendientes		

La elaboración del presente capítulo permitió obtener una mejor comprensión del problema en cuestión, mediante los flujos de trabajo modelamiento del negocio y requerimientos.

- Se identificaron los actores y casos de uso del sistema, los cuales fueron descritos detalladamente.
- Además se construyó el diagrama de casos de uso del sistema, mostrando la relación entre casos de uso y actores, así como la descripción de los mismos.
- También se confeccionó el modelo de dominio representando una guía para los desarrolladores.
- Se definieron los requisitos a cumplir por parte del sistema, tanto funcionales como los no funcionales con sus descripciones. De esta manera se obtuvo una visión entendible de las funcionalidades propuestas para la implementar el Módulo Gestión de Notificaciones.

### CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Teniendo en cuenta las tareas que se deben llevar a cabo para dar por cumplido el presente capítulo, se realizará el análisis y diseño de la solución a partir de la especificación de requisitos previamente efectuada, como los diagramas de clases del diseño, el modelo de datos, el diagrama de despliegue y el modelo de datos, en el mismo se explican sus tablas y sus atributos, obteniéndose el diagrama de clases persistentes para generar a partir del mismo el scripts de la base de datos. Los requisitos serán traducidos a un lenguaje más entendible por el equipo de desarrollo y posteriormente serán diseñados para formar parte del futuro sistema, adecuándose a las características de la arquitectura base definida y sustentado en la utilización de patrones de diseño usados como elemento fundamental. Estos elementos serán descritos en el desarrollo del capítulo.

#### 3.1 Modelo de análisis

Es un modelo conceptual que se encuentra centrado en la abstracción del sistema a desarrollar y posibilita el entendimiento de varios aspectos de la implementación. No tiene en cuenta aquellos elementos relacionados con el diseño pues no toma en consideración las tecnologías a utilizar por el grupo de desarrollo del software; se puede afirmar que es un prototipo de las clases que posteriormente servirán para dar vida al mismo.

Describe las características y comportamiento comunes de un conjunto de cosas que existen en el sistema. Durante su concepción se presentan tres estereotipos conceptuales, que se encargan de identificar a las clases del análisis, estos son **(40)**.

- Interfaz: se encargan de la modelación de toda la interacción que puede existir entre los actores y el sistema. Constituyen las fronteras del sistema.
- Control: representan la coordinación, secuenciación, transacciones y a veces la lógica del negocio. Se emplean a menudo para encapsular el control referido a un Caso de uso (CU).
- Entidad: representa la información de larga duración y a menudo persistente que se maneja en el sistema.

3.1.1 Diagramas de clases del análisis

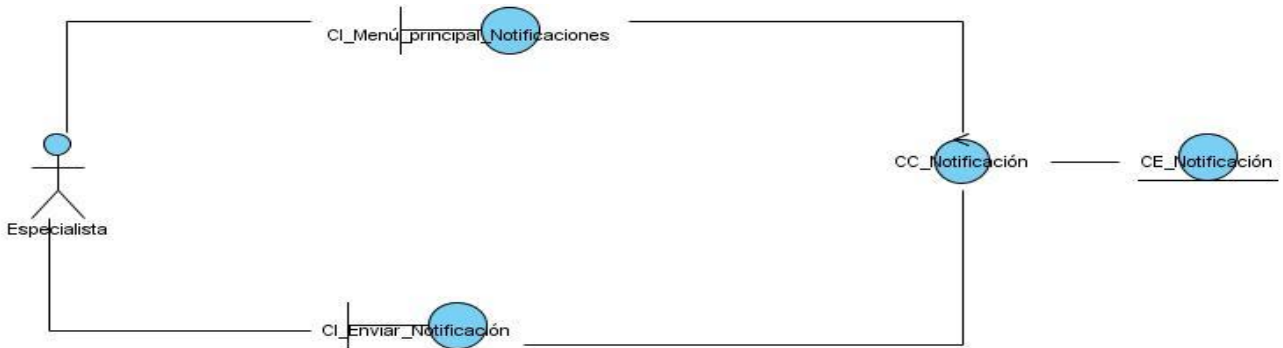


Figura 3: Diagrama de clases del análisis: Enviar notificación

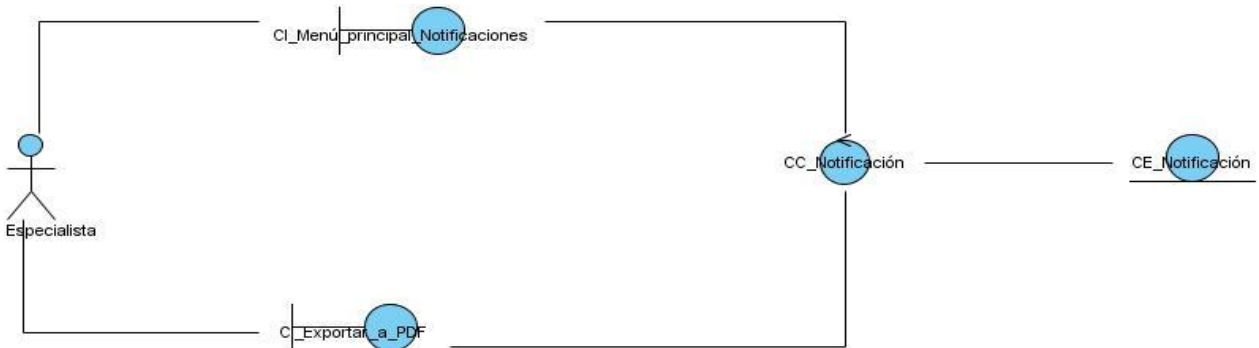


Figura 4: Diagrama de clases del análisis: Exportar a PDF

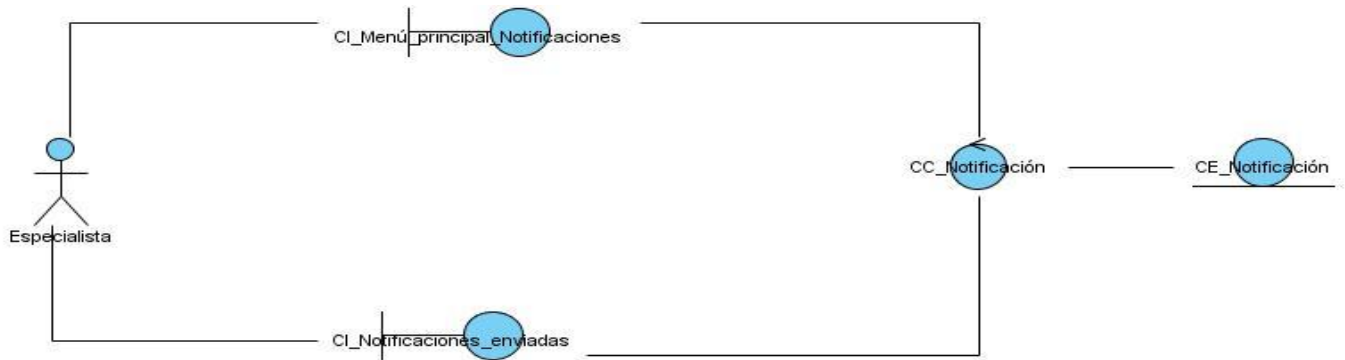


Figura 5: Diagrama de clases del análisis: Listar notificaciones enviadas

3.1.2 Diagramas de interacción del análisis

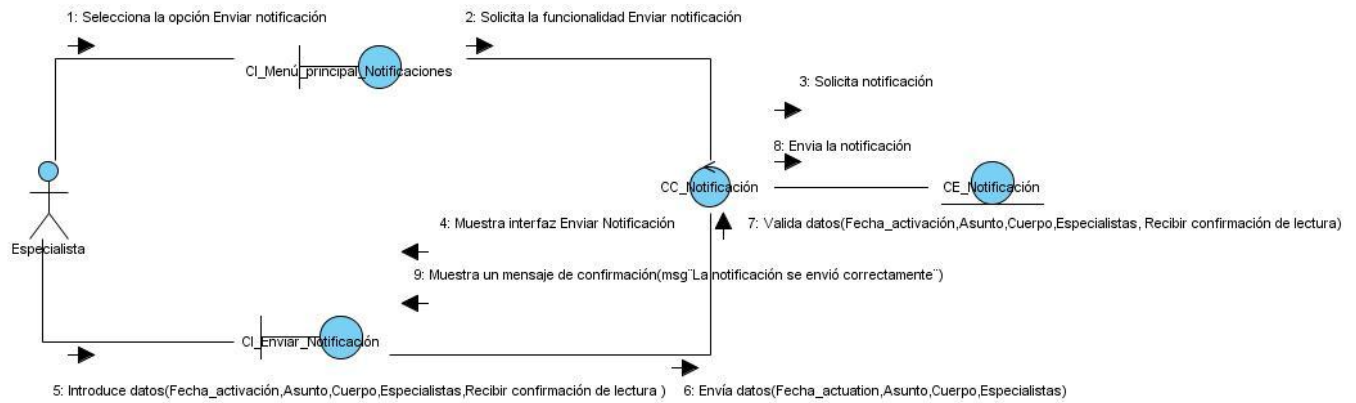


Figura 6: Diagrama de interacción del análisis: Enviar notificación

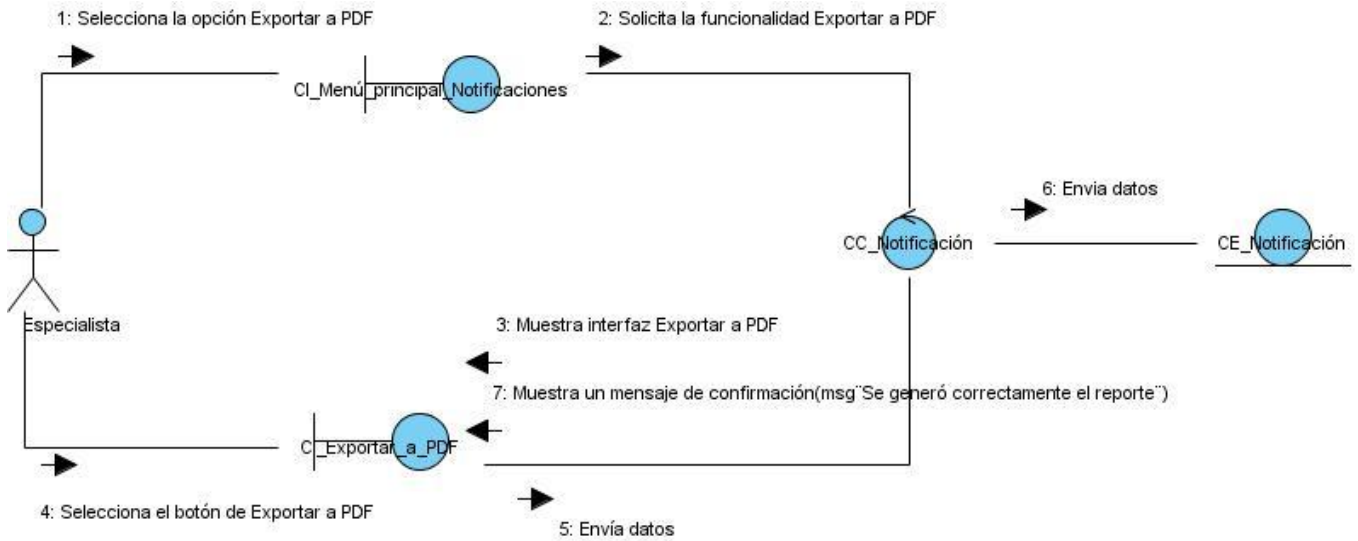


Figura 7: Diagrama de interacción del análisis: Exportar a PDF



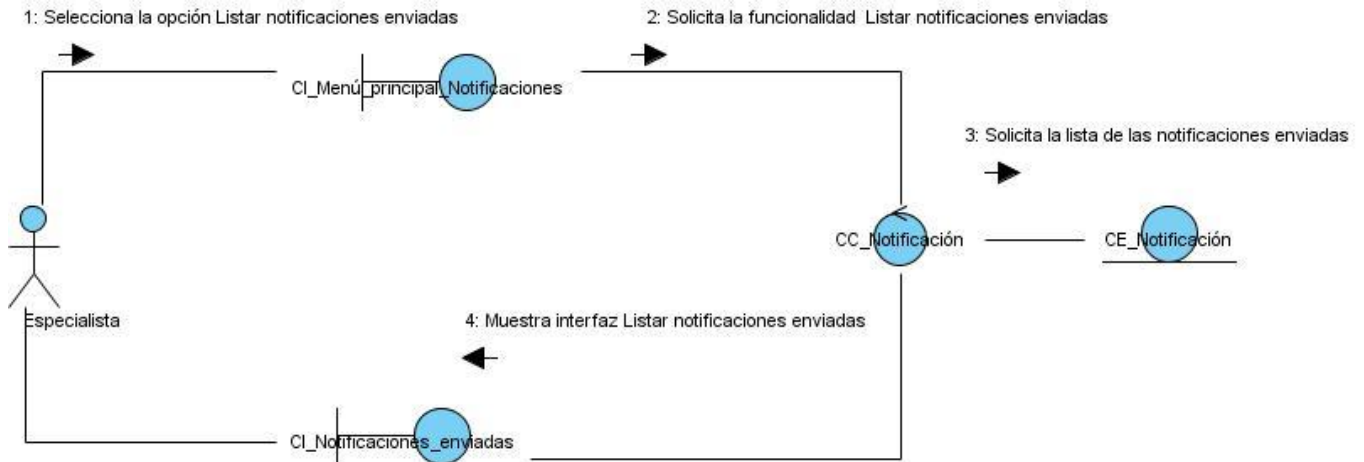


Figura 8: Diagrama de interacción del análisis: Listar notificaciones enviadas

Con la realización de los diagramas de interacción se identificaron las entidades, las interfaces y las entidades controladoras de los casos de uso del Módulo Gestión de Notificaciones. También se obtuvo la lógica de los mensajes para tener un mayor dominio sobre su funcionamiento.

### 3.2 Modelo de diseño

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y para documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto, que abarca todas las clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos. Para la representación de elementos específicos del diseño de aplicaciones web se utilizan los estereotipos web. La extensión UML para web, establece como elementos significativos, tres clases estereotipadas de la siguiente manera (41).

- **Página servidora (Server page):** encargada de la representación de la página que contiene el código que es ejecutado en el servidor. Esta lleva a cabo las acciones correspondientes a construir y generar el resultado HTML, además de realizar peticiones a la capa inferior.
- **Página cliente (Client Page):** presenta el formato XHTML, mezcla datos, presentación y lógica, interpretados por el navegador.
- **Formulario (Form):** elementos de entrada que son parte de una página cliente. Se relacionan directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario.

### 3.2.1 Diagramas de clases del diseño

En busca de concebir un diagrama de clases del diseño con poca complejidad de interpretación y mayor organización; estos fueron estructurados de forma tal que los elementos del diseño se encuentren localizados en la capa a la que pertenece según se define en el patrón Modelo-Vista-Controlador. Los elementos relacionados con el modelo, la vista y el controlador fueron separados. Tomándose en cuenta que, se hace uso de Symfony, es necesario delimitar que elementos del diseño corresponden al framework y a la implementación propia del caso de uso, trabajando de manera transparente al programador (42).

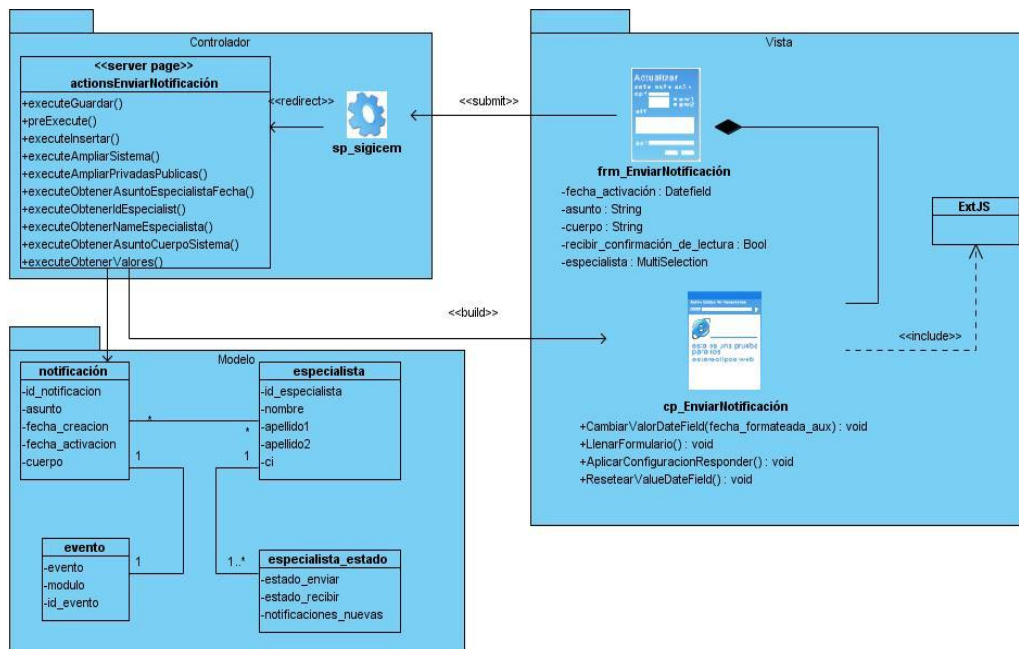


Figura 12: Diagrama de clases del diseño. Caso de uso: Enviar notificación.

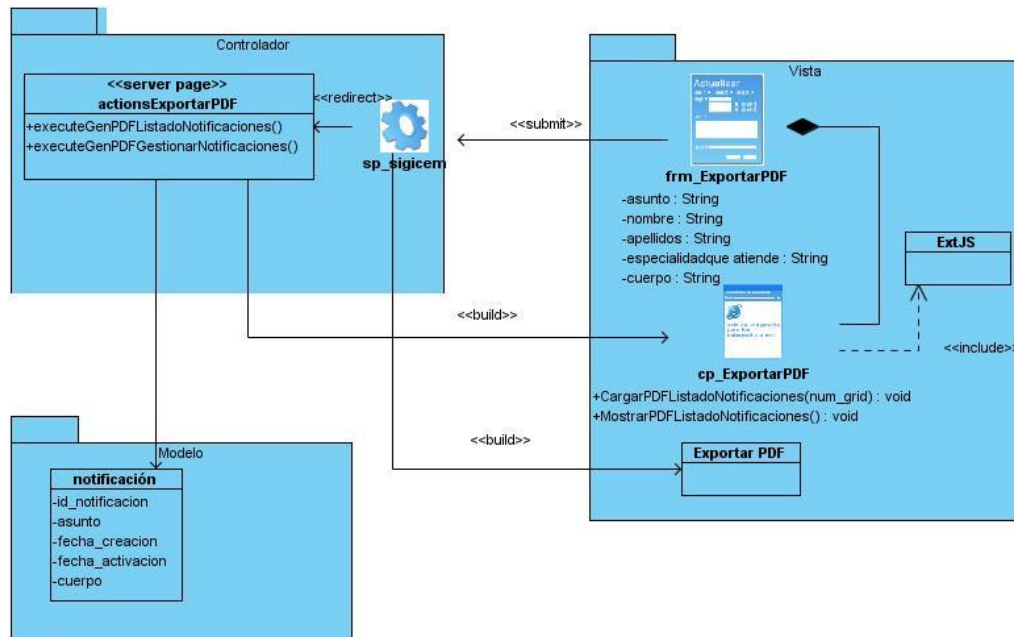


Figura 13: Diagrama de clases del diseño. Caso de uso: Exportar a PDF.

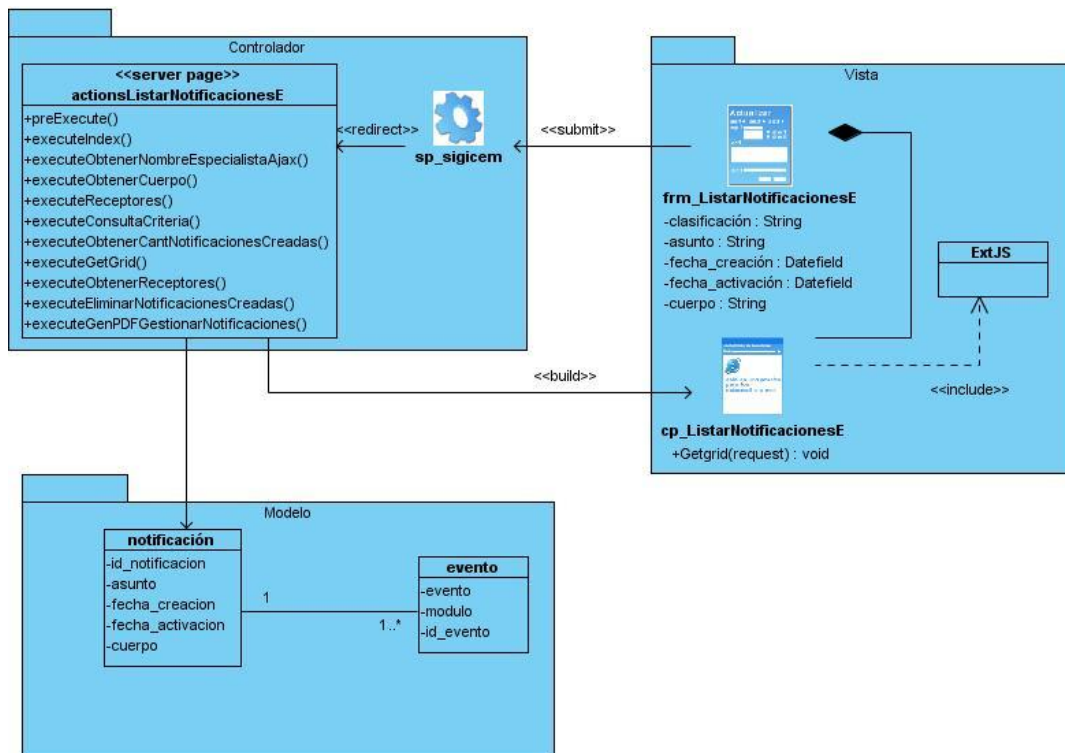


Figura 14: Diagrama de clases del diseño. Caso de uso: Listar notificaciones enviadas.

### 3.2.2 Descripción de las clases del diseño

**Clase:** sigicem

**Descripción:** la página servidora sigicem es el punto de entrada único de toda la aplicación, denominada comúnmente controlador frontal. Todas las peticiones realizadas son gestionadas mediante esta clase. Hace uso del sistema de enrutamiento para realizar la asociación del nombre de una acción y el nombre de un módulo mediante la URL definida por el usuario.

**Clase:** actionsEnviarNotificación

**Descripción:** contiene la lógica de la aplicación del CU Enviar Notificación, la misma se encuentra compuesta por métodos y acciones (métodos especiales) que a su vez utilizan el modelo y define las variables para la vista.

**Clase:** actionsListarNotificacionesE

**Descripción:** contiene la lógica de la aplicación del CU Listar Notificaciones enviadas, la misma se encuentra compuesta por métodos y acciones (métodos especiales) que a su vez utiliza el modelo y define las variables para la vista.

**Clase:** actionsExportarPDF

**Descripción:** contiene la lógica de la aplicación del CU Exportar a PDF, la misma se encuentra compuesta por métodos y acciones (métodos especiales) que a su vez utiliza el modelo y define las variables para la vista.

**Clase:** cp\_EnviarNotificación

**Descripción:** representa el acceso del usuario a la interfaz enviar notificación, esta se encarga de atender todas las selecciones y peticiones que pueda realizar un determinado usuario.

**Clase:** cp\_ListarNotificacionesE

**Descripción:** representa el acceso del usuario a la interfaz listar notificaciones enviadas, esta se encarga de atender todas las selecciones y peticiones que pueda realizar un determinado usuario.

<p><b>Clase:</b> cp_ExportarPDF</p> <p><b>Descripción:</b> representa el acceso del usuario a la interfaz exportar a pdf, esta se encarga de atender todas las selecciones y peticiones que pueda realizar un determinado usuario.</p>
<p><b>Clase:</b> frm_ EnviarNotificación</p> <p><b>Descripción:</b> representa la entrada y salida de datos en el sistema, a través de esta el usuario podrá introducir los datos necesarios para enviar notificaciones.</p>
<p><b>Clase:</b> frm_ ListarNotificacionesE</p> <p><b>Descripción:</b> representa la entrada y salida de datos en el sistema, a través de esta el usuario podrá introducir los datos necesarios para listar notificaciones enviadas.</p>
<p><b>Clase:</b> frm_ ExportarPDF</p> <p><b>Descripción:</b> representa la entrada y salida de datos en el sistema, a través de esta el usuario podrá introducir los datos necesarios para exportar a pdf.</p>

Tabla 5: Descripción de las clases empleadas en los diagramas de clases del diseño.

### 3.3 Patrón arquitectónico

Para la estructuración del diseño se hace uso del patrón Modelo-Vista-Controlador que permite la escalabilidad y flexibilidad del sistema, además de ser el patrón que se utiliza en el proyecto, definido por el personal directivo del Sistema Gestión para Ingeniería Clínica y Electromedicina (SIGICEM). Este patrón separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. La vista se corresponde con las páginas XHTML las cuales son interfaces que el sistema le presenta al usuario, manejan las acciones realizadas sobre la interfaz y recogen la información entrada por este. El controlador se corresponde con las clases controladoras para cada caso de uso, que se encargan del procesamiento de la información en correspondencia con la lógica del negocio en cuestión. El modelo se corresponde con la información manejada en todo el sistema, el cual es una representación orientada a objetos, en forma de clases de entidad, de las tablas de la base de datos del sistema.

Las capas del MVC que implementa Symfony, se encuentran compuestas por siete scripts fundamentalmente, los cuales se distribuyen de la siguiente forma: en el modelo se encuentra la

abstracción de la base de datos y el acceso a los datos; en la vista se podrá encontrar la vista, las plantillas y los layouts correspondientes; mientras que en el controlador se cuenta con el controlador frontal y las acciones **(42)**.

### 3.4 Patrones de diseño

Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos al tratar de lograr ciertos objetivos. Los patrones capturan la experiencia existente y probada para promover buenas prácticas. “Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma” **(43)**.

Durante el transcurso del capítulo 1 de la presente investigación se definió el uso del framework Symfony, este se encuentra concebido de manera tal que puede afirmarse que obliga al uso de varios patrones. Se puede hacer referencia a dos grupos esenciales, los Patrones Generales de Software para Asignar Responsabilidades (GRASP) y Pandilla de los Cuatro (GoF). Dentro de estos se encuentran diversas variantes que pueden ser aplicadas según las necesidades reales que se tengan **(44)**.

#### Patrones GRASP utilizados en el sistema

- Experto: al incluir Propel como ORM, este genera las clases para la gestión de las entidades con las responsabilidades debidamente asignadas. Esta es precisamente la solución que propone el patrón Experto, pues cada una de estas clases cuenta con un conjunto de funcionalidades relacionadas directamente con la entidad que representan.

```

TNotificacionPeer
+getIdNotificacion()
+getIdClasificacion()
+getIdEvento()
+getAsunto()
+getFechaCreacion($format = 'Y-m-d H:i:s')
+getFechaActivacion($format = 'Y-m-d H:i:s')
+getCuerpo()
+setIdNotificacion($v)
+setIdClasificacion($v)
+setIdEvento($v)
+setAsunto($v)
+setFechaCreacion($v)
+setFechaActivacion($v)
+setCuerpo($v)
+hasOnlyDefaultValues()
+delete(PropelPDO $con = null)
+save(PropelPDO $con = null)
+doSave(PropelPDO $con)
    
```

Figura 15: Comportamiento del patrón Experto en el modelado de diseño.

- Creador: en la clase Actions se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase Actions es “creador” de dichas entidades.

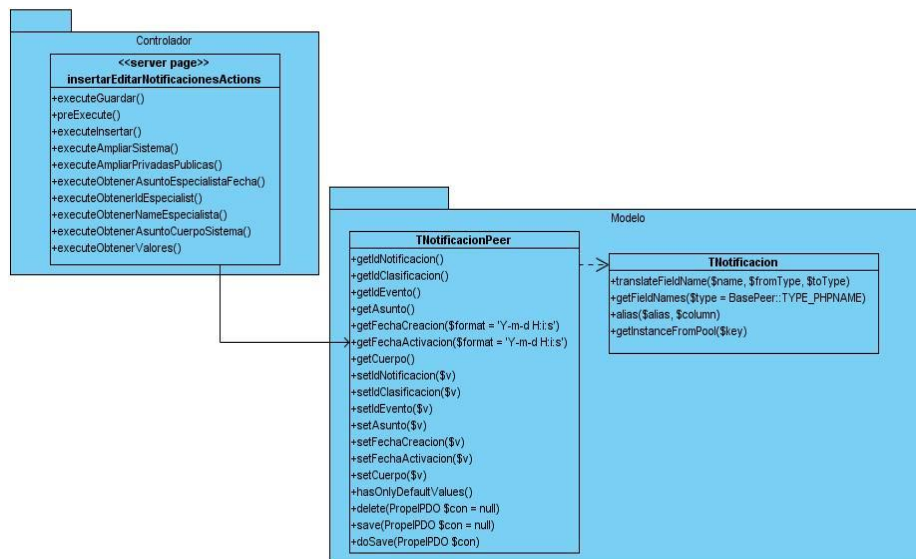


Figura 16: Comportamiento del patrón Creador en el modelado de diseño.

- Alta Cohesión: Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Actions, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties.

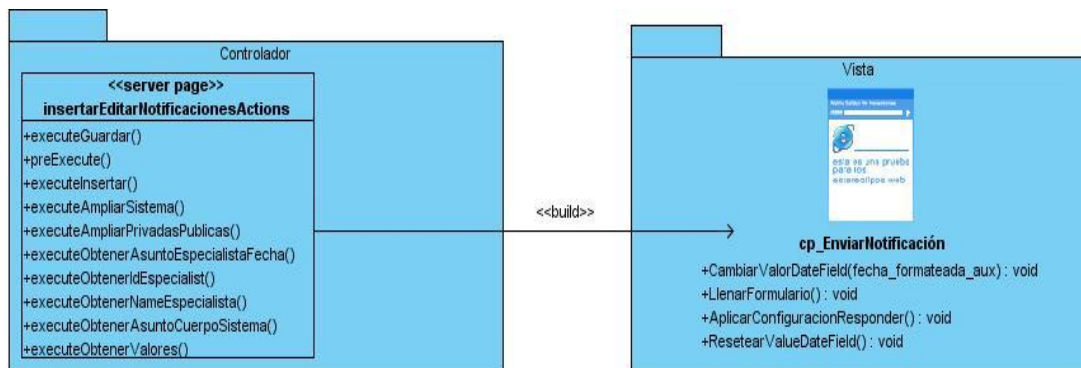


Figura 17: Comportamiento del patrón Alta Cohesión en el modelado de diseño.

- Bajo acoplamiento: en la capa del Modelo se encuentran las clases que implementan la lógica de negocio y de acceso a datos, estas clases tienen pocas asociaciones con otras de la Vista o el Controlador por lo que la dependencia en este caso es baja, poniéndose de manifiesto el patrón Bajo acoplamiento.

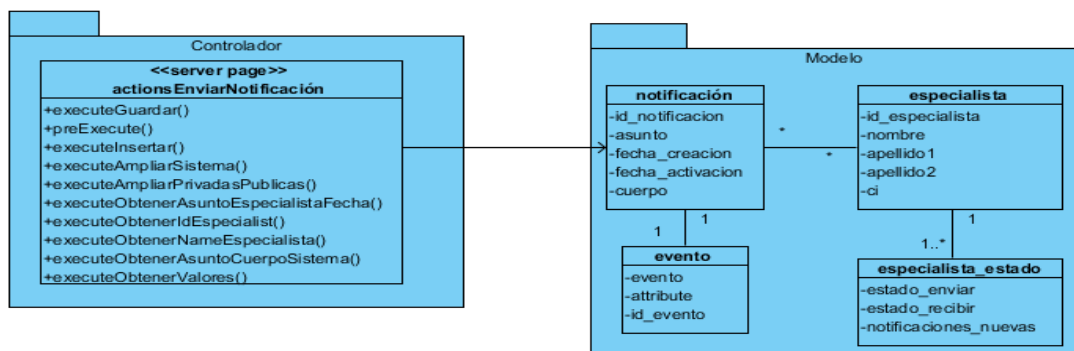


Figura 18: Comportamiento del patrón Bajo acoplamiento en el modelado de diseño.



- **Controlador:** todas las peticiones web son manipuladas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado, es el ejemplo básico que evidencia de forma clara su utilización en Symfony. Este maneja las peticiones del usuario, la seguridad, carga la configuración de la aplicación y otras tareas, es único para cada aplicación. En busca de aminorar un poco la carga que este posee se cuentan con las acciones que contienen las especificaciones de cada página.

insertarEditarNotificacionesActions
+executeGuardar()
+preExecute()
+executeInsertar()
+executeAmpliarSistema()
+executeAmpliarPrivadasPublicas()
+executeObtenerAsuntoEspecialistaFecha()
+executeObtenerIdEspecialista()
+executeObtenerNameEspecialista()
+executeObtenerAsuntoCuerpoSistema()
+executeObtenerValores()

Figura 19: Comportamiento del patrón Controlador en el modelado de diseño.

### Patrones GoF utilizados en el desarrollo del sistema

- **Decorador:** Symfony presenta el denominado archivo layout.php o también conocido como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. Este fichero se complementa con las plantillas, decorándolas y obteniéndose la interfaz final que será mostrada al usuario.

layout
+include_http_metas()
+include_metas()
+include_title()
+include_stylesheets()
+include_javascripts()

Figura 20: Comportamiento del patrón Decorador en el modelado de diseño.

### 3.5 Diagrama de despliegue

El Modelo de Despliegue se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos **(45)**.

- Nodos: elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.
- Dispositivos: nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- Conectores: expresa el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

Para la distribución física del sistema se utilizó un nodo PC\_Cliente que representa la computadora del usuario. La PC\_Cliente también se conecta por el protocolo de Transferencia de Hipertexto (http por sus siglas en inglés) al nodo Servidor Web. Desde el servidor se puede acceder al nodo Servidor de BD mediante el protocolo diseñado para facilitar la reutilización de código de bases de datos (TCP/IP). Además se contará con una impresora conectada mediante el puerto (USB) a la PC\_Cliente para imprimir las notificaciones.

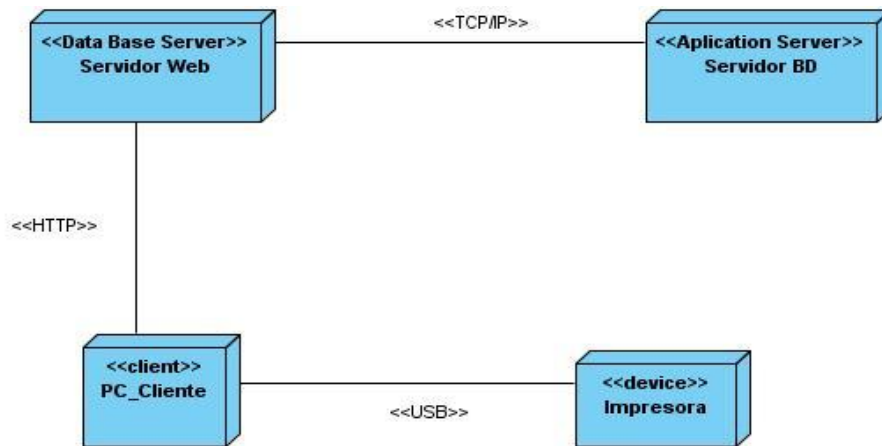


Figura 21: Diagrama de despliegue del Módulo de Notificaciones.

### 3.6 Modelo de datos

El modelo de datos es el artefacto más importante concebido por RUP dentro del proceso que incluye generar la base de datos por esta metodología. Este modelo proporciona una representación visual y física de los datos persistentes del sistema, que en el futuro serán la base de datos. Se obtiene a partir del diagrama de clases persistentes y su forma se expresa mediante un diagrama de UML, siendo sus elementos esenciales del modelo, las entidades, los atributos y las relaciones entre las entidades (46). A continuación se muestra el modelo de datos del Módulo de Gestión de Notificaciones.

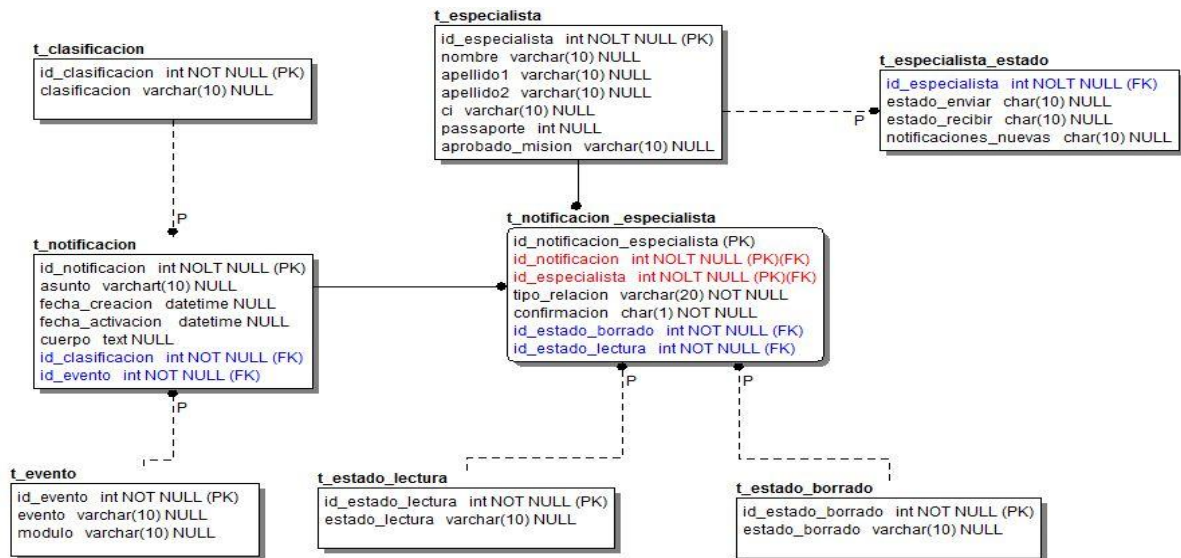


Figura 22: Modelo de datos del Módulo Gestión de Notificaciones.

### 3.6.1 Descripción de las tablas del modelo de datos

A continuación se muestra, para cada una de las entidades que sustentan el Módulo Gestión de Notificaciones, llave primaria (PK por sus siglas en inglés) y llave foránea (FK por sus siglas en inglés), tipo de dato y una breve descripción.

Nombre de la clase: t_clasificacion		
Descripción: tabla que almacena la información de las clasificaciones de las notificaciones.		
Atributo	Tipo	Descripción
Id_clasificacion(PK)	int	En este atributo se almacena el identificador de la clasificación.
clasificacion	varchar	Representa las clasificaciones de las notificaciones que pueden ser (Privadas, Públicas y de Sistema).

<b>Nombre de la clase:</b> t_notificacion		
<b>Descripción:</b> tabla que almacena la información de las notificaciones.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_notificacion(PK)	int	En este atributo se almacena el identificador de las notificaciones.
asunto	varchar	Representa el asunto de la notificación.
Fecha_creacion	datetime	Representa la fecha de creación de la notificación.
Fecha_activacion	datetime	Representa la fecha de activación en que se debe enviar la notificación.
cuerpo	text	Representa la información que contendrá la notificación.
Id_clasificación(FK)	int	Representa la llave foránea que viene de la tabla clasificación.
Id_evento(FK)	int	Representa la llave foránea que viene de la tabla evento.

<b>Nombre de la clase:</b> t_evento		
<b>Descripción:</b> tabla que almacena los eventos ocurridos dentro de todos los módulos de SIGICEM.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

Id_evento(PK)	int	En este atributo se almacena el identificador del evento.
evento	varchar	Representa los eventos ocurridos en el sistema.
modulo	varchar	Representa el módulo donde ocurrió el evento.

<b>Nombre de la clase:</b> t_especialista		
<b>Descripción:</b> tabla que almacena la información de los especialistas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_especialista(PK)	int	En este atributo se almacena el identificador de los especialistas.
nombre	varchar	Representa el nombre de los especialistas.
apellido1	varchar	Representa el primer apellido de los especialistas.
apellido2	varchar	Representa el segundo apellido de los especialistas.
ci	varchar	Representa el carnet de identidad de los

		especialistas.
--	--	----------------

Nombre de la clase: t_especialista_estado		
<b>Descripción:</b> tabla que almacena la información de los especialistas con el estado de las notificaciones.		
Atributo	Tipo	Descripción
Id_especialista(FK)	int	En este atributo se almacena el identificador y es la llave foránea que viene de la tabla especialista.
estado_enviar	char	Representa el estado de envió de la notificación.
estado_recibir	char	Representa el estado de recibo de la notificación.
notificaciones_nuevas	char	Representa si el especialista tiene notificaciones nuevas.

Nombre de la clase: t_notificacion_especialista		
<b>Descripción:</b> tabla que se genera de la relación de mucho a mucho entre las tablas notificación y especialista.		
Atributo	Tipo	Descripción
Id_notificacion_especialista(PK)	int	En este atributo se almacena el

		identificador de la tabla notificación_especialista.
Id_notificacion(FK)	int	Representa el identificador de la tabla notificación.
Id_especialista(FK)	int	Representa el identificador de la tabla especialista.
tipo_relacion	varchar	Representa el tipo de relación de la tabla notificación_especialista que puede ser (remitente o receptor).
confirmacion	char	Representa la confirmación de lectura por 1ra vez de la notificación.
Id_estado_borrado(FK)	int	Representa el identificador de la tabla estado_borrado que viene como llave foránea.
Id_estado_lectura(FK)	int	Representa el identificador de la tabla estado_lectura que viene como llave foránea.

<b>Nombre de la clase:</b> t_estado_borrado		
<b>Descripción:</b> tabla que almacena la información del estado borrado de las notificaciones.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_estado_borrado(PK)	int	En este atributo se almacena el

		identificador de estado_borrado.
estado_borrado	varchar	Representa si las notificaciones fueron borradas.

Tabla 6: Descripción de las tablas del modelo de datos Módulo Gestión de Notificaciones.

En este capítulo se ilustraron los elementos más significativos del modelado del análisis y diseño y del modelo de datos del sistema en desarrollo a partir de los casos de uso especificados en el capítulo # 2.

- Se representaron los diagramas de clases del análisis, diagramas de interacción; especificándose las clases del diseño.
- Se identificaron los patrones de diseño y las características propias de la arquitectura a emplear ya que contribuyen a la reutilización del diseño, identificando aspectos claves que pueden ser aplicados en varias situaciones.
- También se modeló la distribución física del sistema que será empleado en toda la red de centros de electromedicina de todo el país.
- Mediante el diseño de la solución, realizado en el capítulo que recién concluye, donde su principal objetivo fue elaborar el modelo de diseño y modelo de datos el cual es una representación visual y física de los datos persistentes del sistema, se explican sus tablas y sus atributos, obteniéndose el diagrama de clases persistentes para generar a partir del mismo el scripts de la base de datos, queda presentada la propuesta del sistema permitiendo dar paso a la fase posterior dentro del proceso de desarrollo, en este caso: Implementación.



### CAPITULO 4: IMPLEMENTACIÓN

En el presente capítulo se describe lo relacionado con la implementación del sistema de acuerdo a la arquitectura utilizada en la investigación. Además se modela el diagrama de componentes, lo que posibilita que quede conformado el modelo de implementación. También, se explica el tratamiento de errores, la seguridad del sistema, los estándares y estilos a utilizar.

#### **3.7 Modelo de implementación**

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, necesarios para la implantación y despliegue del sistema. Definir la organización del código en términos de subsistemas estructurados en capas. Implementar (codificar y estructurar) clases y objetivos en términos de componentes código fuente, ejecutables, bases de datos **(47)**.

El modelo de implementación del Módulo Gestión de Notificaciones expone una organización en capas, jerarquías de paquetes y subsistemas de implementación. Los mismos contienen componentes y sus relaciones, dividiendo el sistema en partes más manejables. Esto posibilita la reutilización, que se pueda implementar por separado y disminuye el impacto que pueda traer consigo un cambio.

##### **3.7.1 Diagrama de componentes**

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, muestran las opciones de realización incluyendo código fuente, binario y ejecutable **(48)**. A continuación se muestra el diagrama de componentes del Módulo Gestión de Notificaciones del SIGICEM, conformado por los paquetes Vista, Controlador y Modelo.

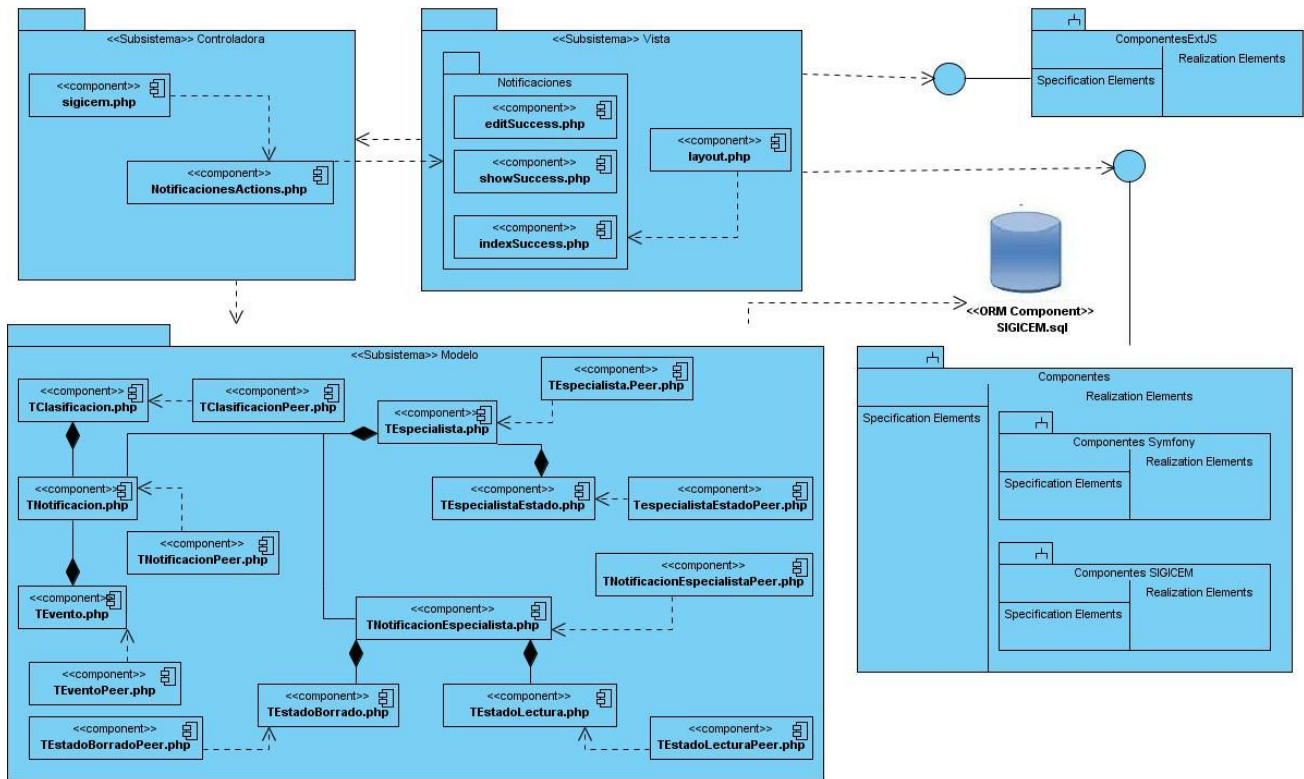


Figura 23: Diagrama de componentes del Módulo Gestión de Notificaciones.

### 3.8 Tratamiento de errores

Un formulario web bien diseñado debe incluir validaciones de los datos proporcionados por el usuario antes de enviar la información al servidor. Para esto se deben realizar en el cliente tantas comprobaciones como sea posible, así se mejoraría el tiempo de respuesta y se reduciría la carga del servidor. Los errores más comunes que se pueden presentar en el funcionamiento del subsistema implementado, están relacionados fundamentalmente con el trabajo de la BD. Para minimizar estos errores se cuenta con cuadros de opciones y menú de selección que facilita la entrada de datos por parte del usuario al subsistema (42). A continuación se visualizan a través de imágenes ejemplos de tratamientos de errores del Módulo Gestión de Notificaciones.



Figura 24: Ejemplo de cuadros de selección utilizados.

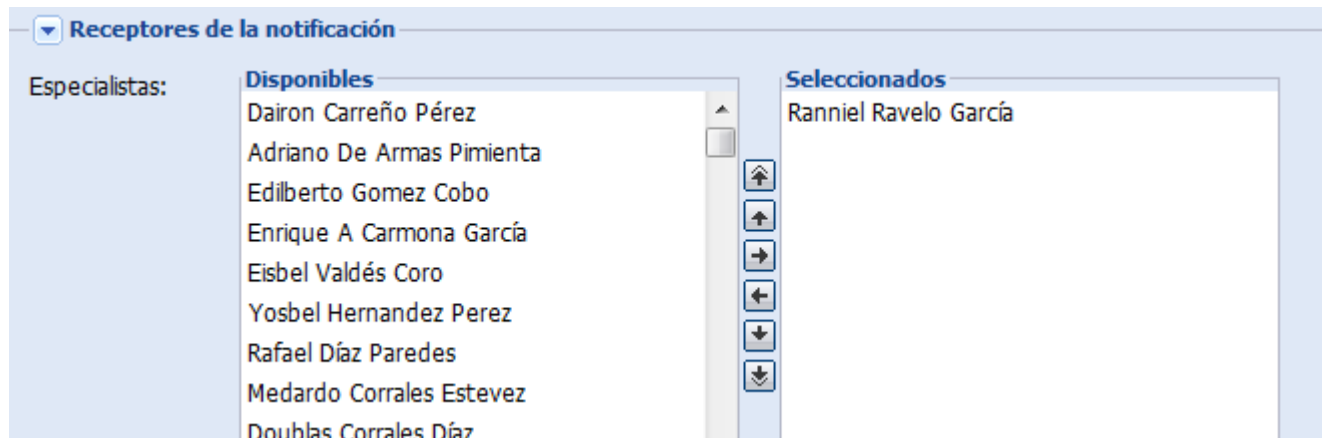


Figura 25: Ejemplo de menú de selección utilizado.

### 3.9 Seguridad

La seguridad informática es un estado de cualquier tipo de información que indica que ese sistema está libre de peligro, daño o riesgo. Se entiende como peligro o daño a todo aquello que pueda afectar su funcionamiento directo a los resultados que se obtienen del mismo **(49)**.

Características de la seguridad informática

- **Integridad:** los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.
- **Confidencialidad:** la información o los activos informáticos son accedidos solo por las personas autorizadas para hacerlo.
- **Disponibilidad:** los activos informáticos son accedidos por las personas autorizadas en el momento requerido.

Para analizar la seguridad de un sistema se debe pensar en la forma en que el mismo pudiera sufrir determinada pérdida o daño, para lo cual es necesario identificar las debilidades del sistema. La seguridad

del módulo Gestión de Notificaciones será responsabilidad del sistema en general (SIGICEM) el cual implementará una jerarquía de accesos para los diferentes usuarios del sistema ya que todos no tendrán los mismos privilegios sobre las posibles acciones a realizar, para ello en el proyecto se utiliza el Plugin sfGuard, ya que posibilita usar en cualquier aplicación la Lista de Control de Acceso (ACL por sus siglas en Inglés) o lo que es lo mismo autorización y autenticación, haciendo uso del modelo usuarios, grupos y permisos. El mismo es desarrollado y soportado en su totalidad por Fabien Potencier, creador de Symfony (50).

En el proyecto se hace uso de la gestión de trazas ya que se registran en la base de datos acciones llevadas a cabo por los usuarios como: inicio o cierre de sesión, acceso a un módulo, modificación de un atributo, entidad o cualquier otra operación sobre el sistema. Otro de los aspectos de la seguridad que se lleva a cabo es la administración de la seguridad ya que el sistema brinda la posibilidad de asignar o denegar permiso a los diferentes roles y usuarios, en el módulo. Todos los permisos son registrados por el sistema.

### 3.10 Estándares

En el campo de la informática se define un estándar como: “Una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad. También significa un modelo o guía que se sigue para realizar un proceso o para no desviarse de un objetivo” (51). De acuerdo a lo planteado anteriormente, y luego de efectuar un estudio de los estilos y estándares definidos por el CESIM, se han identificado aquellos que serán utilizados para la implementación del Módulo Gestión de Notificaciones (42).

Identación	
<b>Inicio y fin de bloque</b>	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque <code>{}</code> . Lo mismo sucede para el caso de las instrucciones <i>if, else, for, while, do while, switch, foreach</i> .
<b>Aspectos Generales</b>	El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; porque este puede variar según la configuración de dicha tecla. Los inicios ( <code>{</code> ) y cierre ( <code>}</code> ) de ámbito deber estar alineados debajo de la declaración a la que

	<p>pertenecen y deben evitarse si hay sólo una instrucción. Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.</p>
<p><b>Comentarios, separadores, líneas, espacios en blanco y márgenes.</b></p>	
<p><b>Ubicación de comentarios</b></p>	<p>Al inicio de cada clase o función y al final de cada bloque de código.</p> <p>Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma.</p>
<p><b>Líneas en blanco</b></p>	<p>Se emplean antes y después de métodos, clases y estructuras. Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.</p>
<p><b>Espacios en blanco</b></p>	<p>Entre operadores lógicos y aritméticos. Se recomienda usar espacios en blanco entre estos operadores para una mayor legibilidad en el código. Ejemplo: producto = nomproducto</p>
<p><b>Aspectos generales</b></p>	<p>Sobre el comentario.</p> <p>Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.</p> <p>Sobre los espacios en blanco.</p> <p>No se debe usar espacio en blanco: Después del corchete abierto y antes del cerrado de un arreglo. Después del paréntesis abierto y antes del cerrado. Antes de un punto y coma.</p>
<p><b>Variables y constantes</b></p>	
<p><b>Aspectos generales</b></p>	<p>El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.</p>
<p><b>Clases y Objeto</b></p>	

<b>Apariencia de clases y objetos</b>	Primera letra en mayúscula. Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing. Ejemplo: MiClase ().
<b>Apariencia de atributos</b>	Primera letra en minúscula. El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamellCasing.
<b>Apariencia de las funciones</b>	Apariencia de las funciones. Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Se empleará notación PascalCasing. Ejemplo: function BuscarUnidad (). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set.
<b>Aspectos generales</b>	El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con solo leerlo se conozca el propósito de los mismos.

Tabla 7: Estándares y estilos de codificación.

En la elaboración del presente capítulo se mostraron los resultados obtenidos durante la etapa de implementación.

- Se confeccionó el diagrama de componentes mostrando la distribución de todos los paquetes para su implementación y se detallaron además las organizaciones y dependencias lógicas existentes entre dichos componentes.
- Fueron identificados los estándares y estilos de codificación a utilizar para la implementación de los componentes definidos y las opciones a efectuar para realizar el tratamiento de errores.

### CONCLUSIONES

Luego de la investigación realizada se puede concluir que se desarrollaron todas las tareas a fin de cumplir los objetivos propuestos:

1. La realización de un estudio previo de sistemas utilizados para la gestión de notificaciones a nivel internacional y nacional, evidenció que los mismos no ofrecen una solución viable al problema planteado en la investigación, por lo que se ratifica el desarrollo del Módulo Gestión de Notificaciones para el Sistema de Gestión para Ingeniería Clínica y Electromedicina.
2. Las tecnologías y herramientas investigadas ofrecen ventajas para la implementación del módulo ya que se integran fácilmente entre sí y garantizan una solución estable y con calidad, cumpliendo con los estándares del desarrollo de software.
3. Mediante la realización de los artefactos pertenecientes a las disciplinas de trabajo definidas y guiados por la metodología del Proceso Unificado de Desarrollo, se logró establecer la documentación necesaria de la investigación, sirviendo como guía al desarrollador para implementar la solución obtenida.
4. La integración del módulo Gestión de Notificaciones del Sistema de Gestión para Ingeniería Clínica y Electromedicina representa una fortaleza para el mismo, además que constituye una característica común en aplicaciones de este dominio, que el sistema no tenía anteriormente.

### RECOMENDACIONES

Al concluir el presente trabajo de diploma, considerando cumplidos los objetivos trazados en el mismo, se recomienda:

1. Realizar un análisis más profundo del negocio que se realiza en los diferentes centros de electromedicina, con el objetivo de identificar un mayor número de eventos que determinen la creación de notificaciones, y así garantizar aún más la inmediatez y calidad en la transmisión de información.
2. Lograr un mecanismo más óptimo a la hora de gestionar la creación de nuevas notificaciones, a partir de eventos que el sistema genera, utilizando funcionalidades y mecanismos que provee el framework Symfony, e integrado con otras tecnologías que permiten una mayor aproximación al envío de información en tiempo real.
3. Investigar y agregar al módulo las tecnologías necesarias para garantizar una mayor persistencia en los datos, garantizando en la medida de lo posible el trabajo sin conexión y logrando una rápida actualización cuando se restablezca la misma.



### REFERENCIAS BIBLIOGRÁFICAS

1. Sosa Vázquez, Yailis y Ruiz Cardoso, Alberto. Análisis y diseño de los procesos de Agregación, Notificaciones y Ajustes al Plan del subsistema de Planificación perteneciente al Sistema Integral de Gestión de Entidades Cedrux. La Habana : Universidad de las Ciencias Informáticas, 2010. pág. 1, Trabajo de Diploma.
2. Pérez Barroso, Javier. Sistema de notificación de eventos. Universidad de las Ciencias Informáticas. 2011. pág. 1, Trabajo de Diploma.
3. Una pedagogía de la comunicación. López Viera, Luís and Kaplún, Mario. Ciudad de La Habana : Caminos.
4. UCI Universidad de las Ciencias Informáticas. [En línea] [Citado el: 5 de junio de 2013.] 2012 Universidad de las Ciencias Informáticas, La Habana, Cuba. Todos los derechos reservados. . <http://www.uci.cu/mision>.
5. Lara Pérez, Yaneilis. Implementación y pruebas de un sistema de RHODA 2.1. Universidad de las Ciencias Informáticas. La Habana : s.n., 2011. pág. 11, Trabajo de diploma.
6. THE FREE DICTIONARY BY FARLEX . [En línea] [Citado el: 10 de noviembre de 2012.] <http://es.thefreedictionary.com/Emisor>.
7. THE FREE DICTIONARY BY FARLEX. [En línea] [Citado el: 10 de noviembre de 2012.] [www.retoricas.com](http://www.retoricas.com) - Todos los derechos reservados - Correo: [admin@retoricas.com](mailto:admin@retoricas.com). <http://es.thefreedictionary.com/Receptor>.
8. THE FREE DICTIONARY BY FARLEX. [En línea] [Citado el: 11 de noviembre de 2012.] <http://es.thefreedictionary.com/evento>.
9. Serrano, Nicolas and Ciordia, Ismael. Openbravo. Openbravo. [En línea] agosto de 2001. [Citado el: 1 de noviembre de 2012.] [www.openbravo.com](http://www.openbravo.com).
10. La red social Facebook. [En línea] marzo de 2010. [Citado el: 10 de noviembre de 2012.] <http://www.aulaclie.es/articulos/facebook.html>. Artículo24.
11. WEBADICTOS.COM Notificaciones Facebook en tu escritorio. [En línea] [Citado el: 13 de enero de 2013.] <http://www.webadictos.com.mx/2013/01/13/notificaciones-facebook-en-tu-escritorio>.

12. [En línea] [Citado el: 13 de enero de 2013.] <http://www.genbeta.com/redes-sociales/facebook-notify-notificaciones-de-facebook-integradas-en-gnome>.
13. [En línea] [Citado el: 13 de noviembre de 2012.] [http://myspace-notifier.uptodown.com/..](http://myspace-notifier.uptodown.com/)
14. ARCHIVOSPC.COM. [En línea] [Citado el: 13 de noviembre de 2012.] <<http://myspace-notifier.archivospc.com>.
15. Pérez Santiesteban, Diana Rosa, et al. Módulo de Notificaciones y Alertas del Sistema de Gestión Universitaria. Universidad de las Ciencias Informáticas. La Habana : s.n., 2012. Ponencia. [uciencia-2012-t46-p1270-ponencia-1476.pdf](http://www.uciencia-2012-t46-p1270-ponencia-1476.pdf).
16. Revista Científico- Técnica Popular de la Juventud Cubana. Sanz, Lucía C. 310, La Habana : Casa Editorial Abril, enero- febrero de 2003. Publicación bimestral.
17. CDAE - Centro de Consultoría y Desarrollo de Arquitecturas Empresariales. [En línea] [Citado el: 1 de diciembre de 2012.] <http://gespro.cdae.prod.uci.cu>.
18. Mendoza Sanchez, María A. informatizate. Metodologías De Desarrollo De Software. [En línea] 7 de junio de 2007. [Citado el: 4 de junio de 2013.] Ing. Informático - UNT, Microsoft Certified Professional - MCP,Analísta y Desarrolladora - TeamSoft Perú S.A.C. . [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).
19. [En línea] [Citado el: 1 de diciembre de 2012.] <http://es.scribd.com/doc/12983329/Metodologia-de-Desarrollo-de-Software>.
20. [En línea] [Citado el: 5 de diciembre de 2012.] <http://resumenuml.bloggratis.es>.
21. Pressman, Roger S. Organization, V. P. Sitio Web oficial Visual-Paradigm. [En línea] 2009. [Citado el: 4 de junio de 2013.] S.I.:McGraw-Hill Companies. ISBN 8448132149.. <http://www.visual-paradigm.com/product/vpuml/>.
22. DB Designer Fork. [En línea] 2009. [Citado el: 19 de mayo de 2013.] <http://dbdesigner-fork.sourceforge.net/>.
23. [En línea] [Citado el: 10 de diciembre de 2012.] <http://www.symfony.es/noticias/2009/10/05/netbeans-ya-incluye-soporte-para-symfony>.
24. Potencier, Fabien y Zaninotto, François. Symfony 1.2, la guía definitiva. 2008.
25. Desarrollo en web. [En línea] 2008. [Citado el: 19 de mayo de 2013.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo..>

26. Programacionweb.net. [En línea] 2009. [Citado el: 19 de mayo de 2013.] <http://www.programacionweb.net/articulos/articulo/?num=686..>
27. [En línea] [Citado el: 10 de diciembre de 2012.] <http://dev.mysql.com/doc/refman/5.0/es/index.html>.
28. Phpmysqladmin.net. [En línea] 2010. [Citado el: 19 de mayo de 2013.] [http://www.phpmyadmin.net/home\\_page/index.php..](http://www.phpmyadmin.net/home_page/index.php..)
29. www.linux-es.org.El rincón delinux para hispanohablantes. [En línea] 1998. [Citado el: 7 de febrero de 2013.] [http://www.linux-es.org/sobre\\_linux](http://www.linux-es.org/sobre_linux).
30. SAD. [En línea] 2011. [Citado el: 20 de mayo de 2013.] <http://mistock.lcompras.biz/tallersoftware/1259-apache-una-alternativa-viable-para-el-servidor-web..>
31. Miranda Levis, Carlos. Lamp: La Plataforma Más Popular para Software Código libres. [En línea] [Citado el: 7 de enero de 2013.] <http://www.socinfo.com/codigolibre/lamp..>
32. Tecnología y Synergix. [En línea] [Citado el: 4 de junio de 2013.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
33. [En línea] [Citado el: 12 de enero de 2013.] <http://plataformaclipse.com..>
34. Mosqueda Leyva, Marilenis y Oliver Rivas, Nayla. Diseño del Módulo de Logopedia del Sistema de Evaluación del neurodesarrollo en niños. Universidad de las Ciencias Informáticas. 2010. pág. 31, Trabajo de diploma.
35. Pello, Javier. Softqanetwork. [En línea] 13 de julio de 2009. [Citado el: 21 de abril de 2013.] <http://www.softqanetwork.com/requisitos-no-funcionales-nfr>.
36. Hassan, Yusef, y otros. Universitat Pompeu Fabra. Hipertext.Net. [En línea] Universidad de Granada. [Citado el: 13 de enero de 2013.] [http://www.upf.edu/hipertextnet/numero-2/disenio\\_web.html](http://www.upf.edu/hipertextnet/numero-2/disenio_web.html).
37. OBSERVATORIO TECNOLÓGICO. [En línea] [Citado el: 13 de enero de 2013.] profesor de ciencias informáticas en la Universidad Purdue (Indiana, EEUU). <http://recursostic.educacion.es/observatorio/web/es/software/software-general/1040-introduccion-a-la-seguridad-informatica?showall=1>.
38. Ceria, Santiago. Casos de Uso. Un Método Práctico para Explorar Requerimientos. [ed.] Cátedra de Ingeniería del Software I. págs. 3-4.
39. Jacobson I, Jonsson M y Wesley Longman, Addison. Ingeniería de Software Orientada a Objetos, Un acercamiento a través de los casos de uso. 1992.

40. RUMBAUGH, James, JACOBSON, Ivar y BOOCH, Grady. El lenguaje unificado de modelado. Manual de referencia. 2000. Capítulos 8 y 13 .
41. Rational Unified Process Extended Help for RUP. 2003.
42. Morera Aragón, Ismaray. Módulo de reportes estadísticos del sistema de gestión para la ingeniería clínica y electromedicina. Universidad de las Ciencias Informáticas. La Habana : s.n., 2011. pág. 34, Trabajo de diploma.
43. Alexander, Christopher. 9: Patrones de Diseño. [En línea] [Citado el: 5 de junio de 2013.] [http://arco.esi.uclm.es/~david.villa/pensar\\_en\\_C++/vol2/C10.html](http://arco.esi.uclm.es/~david.villa/pensar_en_C++/vol2/C10.html).
44. ZANINOTTO, F. y POTENCIER, F. librosweb.es, Symfony 1.1, la guía definitiva. [En línea] 2009. [Citado el: 5 de junio de 2013.] [http://www.librosweb.es/symfony\\_1\\_1](http://www.librosweb.es/symfony_1_1).
45. Diagramas de Despliegue. [En línea] [Citado el: 5 de junio de 2013.] <http://diadspg.blogspot.com/>.
46. Acuña, Karenny Brito. Desarrollo para Aplicaciones Web. ISBN-13: 978-84-692-6641-0.
47. [En línea] [Citado el: 17 de marzo de 2013.] <http://www.slideshare.net/TomRodriguez/implementacion-de-software>.
48. [En línea] [Citado el: 20 de mayo de 2013.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
49. Martínez, David Luis la Red. Seguridad en los Sistemas Operativos. Argentina : Facultad de Ciencias Exactas, Naturales y Agrimensura, 2007.
50. [En línea] [Citado el: 17 de marzo de 2013.] <http://symfonyes.blogspot.com/2009/07/sfguard-plugin.html>.
51. Estándares de diseño web. [En línea] [Citado el: 15 de mayo de 2013.] <http://www.slideshare.net/dwebslide/estandares-de-diseo-web..>

## BIBLIOGRAFÍA

1. [En línea] [Citado el: 1 de diciembre de , 2012.] <http://es.scribd.com/doc/12983329/Metodologia-de-Desarrollo-de-Software>.
2. [En línea] [Citado el: 10 de diciembre de 2012.] <http://dev.mysql.com/doc/refman/5.0/es/index.html>.
3. [En línea] [Citado el: 10 de diciembre de 2012.] <http://www.symfony.es/noticias/2009/10/05/netbeans-ya-incluye-soporte-para-symfony>.
4. [En línea] [Citado el: 12 de enero de 2013.] <http://plataformaclipse.com>.
5. [En línea] [Citado el: 13 de enero de 2013.] <http://www.genbeta.com/redes-sociales/facebook-notify-notificaciones-de-facebook-integradas-en-gnome>.
6. [En línea] [Citado el: 13 de noviembre de 2012.] <http://myspace-notifier.uptodown.com/>.
7. [En línea] [Citado el: 17 de marzo de 2013.] <http://symfonyes.blogspot.com/2009/07/sfguard-plugin.html>.
8. [En línea] [Citado el: 17 de marzo de 2013.] <http://www.slideshare.net/TomRodriguez/implementacion-de-software>.
9. [En línea] [Citado el: 5 de abril de 2013.] [http://librosweb.es/jobeeet\\_1\\_4/capitulo\\_4/el\\_layout.html](http://librosweb.es/jobeeet_1_4/capitulo_4/el_layout.html).
10. [En línea] [Citado el: 5 de abril de 2013.] [http://librosweb.es/jobeeet\\_1\\_4/capitulo\\_4/la\\_arquitectura\\_mvc.html](http://librosweb.es/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html)
11. [En línea] [Citado el: 5 de diciembre de 2012.] <http://resumenuml.bloggratis.es>.
12. [En línea] [Citado el: 6 de abril de 2013.] <http://www.muycomputerpro.com/2012/12/04/a3-software-notificaciones-electronicas/>.
13. [En línea] [Citado el: 7 de febrero de 2013.] <http://www.juarbo.com/ubuntu-10-4-lucid-lynx>.
14. Acuña, Kareenny Brito. Desarrollo para Aplicaciones Web. ISBN-13: 978-84-692-6641-0.
15. Alexander, Christopher. 9: Patrones de Diseño. [En línea] [Citado el: 5 de junio de 2013.] [http://arco.esi.uclm.es/~david.villa/pensar\\_en\\_C++/vol2/C10.html](http://arco.esi.uclm.es/~david.villa/pensar_en_C++/vol2/C10.html).
16. ARCHIVOSPC.COM. [En línea] [Citado el: 13 de noviembre de 2012.] <<http://myspace-notifier.archivospc.com>.

17. Barroso, Javier Pérez. Sistema de notificación de eventos. Universidad de las Ciencias Informáticas, La Habana : 2011. Trabajo de Diploma.
18. CDAE - Centro de Consultoría y Desarrollo de Arquitecturas Empresariales. [En línea] [Citado el: 1 de diciembre de 2012.] <http://gespro.cdae.prod.uci.cu>.
19. Ceria, Santiago. Casos de Uso. Un Método Práctico para Explorar Requerimientos. [ed.] Cátedra de Ingeniería del Software I. págs. 3-4.
20. DB Designer Fork. [En línea] 2009. [Citado el: 19 de mayo de 2013.] <http://dbdesigner-fork.sourceforge.net/>.
21. Definicion.de. [En línea] [Citado el: 11 de noviembre de 2012.] <http://definicion.de/evento/>.
22. Desarrolloweb.com. Arquitectura Cliente-Servidor. [En línea] [Citado el: 15 de noviembre de 2012.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
23. Diagramas de Despliegue. [En línea] [Citado el: 5 de junio de 2013.] <http://diadspg.blogspot.com/>.
24. Entérate, Internet, Cómputo y Telecomunicaciones. . [En línea] [Citado el: 13 de enero de 2013.] <http://www.enterate.unam.mx/Articulos/2006/febrero/arquitectec.htm>.
25. Estándares de diseño web. [En línea] [Citado el: 15 de mayo de 2013.] <http://www.slideshare.net/dwebslide/estandares-de-diseo-web>.
26. Hassan, Yusef, y otros. Universitat Pompeu Fabra. Hipertext.Net. [En línea] Universidad de Granada. [Citado el: 13 de enero de 2013.] [http://www.upf.edu/hipertextnet/numero-2/disenio\\_web.html](http://www.upf.edu/hipertextnet/numero-2/disenio_web.html).
27. INFOMED. Aspectos Generales del Sistema Nacional de Salud. Infomed. Portal de Salud de Cuba. [En línea] [http://www.sld.cu/sistema\\_de\\_salud/aspectos.html](http://www.sld.cu/sistema_de_salud/aspectos.html).
28. Ing. Dennys Javier Hernández Peña, Ing. Ranniel Rivero Sevilla, MSc. Zoila Esther Morales Tabares, Ing. Velmour Muñoz Casals. SIGICEM: Sistema de Gestión para la Ingeniería Clínica y Electromedicina. Universidad de las Ciencias Informáticas. . La Habana : s.n., 2012. Ponencia. [uciencia-2012-t40-p1648-ponencia-2324.pdf](http://www.uciencia-2012-t40-p1648-ponencia-2324.pdf).
29. Introducción a Patrones. [En línea] Facultad de Ciencias, UNAM, 2006. [www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html](http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html).
30. Introducción al software libre. [En línea] [Citado el: 10 de diciembre de 2012.] Capítulo Entornos y tecnologías de desarrollo. <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/sec-ide.html>.

31. Igual ingenieros. [En línea] 2012. <http://iqingenieros.com/inicio/101-p/321-proceso-unificado-de-desarrollo-de-software.html>.
32. Ivanex. Sistema de selección de rutas óptimas. [En línea] 2012. <http://ivanex.wikidot.com/metodologia>.
33. Jacobson I, Jonsson M y Wesley Longman, Addison. Ingeniería de Software Orientada a Objetos, Un acercamiento a través de los casos de uso. 1992.
34. Katia Nereyda Charchabal Barredo, Yalina Morales Batista. Propuesta de diseño y arquitectura de la información de un sistema de notificaciones dentro de la gestión universitaria. Universidad de las Ciencias Informáticas, La Habana : 2010. Trabajo de Diploma.
35. La Norma Iso Iec 9126. . [En línea] [Citado el: 13 de enero de 2013.] <http://www.mitecnologico.com/Main/LaNormalsolec9126>.
36. La red social Facebook. [En línea] marzo de 2010. [Citado el: 10 de noviembre de 2012.] <http://www.aulaclic.es/articulos/facebook.html>. Artículo24.
37. Laimerys Miranda Arango, David Pérez-Borroto Lorenzo. Módulo Gestión de y Clientes y Facturación de Servicios Técnicos en el Sistema de Gestión para Ingeniería Clínica y Electromedicina. Universidad de las Ciencias Informáticas, La Habana : 2011. Trabajo de Diploma.
38. Lara Pérez, Yaneilis. Implementación y pruebas de un sistema de RHODA 2.1. Universidad de las Ciencias Informáticas. La Habana: s.n., 2011. Trabajo de diploma.
39. Llegó la hora del PostgreSQL. [En línea] [Citado el: 6 de marzo de 2013.] [www.http-peru.com/postgresql1.php](http://www.peru.com/postgresql1.php).
40. Machado Díaz, Maidelys, Coca Ribas, Yader Luis. SACCEM: Módulo de gestión de la información del Departamento de Supervisión del CCEEM versión 1.0. Universidad de las Ciencias Informáticas. La Habana : s.n., 2009. Trabajo de diploma.
41. Martínez, David Luis la Red. Seguridad en los Sistemas Operativos. Argentina: Facultad de Ciencias Exactas, Naturales y Agrimensura, 2007
42. Mendoza Sanchez, María A. informatizate. Metodologías De Desarrollo De Software. [En línea] 7 de junio de 2007. [Citado el: 4 de junio de 2013.] Ing. Informático - UNT, Microsoft Certified

- Professional - MCP, Analista y Desarrolladora - TeamSoft Perú S.A.C. .  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).
43. Miranda Levis, Carlos. Lamp: La Plataforma Más Popular para Software Código libres. [En línea] [Citado el: 7 de enero de 2013.] <http://www.socinfo.com/codigolibre/lamp>.
44. Morera Aragón, Ismaray. Módulo de reportes estadísticos del sistema de gestión para la ingeniería clínica y electromedicina. Universidad de las Ciencias Informáticas. La Habana : s.n., 2011. Trabajo de diploma.
45. Mosqueda Leyva, Marilenis y Oliver Rivas, Nayla. Diseño del Módulo de Logopedia del Sistema de Evaluación del neurodesarrollo en niños. Universidad de las Ciencias Informáticas. 2010. Trabajo de diploma. página 31, capítulo 2 epígrafe 2.13.1.
46. Mucho ruido y muchas nueces: Se completó el proyecto genoma humano. Pettinari, Julia. 1, 2003, Vol. II. ISSN 1666-7948.
47. OBSERVATORIO TECNOLÓGICO. [En línea] [Citado el: 13 de enero de 2013.] profesor de ciencias informáticas en la Universidad Purdue (Indiana, EEUU). <http://recursostic.educacion.es/observatorio/web/es/software/software-general/1040-introduccion-a-la-seguridad-informatica?showall=1>.
48. Pello, Javier. Softqanetwork. [En línea] 13 de julio de 2009. [Citado el: 21 de abril de 2013.] <http://www.softqanetwork.com/requisitos-no-funcionales-nfr>.
49. Pérez Santiesteban, Diana Rosa, et al. Módulo de Notificaciones y Alertas del Sistema de Gestión Universitaria. Universidad de las Ciencias Informáticas. La Habana : s.n., 2012. Ponencia. [uciencia-2012-t46-p1270-ponencia-1476.pdf](http://www.uciencia-2012-t46-p1270-ponencia-1476.pdf).
50. Phpmyadmin.net. [En línea] 2010. [Citado el: 19 de mayo de 2013.] [http://www.phpmyadmin.net/home\\_page/index.php..](http://www.phpmyadmin.net/home_page/index.php..)
51. Pons Capote, Olga. Introducción a las Bases de Datos. El Modelo Relacional. Thomson Learning Ibero. 2005. ISBN 8497323963.
52. Potencier, Fabien y Zaninotto, François. Symfony 1.2, la guía definitiva. 2008.
53. Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Quinta edición. 2002. S.I.: McGraw-Hill Companies. ISBN 8448132149.



54. Programacionweb.net. [En línea] 2009. [Citado el: 19 de mayo de 2013.] <http://www.programacionweb.net/articulos/articulo/?num=686..>
55. Reingeniería de los procesos del negocio. s.l.: Departamento de Procesos y Sistemas. Universidad Simón Bolívar.
56. Retoricas.com. [En línea] [Citado el: 11 de noviembre de 2012.] Todos los derechos reservados correo admin@retoricas.com. <http://www.retoricas.com/2009/05/el-receptor-en-comunicacion.html>.
57. Revista Científico- Técnica Popular de la Juventud Cubana. Sanz, Lucía C. 310, La Habana : Casa Editorial Abri, enero- febrero de 2003. Publicación bimestral.
58. RUMBAUGH, James, JACOBSON, Ivar y BOOCH, Grady. El lenguaje unificado de modelado. Manual de referencia. 2000. Capítulos 8 y 13 .
59. Ruz, Fidel Castro. La idea esencial es acercar los servicios primarios a los ciudadanos. Granma. 2011.
60. SAD. [En línea] 2011. [Citado el: 20 de mayo de 2013.] <http://mistock.lcompras.biz/tallersoftware/1259-apache-una-alternativa-viable-para-el-servidor-web..>
61. Seguridad en los Sistemas Operativos. Argentina: Facultad de Ciencias Exactas, Naturales y Agrimensura, 2007.
62. Seguridad en los Sistemas Operativos. Argentina: Facultad de Ciencias Exactas, Naturales y Agrimensura, 2007.
63. Serrano, Nicolas and Ciordia, Ismael. Openbravo. Openbravo. [En línea] agosto de 2001. [Citado el: 1 de noviembre de 2012.] [www.openbravo.com](http://www.openbravo.com).
64. Symfony. Open-Source PHP Web Framework. [En línea] [Citado el: 15 de noviembre de 2012.] <http://www.symfony-project.org>.
65. Tecnología y Synergix. [En línea] [Citado el: 4 de junio de 2013.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
66. THE FREE DICTIONARY BY FARLEX . [En línea] [Citado el: 10 de noviembre de 2012.] <http://es.thefreedictionary.com/Emissor>.

67. THE FREE DICTIONARY BY FARLEX. [En línea] [Citado el: 10 de noviembre de 2012.] [www.retoricas.com](http://www.retoricas.com) - Todos los derechos reservados - Correo: [admin@retoricas.com](mailto:admin@retoricas.com). <http://es.thefreedictionary.com/Receptor>.
68. THE FREE DICTIONARY BY FARLEX. [En línea] [Citado el: 11 de noviembre de 2012.] <http://es.thefreedictionary.com/evento>.
69. UCI Universidad de las Ciencias Informáticas. [En línea] [Citado el: 5 de junio de 2013.] 2012 Universidad de las Ciencias Informáticas, La Habana, Cuba. Todos los derechos reservados. . <http://www.uci.cu/mision>.
70. Una pedagogía de la comunicación. López Viera, Luís and Kaplún, Mario. Ciudad de La Habana : Caminos.
71. -Visual Paradigm. [En línea] [Citado el: 6 de abril de 2013.] <http://www.visual-paradigm.com/product/vpuml>.
72. -Web Taller. [En línea] [Citado el: 5 de abril de 2013.] <http://www.webtaller.com/construccion/lenguajes/html/lecciones/que-es-xhtml.php>.
73. WEBADICTOS.COM Notificaciones Facebook en tu escritorio. [En línea] [Citado el: 10 abril de 2013.] <http://www.webadictos.com.mx/2013/01/13/notificaciones-facebook-en-tu-escritorio>.
74. -WorldLingo. [En línea] 2013. [http://www.worldlingo.com/ma/enwiki/es/Eclipse\\_%28software%29](http://www.worldlingo.com/ma/enwiki/es/Eclipse_%28software%29).
75. [www.linux-es.org](http://www.linux-es.org). El rincón del linux para hispanohablantes. [En línea] 1998. [Citado el: 7 de febrero de 2013.] [http://www.linux-es.org/sobre\\_linux](http://www.linux-es.org/sobre_linux).
76. Yailis Sosa Vázquez, Alberto Ruiz Cardoso. Análisis y diseño de los procesos de Agregación, Notificaciones y Ajustes al Plan del subsistema de Planificación perteneciente al Sistema Integral de Gestión de Entidades CedruX. Universidad de las Ciencias Informáticas. La Habana : s.n., 2010. Trabajo de diploma.
77. ZANINOTTO, F. y POTENCIER, F. [librosweb.es](http://librosweb.es), Symfony 1.1, la guía definitiva. [En línea] 2009. [Citado el: 5 de junio de 2013.] [http://www.librosweb.es/symfony\\_1\\_1](http://www.librosweb.es/symfony_1_1).

### GLOSARIO DE TÉRMINOS

**ACL:** Lista de control de acceso.

**API (Interfaz de Programación de Aplicaciones):** Conjunto de especificaciones para comunicarse con una aplicación, normalmente para obtener información y utilizarla en otros servicios.

**CDAE:** Centro de consultoría y desarrollo de arquitecturas empresariales.

**CENDA:** Centro nacional de derecho de autor.

**CESIM:** Centro de Informática Médica.

**CICEM:** Centro de Ingeniería Clínica y Electromedicina.

**CU:** Casos de Uso.

**CUN:** Caso de uso del negocio.

**CUS:** Caso de uso del sistema.

**DOM:** Modelo de objetos del documento o Modelo de objetos para la representación de documentos.

**EVA:** Entorno Virtual de Aprendizaje.

**Framework:** Se conoce como marco de trabajo y constituye un conjunto de conceptos, metodologías y herramientas de administración y diseño para el desarrollo de forma estandarizada de una aplicación.

**GPL:** Licencia pública general.

**Herramientas CASE:** Conjunto de aplicaciones informáticas orientadas al incremento de la productividad en el desarrollo de software, las siglas CASE vienen dadas por su nombre en inglés Computer Aided Software Engineering que se conoce como Ingeniería de Software Asistida por Computadoras.

**HTML:** HyperText Markup Language (Lenguaje de Marcas de Hipertexto).

**IDE:** Entorno integrado de desarrollo.

**ISO:** Organización internacional de normalización.

**MVC:** Modelo Vista Controlador, es un patrón de arquitectura de software.

**MySQL:** Es utilizado para el diseño y programación de base de datos de tipo relacional. Cuenta con millones de aplicaciones, permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos.

**ORM (Object-Relational Mapping):** el mapeo de objetos-relacional es una técnica de programación utilizada para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos, creando una base de datos orientada a objetos virtual, por encima de la base de datos relacional.

**PHP:**(Hypertext PreProcessor) Lenguaje de programación para el desarrollo de Web dinámicas. Originalmente se conocía como Personal Home Page tools, herramientas para páginas personales (en Internet).

**POO:** Programación orientada a objetos.

**RF:** Requerimientos funcionales.

**RNF:** Requerimientos no funcionales.

**RUP:** Rational Unified Process. El Proceso Unificado es un proceso de desarrollo de software (Conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software).

**SGU:** Sistema de Gestión Universitaria.

**SIGICEM:** Sistema de Gestión para la Ingeniería Clínica y Electromedicina.

**Symfony:** Es un framework que está programado en PHP-5 y está enfocado al desarrollo de aplicaciones web en el mismo lenguaje de programación.

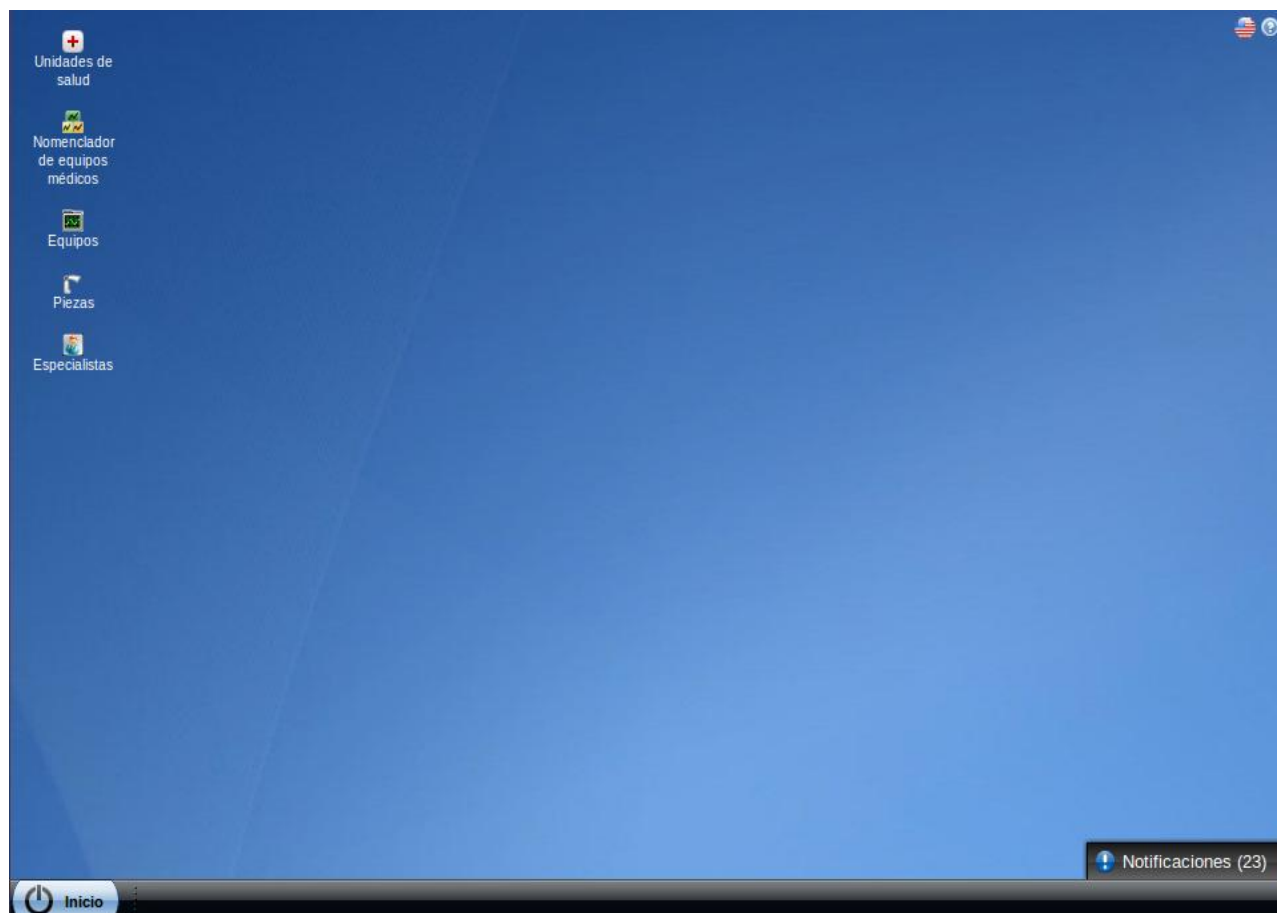
**Sun Microsystems:** es una empresa informática adquirida por Oracle Corporation, anteriormente parte de Silicon Valley, fabricante de semiconductores y software. Sun Microsystems es la creadora de la popular suite ofimática OpenOffice.org y el lenguaje de programación Java.

**TIC:** Tecnologías de la Información y las Comunicaciones.

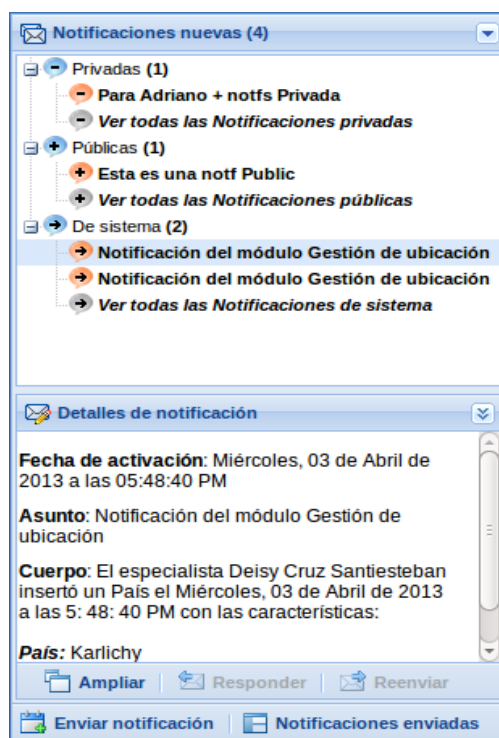
**UCI:** Universidad de Ciencias Informáticas.

**UML:** (Unified Modeling Language) Web Services: Servicio Web. Programa que implementa el protocolo.

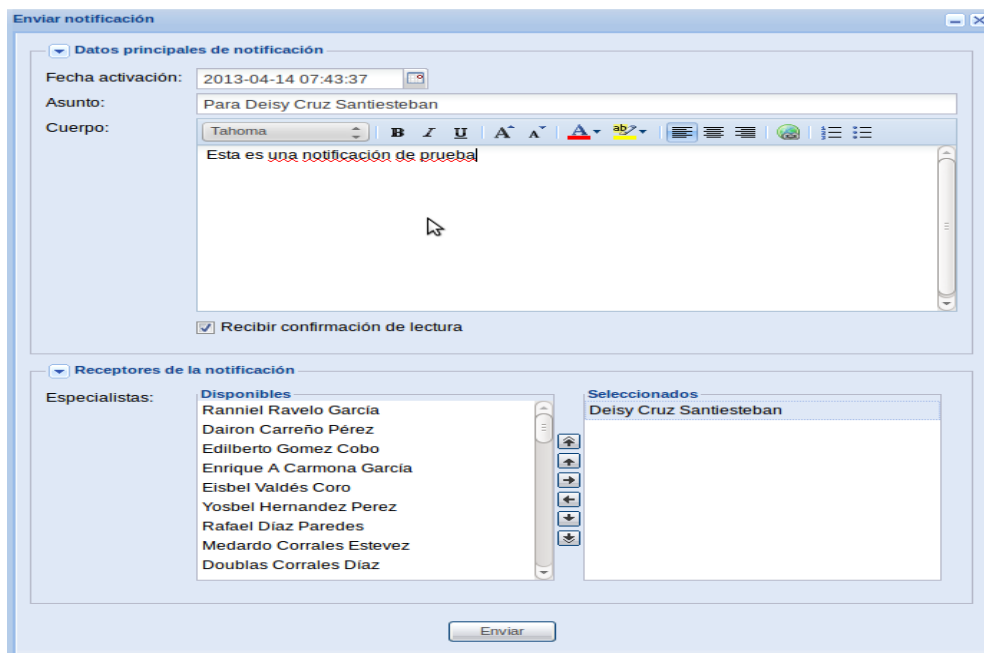
ANEXOS



Anexo1: Escritorio del Sistema de Gestión para Ingeniería Clínica y Electromedicina.



Anexo2: Panel de notificaciones.



Anexo3: Enviar notificación.

**Reenviar notificación**

**Datos principales de notificación**

Fecha activación: 2013-04-14 19:27:30

Asunto: Fwd: Private notification for Adriano

Cuerpo: Tahoma  
Private notification for Adriano

Recibir confirmación de lectura

**Receptores de la notificación**

**Especialistas:**

**Disponibles**

- Ranniel Ravelo Garcia
- Dairon Carreño Pérez
- Edilberto Gomez Cobo
- Enrique A Carmona Garcia
- Eisbel Valdés Coro
- Yosbel Hernandez Perez
- Rafael Diaz Paredes
- Medardo Corrales Estevez
- Doublas Corrales Diaz

**Seleccionados**

Anexo4: Reenviar notificación.

**Responder notificación**

**Datos principales de notificación**

Fecha activación: 2013-04-14 19:26:45

Asunto: Re: Para Adriano + notfs Privada

Cuerpo: Tahoma  
Esta es una notf Privada para Adriano, Madrid Champion!!!

Recibir confirmación de lectura

Anexo5: Responder notificación.



Notificaciones enviadas por el especialista Adriano De Armas Pimienta

Reenviar | Exportar a PDF | Eliminar

<input type="checkbox"/>	Clasificación	Asunto	Fecha de creación	Fecha de activación	Cuerpo	Editar
<input checked="" type="checkbox"/>	Privada	Para Deisy	2013-03-04 06:17:45	2013-03-04 06:17:45		-
<input type="checkbox"/>	Privada	Para Deisy	2013-03-05 02:01:28	2013-03-05 02:01:28		-
<input checked="" type="checkbox"/>	Privada	Para Deisy	2013-03-16 14:26:04	2013-03-16 14:26:04		-
<input type="checkbox"/>	Privada	Para Deisy	2013-03-27 06:48:42	2013-03-27 06:48:42		-
<input checked="" type="checkbox"/>	Privada	Para Deisy 1.1	2013-03-27 06:49:25	2013-03-27 06:49:25		-
<input type="checkbox"/>	Privada	Para Deisy de Adriano	2013-04-13 07:02:22	2013-04-13 07:02:22		-
<input checked="" type="checkbox"/>	Pública	Public notification for Deisy	2013-04-14 08:10:47	2013-04-14 08:10:47		-
<input type="checkbox"/>	Pública	Public notification for Deisy 1.1	2013-04-14 08:11:40	2013-04-14 08:11:40		-
<input checked="" type="checkbox"/>	Pública	Public notification for Deisy 1.2	2013-04-14 08:12:51	2013-04-14 08:12:51		-

Mostrando grupo 1 - 9 Total: 9 filas

Anexo6: Listado de notificaciones enviadas.

Notificaciones recibidas por el especialista Deisy Cruz Santiesteban

Listado de notificaciones privadas

Opciones de notificación | Exportar a PDF | Eliminar

<input type="checkbox"/>	Asunto	Fecha de activación	Detalles	Remitente	Estado
<input type="checkbox"/>	Para Deisy	2013-03-04 06:17:45		Adriano De Armas Pimienta	Leída
<input type="checkbox"/>	Para Deisy 1.1	2013-03-27 06:49:25		Adriano De Armas Pimienta	No leída
<input type="checkbox"/>	Para Deisy de Adriano	2013-04-13 07:02:22		Adriano De Armas Pimienta	Leída

Mostrando grupo 1 - 3 Total: 3 filas

Listado de notificaciones públicas

Opciones de notificación | Exportar a PDF | Eliminar

<input type="checkbox"/>	Asunto	Fecha de activación	Detalles	Remitente	Estado
<input type="checkbox"/>	Public notification for Deisy 1.2	2013-04-14 08:12:51		Adriano De Armas Pimienta	No leída

Mostrando grupo 1 - 1 Total: 1 filas

Listado de notificaciones de sistema

Opciones de notificación | Exportar a PDF | Eliminar

<input type="checkbox"/>	Asunto	Fecha de creación	Detalles	Estado
<input type="checkbox"/>	Confirmación de lectura de notificación del especialista Adriano De Arma...	2013-03-03 07:06:12		Leída
<input type="checkbox"/>	Notificación del módulo Gestión de equipos	2013-03-13 21:10:59		No leída
<input type="checkbox"/>	Notificación del módulo Gestión de ubicación	2013-05-08 08:52:31		Leída

Mostrando grupo 1 - 3 Total: 3 filas

Anexo7: Listado de notificaciones recibidas.

Opciones de administración

**Visualizar notificaciones por especialista**

Especialistas:

Fecha:

Visualizar las notificaciones recibidas de los especialistas

Visualizar las notificaciones enviadas de los especialistas

**Habilitar | Deshabilitar envío y recibo de notificaciones por especialista**

Especialistas:

**Visualizar logs de notificaciones**

Fecha del log:

Anexo 8: Opciones de administración.

Logs de notificaciones del 14 de Abril de 2013

Remitente	Clasificación	Fecha de creación	Fecha de activación	Detalles
Adriano De Armas Pimienta	Pública	2013-04-14 08:10:47	2013-04-14 08:10:47	
Adriano De Armas Pimienta	Pública	2013-04-14 08:11:40	2013-04-14 08:11:40	
Adriano De Armas Pimienta	Pública	2013-04-14 08:12:51	2013-04-14 08:12:51	
SIGICEM	De sistema	2013-04-14 08:20:07	2013-04-14 08:20:07	
SIGICEM	De sistema	2013-04-14 08:20:12	2013-04-14 08:20:12	
Deisy Cruz Santiesteban	Privada	2013-04-14 08:21:21	2013-04-14 08:21:21	

Mostrando grupo 1 - 6 Total: 6 filas

Anexo 9: Logs de notificaciones

Detalles de log

Asunto, Cuerpo y Confirmación de lectura

Asunto: Public notification for Deisy 1.2

Cuerpo: Public notification for Deisy 1.2

Confirmación de lectura

Receptores

Nombre	Apellidos	Especialidad que atiende
Ranniel	Ravelo García	Esterilización
Dairon	Carreño Pérez	Electrónica médica
Edilberto	Gomez Cobo	Electromecánica
Deisy	Cruz Santiesteban	Esterilización

Mostrando grupo 1 - 4 Total: 4 filas

Anexo 10: Detalles del logs.