

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



FACULTAD 7

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

**SUBSISTEMA DE REGISTRO DE EQUIPOS MÉDICOS PARA EL SISTEMA
INFORMATIZADO DE SALUD**

AUTOR:

Denys Pérez-Borroto Lorenzo

TUTORES:

Ing. Velmour Muñoz Casals

Ing. Arieskien Mendoza Guerra

**La Habana, junio 2011
"Año 53 de la Revolución"**



“El futuro de Cuba debe ser necesariamente un futuro de hombres de ciencias...”

“Fidel Castro Ruz”

Datos del contacto

Ing. Velmour Muñoz Casals.

Correo electrónico: vcasals@uci.cu

Profesora graduada de Ingeniería en Ciencias Informáticas en el año 2007 en la Universidad de la Ciencias Informáticas, posee la categoría docente de Instructor. Ha sido tutora de 5 tesis de grado en diferentes años, así como Jefa de un tribunal en el año 2008. Fue Jefa del Proyecto de la UCI con la Oficina de Información del Consejo de Estado y actualmente desempeña el rol de Analista en el proyecto Electromedicina del Dpto. Sistemas de Apoyo a la Salud.

Ing. Arieskien Mendoza Guerra.

Correo electrónico: amendoza@uci.cu

Profesor graduado de Ingeniero en Ciencias Informáticas del curso 2007-2008. Líder de desarrollo del proyecto SIGICEM durante 2 años. Ha impartido la asignatura de PP3, Gráfico por computadoras y Segundo Perfil. En el curso 2008-2009 formó parte del tribunal en 6 tesis de grado y fue tutor de 2 tesis y en el curso 2009-2010 de una. Actualmente especialista del Centro de desarrollo de Sistemas de Informática Médica y profesor adjunto a la Facultad 7.

Agradecimientos

Son muchas las personas que de una forma u otra, me han ayudado en el quehacer cotidiano de esta hermosa escuela que es la Universidad de las Ciencias Informáticas, que más que una escuela es como si fuera mi casa también, puesto que la mayor parte del tiempo me lo he pasado aquí en los cinco años de la carrera. Quiero agradecerles a todos los que siempre me apoyaron y confiaron en mí. Primeramente voy a empezar por una persona que no se encuentra en estos momentos con nosotros como estudiantes, pero que siempre estuvo ahí dando su apoyo desde el primer año con todo el que lo necesitara, ella es Amailis Cueto que se encuentra de misión en Venezuela dando un paso al frente una vez más por nuestra universidad e hizo todo lo posible porque este trabajo de diploma siguiera adelante sin ningún tipo de problemas para cuando ella estuviera ausente. Agradecerles también a mis queridos tutores Velmour Muñoz Casals y Arieskjen Mendoza Guerra que siempre me apoyaron en todo momento, y luchábamos y discutíamos para que todo saliera bien y como debía ser. Gracias Vel por tu carácter y forma de ser que fue eso lo que me impulsó a vencer cualquier dificultad y a seguir adelante, gracias a Arieskjen por tu espíritu dinámico y rapidez en la toma de decisiones cada vez que se presentaba un problema que para mi era casi imposible de resolver, de los dos aprendí mucho y por todas esas cosas gracias. Quiero agradecer también a todo el grupo de Electromedicina, a Zoila, Ranniel, Yirlenis y a Denys todos son buenos y excelentes profesores.

Agradecer a todos mis colegas y compañeros del apartamento 89204 y a los que entraron conmigo en primer año, a Damián, Raidel, Arrebato, Franklin, Melquiades y José Carlos que ya no están en la escuela, Luismel, Renan, René, Mojena, Guillermo, Raisal y a todas las niñas hermosas que han compartido buenos y malos momentos tanto en la escuela como en el aula.

Por último y no deja de ser menos que nada, sino al contrario los deje para el final porque son lo más especial que me han pasado en la vida, y es mi familia mi mamá, mi papá, mi abuela y mi hermano, que es

Agradecimientos

la causa principal por la cual estoy aquí presente. A mis padres María Matilde y Heriberto por darme lo mejor en todo momento, y enseñarme a ser cada día una persona digna y respetable, y a mi abuelita Alicia que la quiero con toda mi alma y siempre me apoya incluso aunque yo no tenga la razón, gracias por hacer de mi alguien mejor en la vida, muchas gracias.

Dedicatoria

Quiero dedicar este Trabajo de Diploma a mis padres, María Matilde Lorenzo Arrojo y Heriberto David Pérez-Borroto Sánchez, a mi abuelita Alicia Arrojo Olazabal y a mi hermano David Pérez-Borroto Lorenzo por ser las personas que más quiero en este mundo y estoy dispuesto a dar mi vida...

Denys Pérez-Borroto Lorenzo.

RESUMEN

La Universidad de las Ciencias Informáticas (UCI), juega un papel fundamental en el desarrollo de aplicaciones para la informatización del país, convirtiéndose en un importante eslabón en los sectores representativos de la sociedad.

Como parte del desarrollo de las aplicaciones en la salud, surge en la UCI el Centro de Informática Médica (CESIM), que tiene varios departamentos enfocados al mejoramiento de la calidad de la atención médica, dentro de los que se encuentran el departamento Sistema de Apoyo a la Salud (SAS), el cual se dio a la tarea de mejorar la gestión de la información de los equipos médicos en el Sistema Informatizado de Salud (SISALUD).

El objetivo de la presente investigación es desarrollar un subsistema para SISALUD que permita el registro de los equipos médicos, de esta forma posibilita la gestión de la información de los mismos en las instituciones de salud.

En el desarrollo del subsistema se utilizó Symfony como framework, PostgreSQL de sistema de gestión de bases de datos y como herramienta CASE el Visual Paradigm que hace uso del lenguaje de modelado UML. Todo esto proceso guiado por la Metodología de desarrollo RUP.

Como beneficio fundamental del subsistema se encuentran la mejora de la gestión de la información de los equipos médicos en el Sistema Informatizado de Salud (SISALUD), que facilitará que la información de entrada y salida de los equipos médicos en las instituciones de salud se comparta y se aproveche de forma más eficiente y como un recurso colectivo.

Palabras Claves: gestión de la información, equipos médicos, informatización de la salud.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 EQUIPOS MÉDICOS	6
1.2 GESTIÓN DE LA INFORMACIÓN	6
1.3 SISTEMAS EXISTENTES EN EL ÁMBITO INTERNACIONAL	7
1.4 SISTEMAS EXISTENTES EN EL ÁMBITO NACIONAL	8
1.5 METODOLOGÍA DE DESARROLLO: RUP	9
1.6 LENGUAJE UNIFICADO DE MODELADO: UML 2.1	10
1.7 SISTEMA GESTOR DE BASE DE DATOS: POSTGRESQL 5.1	11
1.8 VISUAL PARADIGM FOR UML 3.6	11
1.9 NETBEANS 6.9	12
1.10 SYMFONY 1.4.....	13
1.11 TECNOLOGÍA FRAMEWORK IU: EXTJS 2.2.....	14
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	16
2.1 BREVE DESCRIPCIÓN DEL PROBLEMA.....	16
2.2 MODELO DE NEGOCIO	17
2.3 DIAGRAMA DE CASO DE USO DEL NEGOCIO	19
2.4 DESCRIPCIÓN DE CASOS DE USOS DEL NEGOCIO	20
2.5 DIAGRAMA DE ACTIVIDAD CORRESPONDIENTE A CADA CASO DE USO.....	23
2.6 MODELO DE OBJETO	26
2.7 ESPECIFICACIÓN DE REQUISITOS	26
2.8 MODELO DEL SISTEMA.....	30
2.9 DIAGRAMA DE CASOS DE USO DEL SISTEMA	31
2.10 DESCRIPCIÓN DE ALGUNOS CASOS DE USOS DEL SISTEMA.....	34
CAPÍTULO 3: DESCRIPCIÓN DEL SISTEMA.....	38
3.1 ANÁLISIS DE COMPONENTES DE SIGICEM.....	38

Tabla de Contenido

3.2	MODELO DE ANÁLISIS	38
3.3	MODELO DE DISEÑO	40
3.4	PATRÓN ARQUITECTÓNICO	41
3.5	PATRONES DE DISEÑO EMPLEADOS.....	43
CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA		51
4.1	MODELO DE DATOS	51
4.2	MODELO DE IMPLEMENTACIÓN.....	52
4.3	TRATAMIENTO DE ERRORES.....	53
4.4	SEGURIDAD	54
4.5	MODELO DE DESPLIEGUE.....	54
CONCLUSIONES		56
RECOMENDACIONES.....		57
ANEXO1.....		66
2.4.11	REFERENCIAS BIBLIOGRÁFICAS.....	58
2.4.12	BIBLIOGRAFÍA.....	61
GLOSARIO DE TÉRMINOS.....		64

Introducción

Los avances en los sectores de la informática y las comunicaciones han sacudido al mundo durante las últimas décadas, cubriendo un enorme abanico de funciones, que van desde las más simples cuestiones domésticas hasta los cálculos científicos más complejos. Esta disciplina se aplica a numerosas y variadas áreas del conocimiento o la actividad humana. La rapidez con que cambian las tecnologías, establece una dinámica diferente en las sociedades, imponiendo nuevos retos a alcanzar, para no quedar rezagados en el camino dominante de la informatización.

A partir de las tendencias actuales, la evolución de las Tecnologías de la Informática y las Comunicaciones (TIC) en los próximos años han provocado un gran impacto en el ámbito social, económico, político y cultural propiciando un impulso hacia lo novedoso. El mundo está viviendo una época de innovaciones profundas que implican cambios en el impulso de la ciencia y la tecnología. En tal sentido, puede señalarse que las TIC fomentan el avance del desarrollo tecnológico.

Cuba no está aislada de todo el acontecer tecnológico que sucede en el mundo, y a pesar de la situación económica que atraviesa, la idea de alcanzar un progreso sostenible, en vez de debilitarse se ha reforzado cada día más. Surgida al calor de la batalla de ideas por una idea del Comandante en Jefe Fidel Castro Ruz y precisamente para impulsar el desarrollo de la informática en Cuba, se crea la Universidad de las Ciencias Informáticas (UCI), quien tiene como objetivo formar a sus estudiantes bajo la premisa de combinar la docencia con la producción; preparándolos como futuros informáticos y hombres de ciencia.

La facultad 7 de dicha institución asume el reto de la informatización del Sistema de Salud Pública, convirtiéndose en el eslabón productivo el Centro de Informática Médica (CESIM) que tiene como misión ser un centro de excelencia dedicado al desarrollo de productos, sistemas, servicios y soluciones de alta calidad y competitividad para la optimización del trabajo y mejoramiento de la atención médica. Cuenta con 7 departamentos, dentro de los que se encuentra el departamento Sistema de Apoyo a la Salud (SAS). Este departamento tiene entre sus tareas fundamentales mejorar la gestión de la información de los equipos médicos para Sistema Informatizado de Salud (SISALUD). Ya que este sistema, no cuenta con el reporte defectación, que recoge datos sobre el historial de incidencias, alta, y listado de todos los

equipos en las unidades de salud, por lo que imposibilita la gestión de la información de los mismos en las diferentes instituciones de salud.

Basado en las necesidades anteriormente descritas, se plantea el siguiente **problema a resolver**: ¿Cómo lograr la gestión de la información de los equipos médicos para el Sistema Informatizado de Salud?

El **objeto de estudio** se centra en el proceso de gestión de la información en el Sistema Informatizado de Salud, enmarcado en el **campo de acción** el proceso de gestión de la información de equipos médicos en el Sistema Informatizado de Salud.

El **objetivo general** consiste en desarrollar un subsistema que permita el registro de equipos médicos para el Sistema Informatizado de Salud.

Tareas de la investigación:

1. Analizar el estudio del arte referente a la gestión de equipos médicos obteniéndose las bases de la investigación propuesta.
2. Asimilar las herramientas y tecnologías propuestas por el Centro de Informática Médica (CESIM) y el Departamento de Sistemas de Apoyo a la Salud (SAS) obteniendo el ambiente de desarrollo y arquitectura del sistema.
3. Determinar los servicios del Sistema Informatizado de Salud para la solución propuesta.
4. Diseñar los artefactos correspondientes a las disciplinas de RUP para la obtención de la documentación necesaria acerca de la investigación.
5. Diseñar prototipo de Interfaz de Usuario.
6. Implementar las funcionalidades con las que contará el subsistema.

Métodos científicos seleccionados

Entender los detalles del problema existente y qué se desea resolver, puede resultar en ocasiones engorroso, debido a esto es necesario al comienzo de la investigación definir los principales métodos científicos que posteriormente se utilizarán. Estos métodos permiten estudiar las características que no son observables y las fenomenológicas, respectivamente.

Según plantea Rolando Alfredo Hernández, estos métodos: *representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.* (12) Pueden ser encontrados distintos métodos, los cuales pueden ser aplicados a las investigaciones para obtener de manera más rápida y confiable los resultados esperados. Para la presente investigación se han definido los que se listan a continuación:

- ◆ **Histórico – Lógico:** permite la realización del estudio de los fenómenos y acontecimientos en un orden cronológico, para conocer la evolución y desarrollo de la esencia de la investigación que se está desarrollando.
- ◆ **Analítico-Sintético:** propiciará resumir, enunciar y describir los requerimientos funcionales de la solución planteada, debido a que este método permite buscar la esencia del problema que se está tratando, los rasgos que lo caracterizan y lo distinguen. Además permite la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
- ◆ **Modelación:** método teórico que ofrece parte de la información necesaria acerca del objeto que se estudia, se trata de explicar la realidad con la creación de diagramas; los cuales pueden ser presentados en sustitución de la realidad. Mediante su utilización se elaborarán diferentes tipos de diagramas (incluidos en la concepción de UML 2.1, especialmente) que brindarán información clara sobre el tema de estudio, permitiendo descubrir nuevas relaciones y cualidades del objeto de estudio.
- ◆ **Grupo de Discusión:** método empírico que se basa en efectuar una reunión de varias personas que discuten sobre un tema de interés común, con la ayuda de un coordinador y un secretario. A través de este se busca adquirir más información sobre un tema, tomar decisiones conjuntas o ambas acciones.

Como **aporte y novedad** se espera:

Gestión y control efectivo, sencillez, rapidez y ahorro: De una forma sencilla, el Sistema Informatizado de Salud (SISALUD) tiene acceso instantáneo a toda la información necesaria referente a los equipos médicos, en cada una de las instituciones de salud.

Uso racional de los recursos: Facilitará que la información de entrada y salida de los equipos médicos se comparta y se aproveche de forma más eficiente y como un recurso colectivo. Como consecuencia, se reducen drásticamente situaciones como la duplicidad de documentos archivados.

Productividad y valor añadido: Este sistema de gestión de información de los equipos médicos, además de ahorro de costes, genera una productividad y valor añadido adicionales, originados por el rápido acceso a la información dentro del SISALUD, como son listados de equipos defectados, historial de incidencias de cada uno de ellos, así como el registro de todos los equipos en las unidades de salud, a través del perfeccionamiento y automatización de los procesos relacionados.

Finalmente todo esto conllevará a una agilización de la toma de decisiones ante eventos ocurridos con el equipamiento médico.

El documento de la investigación científica está estructurado por capítulos como se muestra 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 definen las características del sistema y se enmarca en el objeto de estudio. Además, se hace un análisis del modelo del negocio correspondiente al sistema. También se definen requisitos funcionales y no funcionales así como un prototipo de interfaz externa.

Capítulo 3: Análisis y Diseño del sistema.

Se centra en la modelación detallada y la construcción de la estructura de la aplicación. En este capítulo se definen la estructura y los elementos del diseño, se muestran los diagramas de clases del análisis y del diseño de los casos de uso así como el modelo de datos. Conformándose, finalmente, el modelo de diseño, el cual constituye una base para la futura implementación.

Capítulo 4: Implementación.

Se describe la implementación del sistema propuesto a través del diagrama de componentes y la distribución del mismo mediante el diagrama de despliegue. Se realiza la descripción de los componentes del sistema.

Capítulo 1: Fundamentación teórica

En el presente capítulo se hace mención de un grupo de conceptos que son importantes para el desarrollo de la investigación y que serán referenciados en el resto del documento. Así mismo se efectúa un análisis del los sistemas de registros de equipos médicos en el ámbito internacional, nacional y en la UCI. Además se expone las principales características de las herramientas, tecnologías y metodología que se emplearán en el desarrollo del sistema.

1.1 Equipos médicos

Los equipos médicos no es más que la tecnología electrónica en medicina, en todo su ciclo de vida: adquisición, instalación / validación, mantenimiento, uso y retirada al final de su vida útil.

Los equipos médicos son el resultado de combinar:

- ◆ La tecnología electrónica más actual.
- ◆ El alto nivel científico de la medicina y de la más avanzada mundialmente.
- ◆ La concepción humanista y la racionalidad del sistema de salud. (1)

1.2 Gestión de la información

Gestión de la información es el proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma, y se establece como un recurso básico para cualquier organización. (2)

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades. (3)

1.3 Sistemas existentes en el ámbito internacional

1.3.1 Sistema informático móvil para la gestión de los equipos médicos

La utilización de dispositivos móviles está hoy en día vigente en diversas esferas de la vida, convirtiéndose en una herramienta eficaz en la gestión del equipamiento médico. Con el objetivo de aprovechar estas tecnologías, en el ámbito de la ingeniería biomédica, es que se desarrolló este sistema. El mismo utiliza como plataforma central un teléfono móvil, el cual mediante la conexión de un servidor web, es capaz de enviar o recibir información relativa a algún equipo médico. Mediante la decodificación de un tipo de código de barra de 2D, conocidos como QR-Codes, se simplifica y dinamiza el proceso de gestión. Dichos códigos identifican los equipos médicos en una base de datos para que al ser fotografiados y decodificados mediante el dispositivo móvil se pueda acceder a la información relevante del equipo en cuestión. (4)

El proyecto en su estado actual constituye una herramienta de apoyo básica en el mantenimiento del equipamiento biomédico. Es además una alternativa moderna, competitiva y económica en el mercado actual. (5)

1.3.2 Sistema de mantenimiento asistido por computador (SYSMANCOR)

Sistema de mantenimiento asistido por computador, orientado al control de riesgos en equipos médicos. (6)

Características principales:

- ◆ Nomenclatura ECRI.
- ◆ Denominación de la propiedad.
- ◆ Planificación de mantenimiento preventivo.
- ◆ Gestión de órdenes de trabajo.
- ◆ Cálculo de tiempo medio entre fallos.
- ◆ Reportes e indicadores.
- ◆ Cálculo de costos.
- ◆ Análisis de productividad.
- ◆ Historial de incidencias.

Características del sistema:

- ◆ Sistema operativo: Windows 98, Milenium, 2000, XP o Vista.
- ◆ Procesador Pentium III o superior.
- ◆ Espacio en disco 4 GB. (7)

1.3.3 Equipos de control: Grupo Industrial Advantage Tecnología

Uno de los principales equipos de control que se fabrica en Grupo Industrial Advantage Tecnología son los equipos de control para cámaras térmicas para el tratamiento a embalajes de madera según la norma NOM-144 SEMARNAT. Estos equipos se fabrican especialmente para controlar los equipos instalados en una cámara térmica que van desde motores, quemadores, controladores de temperatura, relevadores, equipos de monitoreo, bombas, alarmas visuales, alarmas auditivas, encendido de lámparas para alumbrado, alimentación para equipos externos, etc. Estos cuentan con una gran variedad de accesorios según las necesidades de la cámara. (8)

Estos equipos se pueden operar de manera manual o automática. De manera manual se utiliza normalmente para hacer pruebas de funcionamiento de los diferentes equipos que se controlan y de manera automática es como funciona habitualmente en todos los procesos. (9)

1.4 Sistemas existentes en el ámbito nacional

1.4.1 Sistema Automatizado Cubano para el Control de Equipos Médicos (SACCEM)

Es un sistema de vigilancia para los Equipos Médicos, como parte del seguimiento pos mercado que permita identificar y alertar sobre eventos adversos ocurridos con los mismos dentro del territorio nacional. Para su implementación se basa en un Sistema de Información Nacional de Vigilancia que sistematiza, colecta datos, procesa y difunde las informaciones sobre incidentes, fallas, “problemas”, o eventos adversos asociados a Equipamientos Medico. Para ello se ha desarrollado una aplicación informática para el control y seguimiento de los equipos médicos instalados en el Sistema Nacional de Salud (SNS) que permiten controlar la gestión de la información de estos equipos en la etapa de post comercialización. (10)

Dicho sistema fue desarrollado siguiendo las pautas propuestas por el SNS para su informatización. Con la culminación y puesta en práctica del mismo se habrá dado un paso de avance para la automatización y control de la Vigilancia de Equipos Médicos por el CEEM en el sistema nacional de Salud. (11)

Análisis comparativo con la propuesta de solución

Después de hacer un estudio del estado del arte se llega a la conclusión que dichos sistemas poseen funcionalidades que se desea poseer en el nuevo módulo que será implantado, pero las mismas no satisfacen todas las expectativas existentes. Por ejemplo: el Sistema informático móvil para la gestión de los equipos médicos, resulta muy costoso su implantación y en el mercado mundial, el país no cuenta con los recursos necesarios para poner en las manos de trabajadores móviles para la gestión de mantenimiento de equipos médicos, además de toda la tecnología que esto trae consigo, debido a la resistencia hostil, y el bloqueo existente actualmente en la isla.

SYSMANCOR tiene la facilidad que hace uso de un ordenador para el control y gestión pero el mismo necesita de sistema operativo: Windows 98, Milenium, 2000, XP o Vista, en la cual en el proyecto se trabaja con SO Linux y necesita además de un procesador Pentium III o superior, la cual imposibilita guiarse por ese sistema. El Centro de Control de Equipos Médicos utiliza una herramienta que sirve de base para la actual aplicación, es rentable y fácil de utilizar para lograr los objetivos trazados, esto conlleva a ser posible la respuesta al problema que se plantea, que no es más que la propuesta de un módulo o sistema que permita la gestión de equipos médicos entre CNE y SISALUD, pero dicho sistema no satisface todas las necesidades del subsistema que se desea desarrollar.

1.5 Metodología de Desarrollo: RUP

Una metodología de ingeniería del software es un proceso para producir software de forma organizada, la cual emplea una colección de técnicas y convenciones de notación predefinidas.

El Proceso Unificado Racional (Rational Unified Process) en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. (13)

Características del Proceso Unificado

1. Está dirigido por los casos de uso:
 - ◆ Desde la especificación hasta el mantenimiento
2. Se centra en la arquitectura:
 - ◆ La arquitectura es prioritaria desde el principio hasta el final
 - ◆ Se facilita el refinamiento progresivo de la arquitectura
3. Iterativo e incremental:
 - ◆ El trabajo se divide en iteraciones pequeñas en función de la importancia de los casos de uso y el análisis de riesgos. (14)

1.6 Lenguaje Unificado de modelado: UML 2.1

El Lenguaje Unificado de Modelado (UML) es una de las herramientas más emocionantes en el mundo actual del desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas. (15)

Es el más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. (16)

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en la implementación del software entregar 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. (17)

1.7 Sistema Gestor de Base de Datos: PostgreSQL 5.1

Los Gestores de Bases de Datos (SGBD) permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Los SGBD poseen grandes ventajas entre las que se destacan:

- ◆ Facilidad de manejo de grandes volúmenes de información.
- ◆ Gran velocidad de ejecución de las consultas.
- ◆ Independencia del tratamiento de información.
- ◆ Seguridad de la información. (18)

PostgreSQL es un gestor de bases de datos orientadas a objetos muy conocido y usado en entornos de software libre ya que cumple con un conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales.

Las principales mejoras en PostgreSQL incluyen:

- ◆ Control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad mientras la base de datos permanece disponible para consultas.
- ◆ Posee importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos y disparadores.
- ◆ Posee mejoras en los tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
- ◆ La velocidad del código del motor de datos es de aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80%. (19)

1.8 Visual Paradigm for UML 3.6

Visual Paradigm es una herramienta CASE para modelamiento UML muy potente, gratuita, fácil de instalar, utilizar y actualizar. Te permite dibujar todo tipo de diagramas UML, revertir código fuente a

modelos UML, generar código fuente desde los diagramas UML, y mucho más. Incluye los objetos más recientes de UML además de diagramas de casos de uso, diagramas de clase, diagramas de componentes, reversa instantánea para Java, C++, DotNetExe/dll, XML, XML Schema, y Corba IDL, ofrece soporte para Rational Rose, integración con Microsoft Visio, además permite generar reportes y documentación en HTML/PDF. Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (20)

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. Visual Paradigm 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, capaz de conectar en caché, 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, SybaseAdaptive Server Enterprise, Sybase SQL Anywhere, Microsoft SQL Server, PostgreSQL, MySQL y otros. (21)

1.9 NetBeans 6.9

La integración 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, para visualizar el resultado sin necesidad de utilizar una consola de comandos externa. (22)

Características principales:

- ◆ Auto-completado y documentación de funciones PHP: Rápido acceso a la documentación de PHP, y si se necesita más información se provee el link directo a la función.
- ◆ Auto-completado de código propio: Esto es una consecuencia del punto anterior, al documentar código con el formato esperado, estos serán mostrados.
- ◆ Atajos de teclado muy útiles: Permite a través de varios comandos la ejecución de numerosas funcionalidades y existen muchas más como integración con Xdebug, soporte para Symfony, Zend Framework, Smarty, historia local para archivos. Entre otros. (23)

1.10 Symfony 1.4

Symfony es un framework PHP que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes y aburridos de las aplicaciones web, esto permite dejar que el programador se dedique a aportar valor y así desarrollar las características únicas de cada proyecto. El mismo aumenta exponencialmente la productividad y ayuda a mejorar la calidad de tus aplicaciones web para aplicar todas las *buenas prácticas* y patrones de diseño que se han definido para la web. (24)

Posee varias características que posibilita la fácil creación de aplicaciones, entre ellas están:

- ◆ Fácil de instalar y configurar en sistemas Windows, Mac y Linux
- ◆ Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server)
- ◆ Compatible solamente con PHP 5 desde hace años, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP
- ◆ Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización
- ◆ Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma

Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Proporciona una arquitectura, componentes y herramientas a los desarrolladores para construir aplicaciones web complejas más rápido. (25)

1.11 Tecnología Framework IU: EXTJS 2.2

Las aplicaciones web tradicionales tienen problemas como la recarga continua de las páginas cada vez que el usuario pide nuevo contenido, o la poca capacidad multimedia, para lo cual se han hecho necesarios plug-ins externos.

ExtJS encaja dentro de este esquema como un motor que permite crear Aplicaciones Ricas en Internet (RIA) mediante Java script. Si se enmarca a ExtJS dentro del desarrollo RIA, éste sería el render de la aplicación que controla el cliente y que ese encarga de enviar y obtener información del servicio.

Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas con los componentes predefinidos así como un manejador de layouts similar al que provee Java Swing, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari, etc.).

Además la ventana flotante que provee ExtJS es excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido lo cual le da una ventaja tremenda frente a otros. (26)

EXT JS se destaca por ser rápido y poseer widgets personalizables y una licencia *comercial* y otra *open source*. Tiene una API muy intuitiva, funciona en la mayoría de los navegadores y se encuentra muy bien documentado. También contiene numerosas funcionalidades que permiten añadir interactividad a las páginas HTML, como: cuadros de diálogo y Quicktips para mostrar mensajes de validación e información sobre campos individuales. (27)

En el capítulo, se estudiaron algunos sistemas existentes utilizados para la gestión de equipos médicos. Se definieron los principales conceptos relacionados con el sistema y se caracterizaron los lenguajes, tecnologías y metodologías a utilizar para el desarrollo del Subsistema Registro de Equipos Médicos.

Una vez concluida la investigación sobre los sistemas existentes para el registro de equipos médicos se contempló la necesidad de desarrollar una aplicación para las diferentes áreas de salud, ya que los mismos no cumplen con todos los requerimientos propuestos por el cliente, además son propietarios y hay que pagar elevados costos para la adquisición de sus licencias. Asimismo, se definieron las herramientas

a utilizar para la documentación e implementación de la solución propuesta, adoptando como ambiente de desarrollo según las pautas establecidas por el CESIM y el Departamento SAS la metodología de desarrollo RUP, apoyado en el lenguaje de modelado UML en su versión 2.1; como IDE de desarrollo NetBeans en su versión 6.9, Visual Paradigm 3.6 y BD Designer 4 Fork 1.0 como herramientas CASE; empleando los frameworks Symfony 1.4 y ExtJS 2.2, todo esto soportado sobre una plataforma con PostgreSQL 8.4, Apache 2.2 y como sistema operativo GNU Linux, distribución Ubuntu 10.4.

Capítulo 2: Características del sistema

En el presente capítulo del documento de la investigación científica se realiza a continuación una breve descripción del problema, con el objetivo de facilitar el entendimiento del mismo por parte del cliente y por los futuros lectores del documento. Además ofrece la posibilidad de explicar el flujo del negocio a través de los diagramas correspondientes modelados en la herramienta CASE Visual Paradigm, detallándose los actores y trabajadores que intervienen en la situación problemática planteada. Igualmente, se describen las reglas del negocio, así como se definen los requisitos funcionales y no funcionales del sistema, finalizando con la representación del diagrama de caso de usos del sistema y la descripción de estos.

2.1 Breve descripción del problema

Cuba actualmente cuenta con un sistema; el Sistema para la Ingeniería Clínica y Electromedicina (SIGICEM) que permite tener un registro de todas las piezas disponibles a nivel nacional y cuántas piezas están en déficit. Se realizan estudios para ver qué equipos son más eficaces, económicos y de larga durabilidad, permite además tener conocimiento de las condiciones del equipo. Así como pronosticar qué piezas son más eficientes que otras y evaluar la asignación del presupuesto destinado para la compra de piezas en este sector. A pesar de estas ventajas, el sistema carece de un registro de equipos médicos, que permita tener el control de las entradas y salidas de los mismos en cada una de las Instituciones de Salud.

Se desea por parte del Sistema informatizado de Salud (SISALUD), hacer uso de esta funcionalidad, para tener constancia física de la existencia de estos equipos, así como su uso en las diversas Instituciones de Salud, brindando la posibilidad de un mejor registro que contribuya con calidad y eficiencia el tratamiento con los mismos.

2.2 Modelo de negocio

Permite tener una visión clara de cómo fluye toda la información referente a los equipos médicos y por lo tanto definir casos de usos, roles y responsabilidades que serán reflejadas en los modelos de casos de uso del negocio y de objetos plasmados a continuación.

2.2.1 Especificación de las reglas del negocio

Describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. El subsistema de registro de equipos médicos posee ciertas reglas, que se han de tener en cuenta a la hora de modelar los casos de usos que de él se derivan, previendo que puedan ser controlados para que el negocio no colapse.

Número.	Tipo	Nombre	Descripción
2.1	Textuales	Baja por inventario	Para darle baja por inventario a un equipo, debe haberse dado primero baja técnica (si la baja es por deterioro del equipo) o haber sido dado de alta en inventario en la unidad de destino si es un traslado.
2.2	Relación	Permiso para baja por inventario.	Un equipo solo puede ser dado de baja por inventario por el administrador de la Unidad de Salud.
2.3	Relación	Permiso para baja técnica	Un equipo solo puede ser dado de baja técnica por el electromédico de la Unidad de Salud o un especialista de electromedicina, dependiendo de la complejidad del equipo.

2.4	Textuales	Notificación de equipo	Cuando un equipo llega a la Unidad de Salud se le envía una notificación al electromédico.
2.5	Relación	Llegada de equipo	El encargado de defectar un nuevo equipo a la unidad de salud es el electromédico.
2.6	Textuales	Traslado equipo	Cuando un equipo es trasladado de la Unidad de Salud este no culmina el proceso hasta haber obtenido su número de inventario de la Unidad de Salud destino.

Tabla 1 Justificación de los actores.

2.2.2 Justificación de los actores

Actor	Descripción
Especialista	Es el encargado de hacer los reportes de incidencias de equipos en la Unidad de salud.
Unidad de Salud	Es el encargado de recibir el equipo y dar baja al mismo, ya sea por traslado de unidad o por deterioro del equipo.

Tabla 2 Justificación de los actores.

2.2.3 Justificación de los trabajadores

Trabajador	Descripción
Electromédico	Encargado de revisar los datos del nuevo equipo, efectuando la defectación del mismo, registra las incidencias de cada uno y puede dar baja técnica a estos si lo requieren.
Administrador	Persona encargada de dar ubicaciones, especialista y programas del nuevo equipo, así como dar baja por inventario o realizar un traslado del mismo.

Tabla 3 Justificación de los trabajadores.

2.3 Diagrama de Caso de Uso del Negocio

Un caso de uso del negocio representa una funcionalidad dentro del negocio que se estudia, por lo que se corresponde con una secuencia de acciones con un orden lógico y que producen un resultado observable para ciertos actores del negocio.

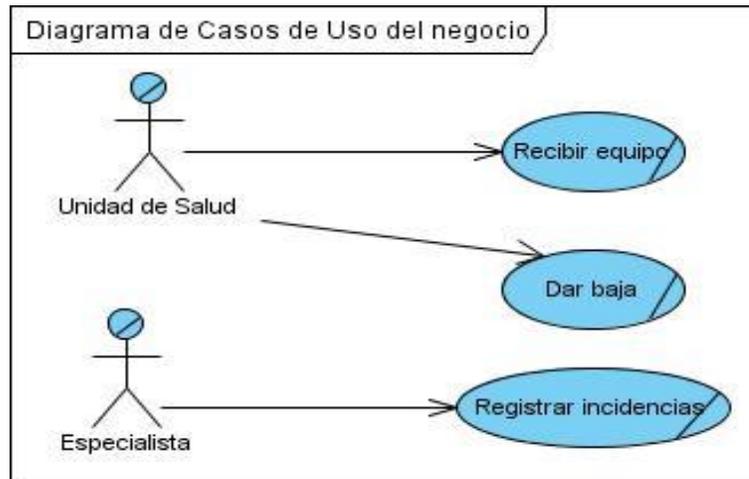


Figura 1 Diagrama de caso de Uso del Negocio.

2.4 Descripción de Casos de Usos del Negocio

A continuación se muestra la descripción de los casos de usos del negocio que se ha identificado.

Caso de Uso del Negocio	Recibir equipo	
Actores	Unidad de Salud	
Resumen	El caso de uso inicia cuando se recibe un nuevo equipo en la unidad de salud, posteriormente se registran sus datos y se archivan en la tarjeta de estiba, adicionándose el equipo en el listado de reportes de equipos a defectar, finalizando así el caso de uso.	
Acción del actor	Respuesta del proceso de negocio	

<p>1. La unidad de salud recibe los datos archivándolos en la tarjeta de estiba y le envía una notificación al electromédico de la llegada del equipo a la institución.</p>	<p>1.1. El electromédico revisa los datos y defecta el equipo.</p> <p>1.4 El administrador de la unidad de salud le asigna los servicios, departamento y programas a los que va a pertenecer el equipamiento, así como su número de inventario.</p> <p>1.5 Posteriormente el administrador actualiza el listado de equipos de la unidad de salud.</p>
<p>Otras secciones</p>	<p>Actividad 1.2 Si los campos no se han llenado correctamente o no se han validado, no se actualizará el listado de equipos.</p>

Tabla 4. Descripción del Caso de Uso del Negocio Recibir equipo.

<p>Caso de Uso del Negocio</p>	<p>Registrar incidencias</p>	
<p>Actores</p>	<p>Especialista</p>	
<p>Resumen</p>	<p>El caso de uso inicia cuando un equipo está reportado como roto o en mantenimiento, efectuándose una orden de servicio, la cual refleja las incidencias del equipo; registrándose la misma en el historial de incidencias, finalizando así el caso de uso.</p>	
<p>Acción del actor</p>	<p>Respuesta del proceso de negocio</p>	
<p>1. El especialista realiza el reporte de rotura de equipo</p>	<p>1.1. Se recibe los datos de la solicitud y se crea la misma.</p> <p>1.2. Se actualiza el historial de incidencias del equipo.</p> <p>1.3 Se envía un mensaje al especialista "Reporte</p>	

	correctamente”.
Otras secciones	Actividad 1.1 Si los campos no se han llenado, no se crea la solicitud.

Tabla 5. Descripción del Caso de Uso del Negocio Registrar incidencias.

Caso de Uso del Negocio	Dar baja
Actores	Unidad de Salud
Resumen	El caso de uso inicia cuando el administrador de la unidad de salud solicita la baja de un equipo médico al electromédico que atiende la institución, ya sea por problemas técnicos o por traslado de este a otra unidad de salud, finalizando así el caso de uso.
Acción del actor	Respuesta del proceso de negocio
1. La unidad de salud realiza la solicitud de dar baja al equipo.	<p>1.1. Se muestra un interfaz donde se escogerá el tipo de solicitud que se requiera según el caso.</p> <p>1.2. En caso de dar baja técnica se actualizará el historial de incidencias del equipo.</p> <p>1.3. En caso de ser un traslado, al equipo se le da baja por inventario, eliminándose este del listado de equipos médicos de la unidad de salud.</p>

Tabla 6. Descripción del Caso de Uso del Negocio Dar baja.

2.5 Diagrama de actividad correspondiente a cada Caso de Uso

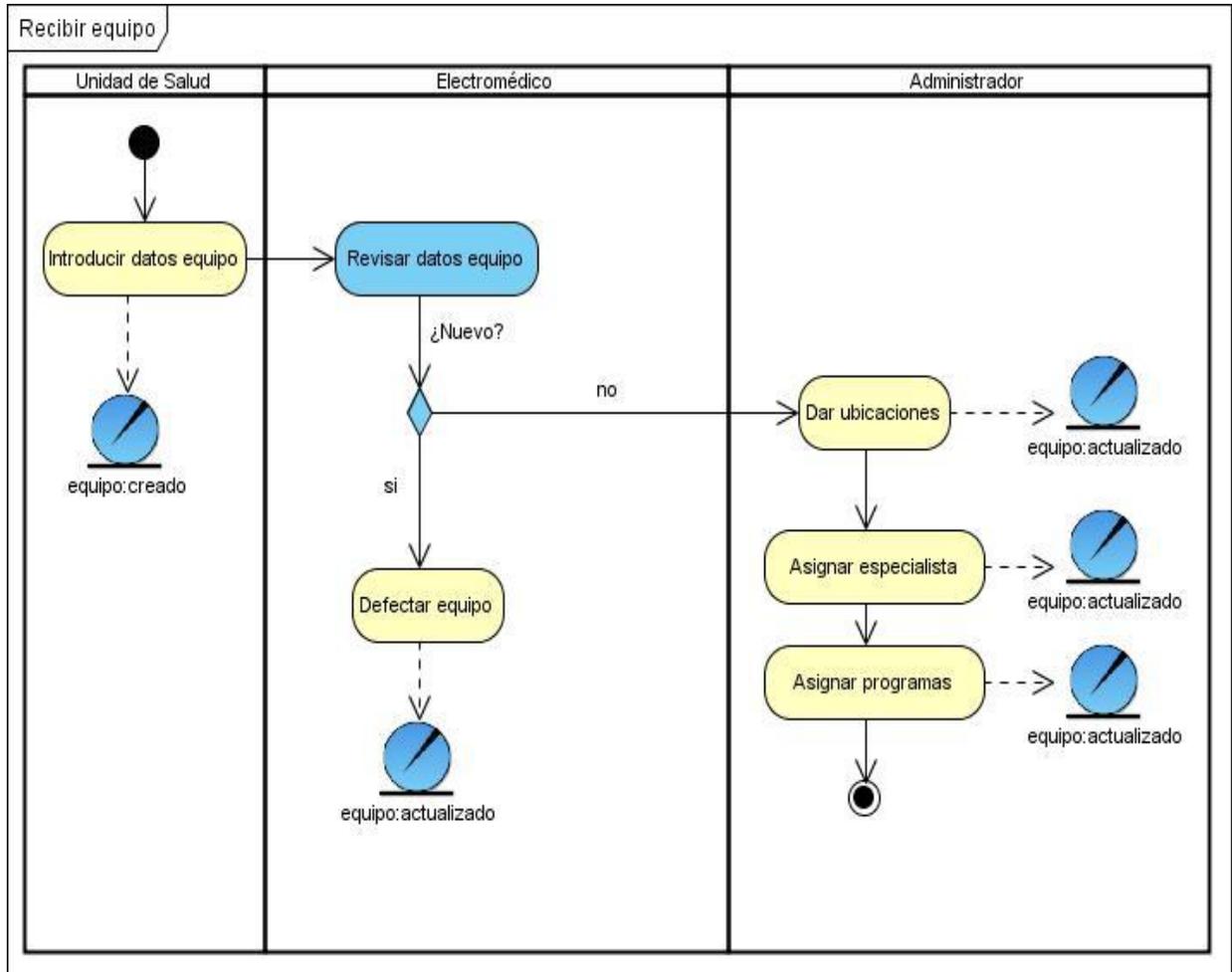


Figura 2 Diagrama de Actividad del CUN Recibir equipo.

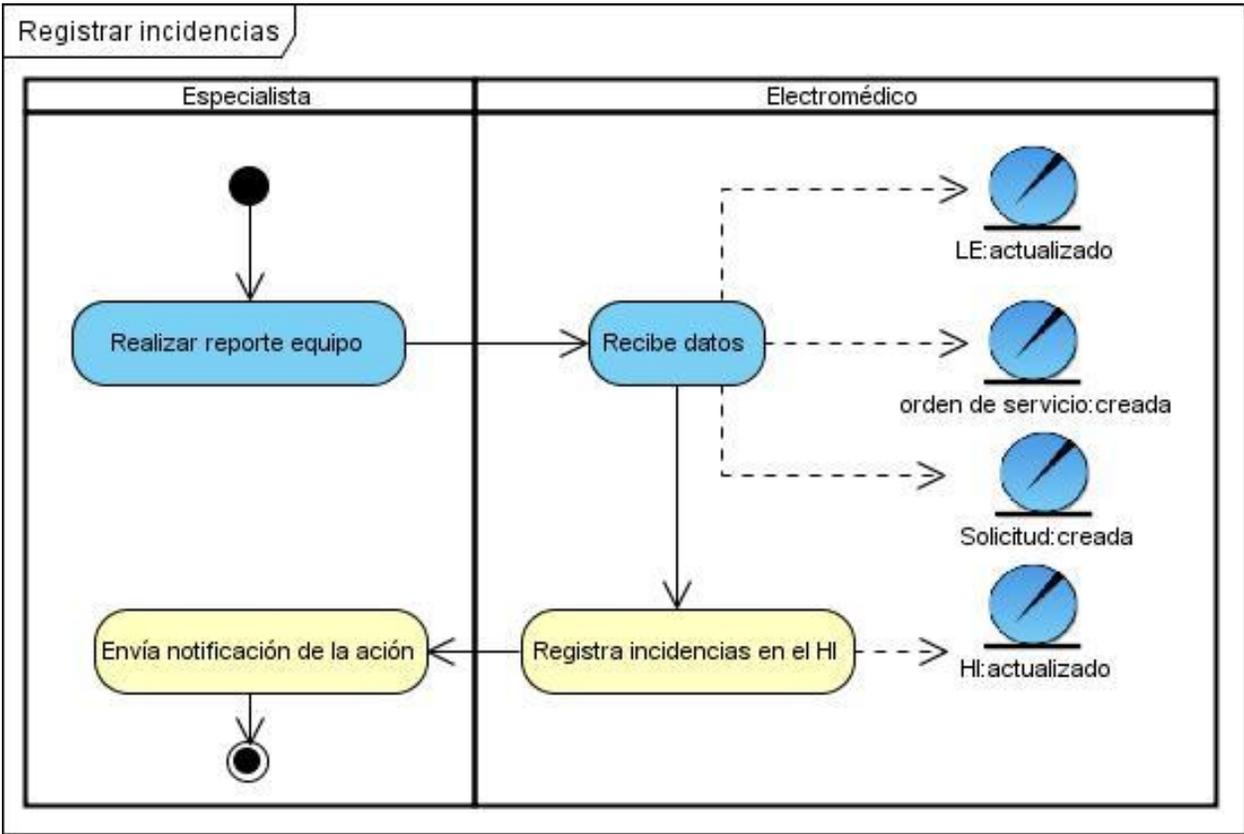


Figura 3 Diagrama de Actividad del CUN Registrar incidencias.

Leyenda

LE: Listado de equipos.

HI: Historial de incidencias.

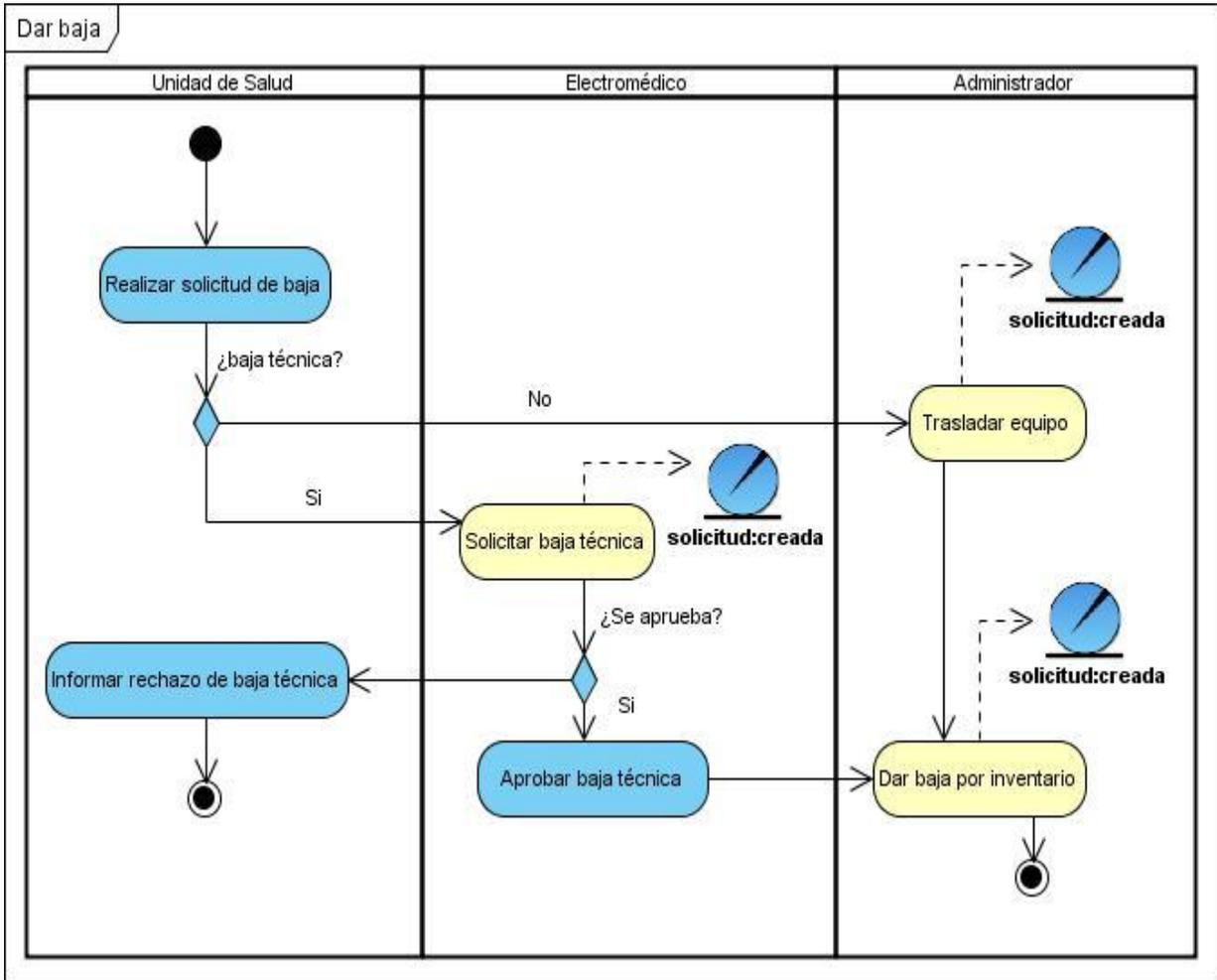


Figura 4 Diagrama de Actividad del CUN Dar baja.

2.6 Modelo de objeto

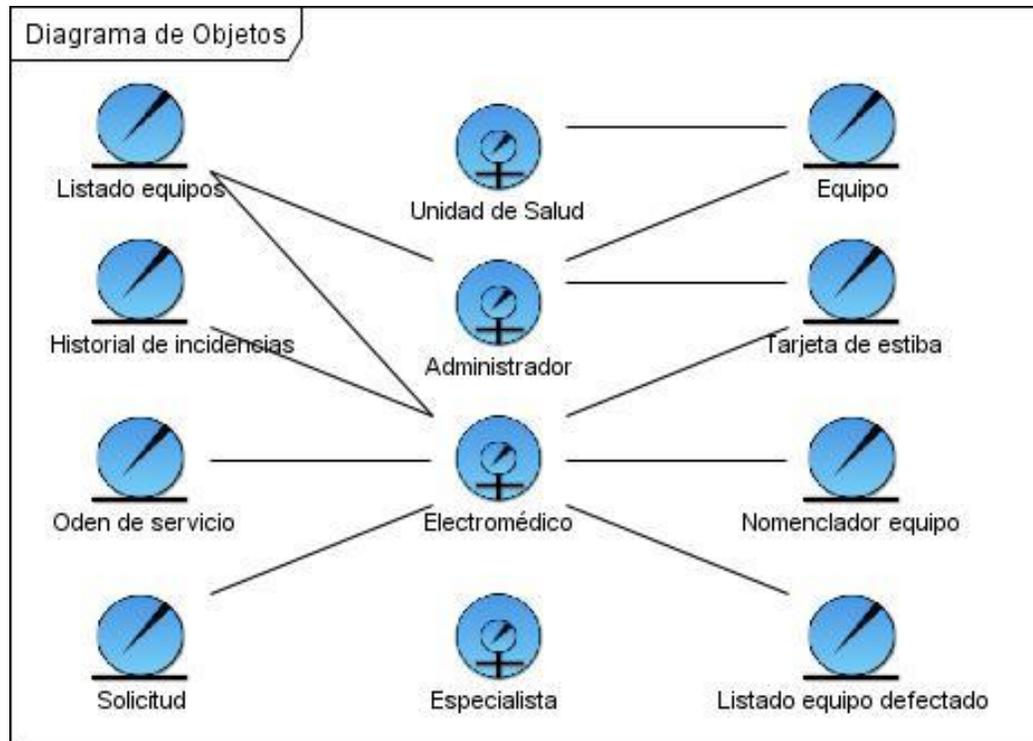


Figura 5 Modelo de Objeto.

2.7 Especificación de requisitos

La captura de requisitos es una de las actividades fundamentales que se desarrolla en el flujo de trabajo de Requerimientos, los mismos pueden dividirse en requerimientos funcionales y no funcionales. Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. (28)

2.7.1 Requisitos funcionales del sistema

Gestionar reporte de equipo sin defectar

RF 1.1 Insertar reporte de equipo sin defectar.

RF 1.2 Buscar reporte de equipo sin defectar.

RF 1.3 Mostrar listado de reportes de equipos sin defectar.

Gestionar equipo defectado

RF2.1 Buscar equipo defectado.

RF 2.2 Mostrar listado de equipos defectados.

Gestionar equipos en inventario

RF3.1 Registrar número de inventario del equipo.

RF3.2 Eliminar equipo en inventario.

RF 3.3 Modificar equipo en inventario.

RF 3.4 Buscar equipo en inventario.

RF 3.5 Mostrar listado de equipos en inventario.

RF 3.6 Imprimir listado de equipos en inventario.

Gestionar equipos en el origen

RF 4.1 Mostrar listado de equipos en el origen.

RF 4.2 Buscar equipo en el origen.

RF 4.3 Imprimir listado de equipos en el origen.

Gestionar equipos en el destino

RF 5.1 Mostrar listado de equipos en el destino.

RF 5.2 Buscar equipo en el destino.

RF 5.3 Imprimir listado de equipos en el destino.

Gestionar reporte de equipo por rotura

RF 6.1 Crear reporte de equipo por rotura.

RF 6.2 Mostrar reporte de equipo por rotura.

RF 6.3 Buscar reporte de equipo por rotura.

RF 6.4 Imprimir reporte de equipo por rotura.

2.7.2 Requisitos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades representan las características que hacen al producto atractivo, usable, rápido o confiable.

1. Usabilidad

La aplicación 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, se hace necesario el uso de un menú que le facilite, a los mismos el trabajo con el sistema.

2. Apariencia o interfaz externa

La aplicación será diseñada con una interfaz amigable, fácil de usar por el usuario, similar a la interfaz 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.

3. Rendimiento

El sistema 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. El sistema 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; todo esto se logrará luego de una validación de los datos, manipulación de eventos en el cliente, y en el servidor, aquellas que por cuestiones de seguridad o acceso a los datos lo requieran.

4. Soporte

Una vez puesta en marcha la aplicación, utilizada por los usuarios finales, se recogerá toda la información referente a los defectos, para incorporar las mejoras sugeridas al subsistema.

5. Portabilidad

El subsistema será multiplataforma (Linux-Windows) lo que permitirá ejecutarse sobre diferentes sistemas operativos sin importar sus versiones, y sin necesidad de modificar su código fuente.

6. Seguridad

El subsistema contará con una política de seguridad diseñada en base a la restricción de usuarios en dependencia del nivel jerárquico que cumpla su rol dentro del mismo.

7. Confiabilidad

El subsistema 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. Además la interfaz se realizará fundamentalmente, con campos de selección para garantizar la seguridad de la información contenida en la base de datos.

8. Políticos-culturales

Se deberá hacer un uso correcto del idioma español en la Interfaz de la aplicación, con logotipos e imágenes que se encuentren en correspondencia con el carácter de la misma.

9. Legales

El subsistema 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 aunque no es libre, la Universidad de las Ciencias Informáticas posee licencia para su uso.

10. Software

Se requiere del navegador Mozilla Firefox 3.6 para la visualización de la aplicación. Se usará como gestor de bases de datos MySQL Server, garantizando 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.

11. 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.

2.8 Modelo del sistema

Es una de las actividades fundamentales que se desarrollan en el Flujo de trabajo de Requerimientos durante la fase de inicio del desarrollo de un software con el objetivo de definir el ámbito del sistema. (29)

8.1 Actores del sistema

Trabajador	Descripción
Electromédico	Es el encargado de la gestión de de los equipos médicos. Es el que los inicia y se encarga de realizar cualquier operación relacionada con el sistema.
Administrador	Es el encargado del chequeo y control de los equipos médicos en la Unidad de Salud. Inicia todo el proceso de inventario a un equipo.

Tabla 7. Actores del Sistema.

2.9 Diagrama de Casos de Uso del Sistema

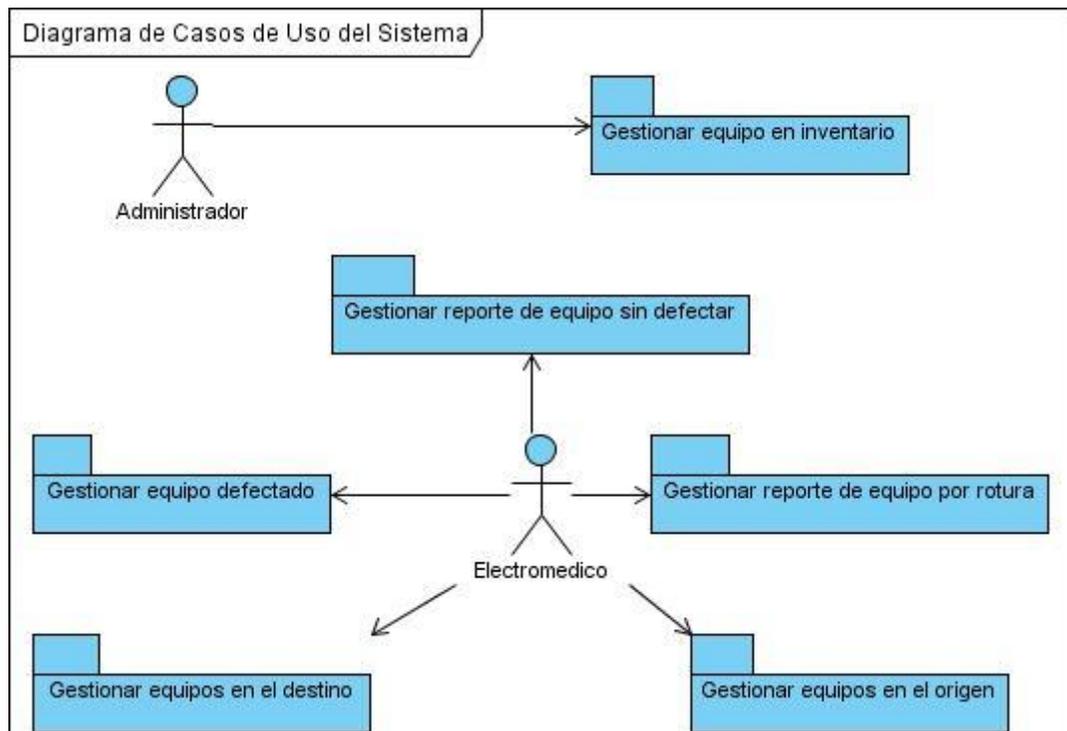


Figura 6 Diagrama de paquetes de casos de usos del sistema.

2.9.1 Diagrama de CUS: Gestionar equipo en inventario

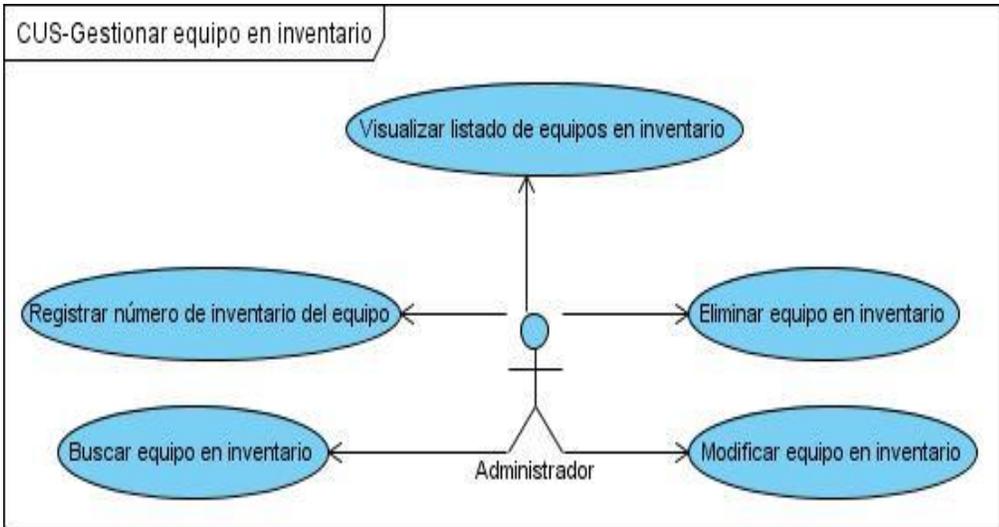


Figura 7 Diagrama de casos de uso del sistema: Gestionar equipo en inventario.

2.9.2 Gestionar reporte de equipo sin defectar

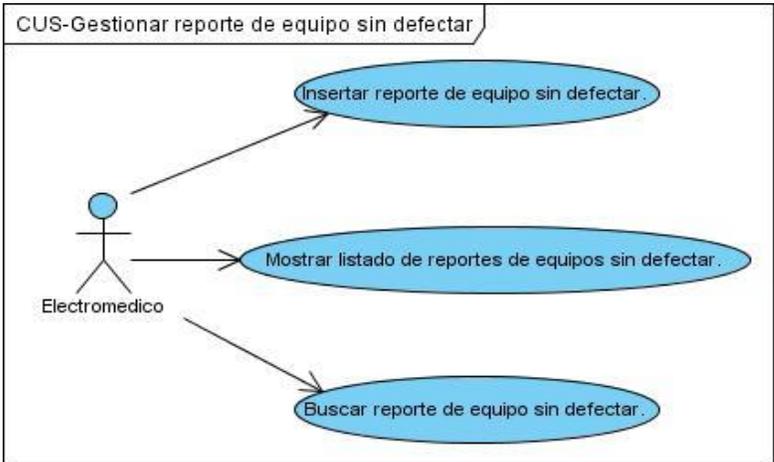


Figura 8 Diagrama de casos de uso del sistema: Gestionar reporte de equipo sin defectar.

2.9.3 Gestionar equipo defectado



Figura 9 Diagrama de casos de uso del sistema: Gestionar equipo defectado.

2.9.4 Gestionar reporte de equipo por rotura

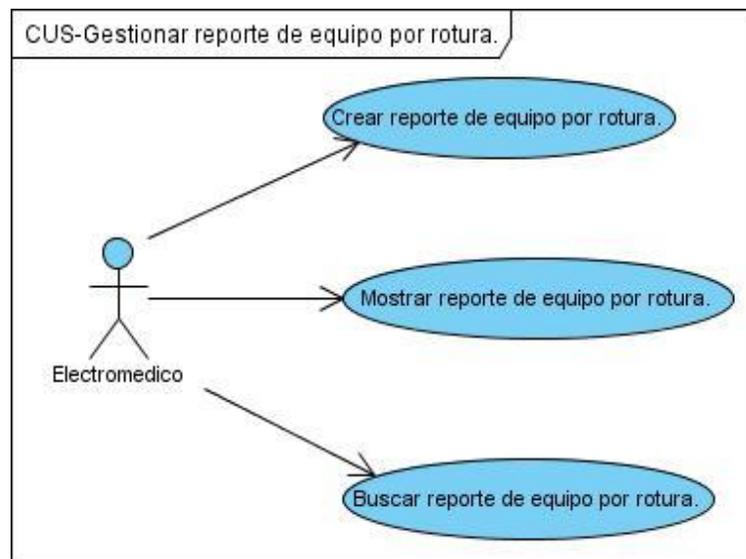


Figura 10 Diagrama de casos de uso del sistema: Gestionar reporte de equipo por rotura.

2.9.5 Gestionar equipos en el destino



Figura 11 Diagrama de casos de uso del sistema: Gestionar equipos en el destino.

2.9.6 Gestionar equipos en el origen



Figura 12 Diagrama de casos de uso del sistema: Gestionar equipos en el origen.

2.10 Descripción de algunos casos de usos del sistema

Caso de Uso:	Insertar reporte de equipo sin defectar.
Actores:	Electromédico.

Resumen:	El caso de uso inicia cuando el actor accede a la opción Insertar reporte de equipo sin defectar, el sistema brinda la posibilidad de introducir y seleccionar los datos para Insertar reporte de equipo sin defectar, el actor introduce los datos del reporte de equipo sin defectar, el sistema inserta reporte de equipo sin defectar, finalizando así el caso de uso.
Referencias	RF 1

Tabla 8. Descripción de casos de uso del sistema: Insertar reporte de equipo sin defectar.

Caso de Uso:	Buscar reporte de equipo sin defectar
Actores:	Electromédico
Resumen:	El caso de uso inicia cuando el actor accede a la opción Buscar reporte de equipo sin defectar, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el reporte de equipo sin defectar, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los reporte de equipo sin defectar que cumplen con los criterios de búsqueda, el caso de uso termina.
Referencias	RF 1.2

Tabla 9. Descripción de casos de uso del sistema: Buscar reporte de equipo sin defectar.

Caso de Uso:	Visualizar listado de reportes de equipos sin defectar.
Actores:	Electromédico
Resumen:	El caso de uso inicia cuando el actor accede a Visualizar listado de reportes de equipos sin defectar, el sistema muestra todas las entidades con la información correspondiente a mostrar, permite ordenar o filtrar por algún dato seleccionado, el caso de uso termina.

Referencias	RF 1.3
--------------------	--------

Tabla 10. Descripción de casos de uso del sistema: Visualizar listado de reportes de equipos sin defectar.

Caso de Uso:	Modificar equipo en inventario.
Actores:	Administrador
Resumen:	El caso de uso inicia cuando el actor selecciona un equipo en inventario y accede a la opción Editar, el sistema muestra los datos del equipo en inventario y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos del equipo en inventario, el caso de uso termina.
Referencias	RF 1.8

Tabla 11. Descripción de casos de uso del sistema: Modificar equipo en inventario.

Caso de Uso:	Imprimir listado de equipos en inventario
Actores:	Administrador
Resumen:	El caso de uso inicia cuando el actor accede a la opción Imprimir listado de equipos en inventario o presiona el icono que indica la acción, el sistema genera el reporte. El actor puede imprimir el reporte, consultarlo o guardarlo según sus necesidades, el caso de uso termina.
Referencias	RF 1.11

Tabla 12. Descripción de casos de uso del sistema: Imprimir listado de equipos en inventario.

En el capítulo se describe la modelación del negocio, conociéndose el procedimiento para el registro de equipos. Se identificaron los actores y trabajadores del negocio, así como las reglas a tener en cuenta durante todo el proceso, los casos de uso del negocio y la descripción de los mismos.

Se definieron los requisitos funcionales y se obtuvo un listado de requerimientos no funcionales a tener en cuenta para el desarrollo de la aplicación así como varios paquetes que contienen a los casos de uso del sistema que pueden agruparse lógicamente, mostrándose los diagramas de casos de uso para cada paquete. Igualmente se describen los casos de uso del sistema, así como los actores del mismo, dando una visión global de cómo está prevista su implementación y cómo va a ser su funcionamiento.

Capítulo 3: Descripción del sistema

En el presente capítulo se abordarán los aspectos relacionados con el flujo de trabajo análisis y diseño; el cual tiene como objetivo transformar los requerimientos del diseño para el desarrollo del sistema a través de la aplicación de una arquitectura. Este flujo de trabajo se considera como un mecanismo importante dentro del proceso unificado de desarrollo del software porque permite la creación de la documentación necesaria para el soporte de la solución propuesta en esta investigación. Seguidamente, se presenta los diagramas de clases del diseño de cada uno de los casos de uso del sistema, así como los de interacción de los casos de uso más críticos y se explican además, los patrones arquitectónicos utilizados para la implementación del Subsistema Registro de Equipos Médicos.

3.1 Análisis de componentes de SIGICEM

Con el objetivo de agilizar el proceso de desarrollo en el Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM) y en aras de disminuir el tiempo utilizado para la implementación del Subsistema Registro de Equipos Médicos se ha concebido por el grupo de desarrollo del proyecto SIGICEM diferentes clases auxiliares que ofrecen la posibilidad de acelerar y facilitar la realización de las tareas comunes durante el proceso del software.

Se cuenta con funcionalidades que se encargan del trabajo con las interfaces visuales, a través de la creación de componentes incluidos en el framework de presentación ExtJS, gestionados de forma sencilla por el framework de trabajo Symfony. Igualmente, posee clases externas que han sido ajustadas a las pautas definidas para el SIGICEM; entre las cuales se destacan las que permiten la importación y exportación de documentos en formatos XLS y PDF.

3.2 Modelo de Análisis

Durante el modelado del análisis se refinan y estructuran los requisitos obtenidos con anterioridad, profundizándose en el dominio de la aplicación, lo que permitirá un mejor entendimiento del problema

planteado para la modelación de la solución mediante la creación de diagramas representados se por paquetes.

Diagramas de clases del análisis



Figura 13 Diagrama de clases del análisis. Caso de uso: Buscar equipo sin defectar



Figura 14 Diagrama de clases del análisis. Caso de uso: Crear reporte de equipo por rotura.

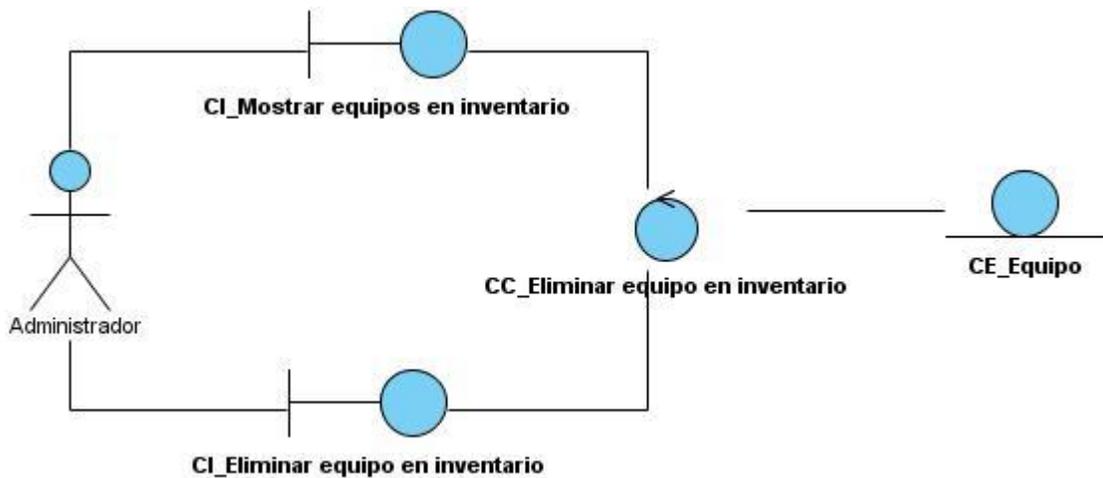


Figura 15 Diagrama de clases del análisis. Caso de uso: Eliminar equipo en inventario.



Figura 16 Diagrama de clases del análisis. Caso de uso: Mostrar listado de equipos en el origen.

Diagramas de interacción. Comunicación



Figura 17 Diagrama de comunicación. Caso de uso: Buscar equipo sin defectar.

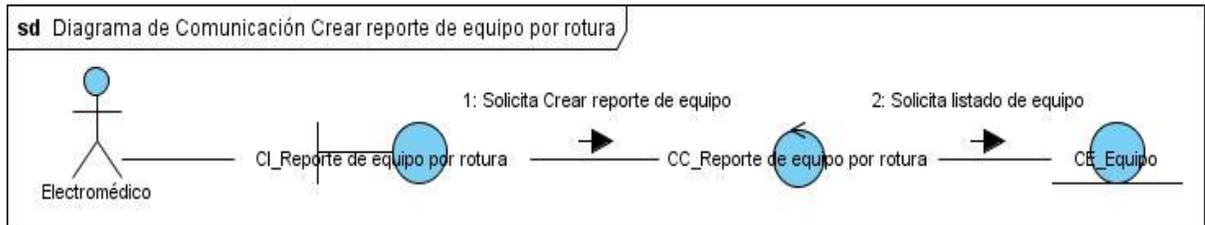


Figura 18 Diagrama de comunicación. Caso de uso: Crear reporte de equipo por rotura.

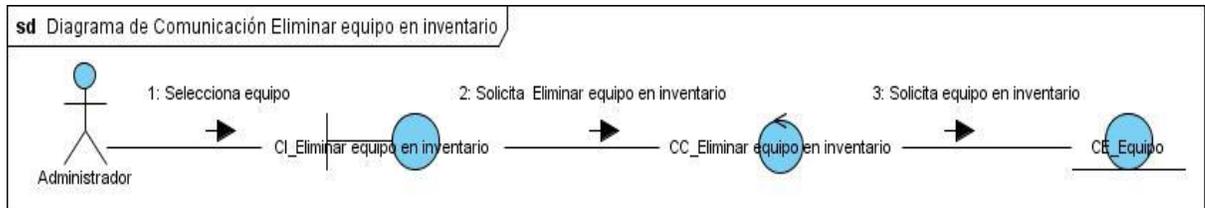


Figura 19 Diagrama de comunicación. Caso de uso: Eliminar equipo en inventario.

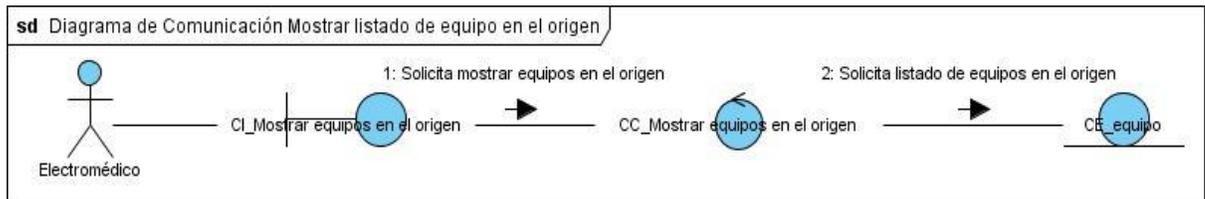


Figura 20 Diagrama de comunicación. Caso de uso: Buscar equipo sin defectar.

3.3 Modelo de Diseño

El modelo de diseño describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Este modelo se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica de la aplicación. Para una mejor calidad del diseño

fueron aplicados patrones de diseño durante la realización de los diagramas de clases permitiendo asignar las responsabilidades a los objetos y diseñar la colaboración entre ellos. (30)

3.4 Patrón arquitectónico

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software, este separa los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: Modelo, Vista y Controlador. Este modelo especifica un conjunto de subsistemas con sus responsabilidades y una serie de consideraciones a tener en cuenta para organizar los distintos componentes. Su popularidad se debe a que permite organizar aplicaciones grandes de manera mucho más fácil. Con su implementación se logra obtener una separación entre los distintos componentes que permite lograr mayor sencillez para crear distintas representaciones de los mismos datos. Incrementa en gran medida la reutilización y flexibilidad, así como que las soluciones desarrolladas suelen ser más escalables. (31)

En la solución propuesta es el MVC el patrón arquitectónico empleado, como se puede deducir de su nombre, consiste en separar de la mejor forma posible las capas de:

- ◆ **Modelo:** contiene la lógica de negocio, la base de datos se encuentra almacenada en esta capa, en la que se engloban los datos y las funcionalidades. Se comporta de manera independiente a cualquier representación de salida y comportamiento de entrada, asegurando la integridad de los datos y permitiendo derivar nuevos.
- ◆ **Vista:** es la parte que utilizan los usuarios para interactuar con la aplicación, simplemente la página HTML que se muestra al usuario. Esta capa se centra en presentar al modelo en un formato adecuado para el intercambio con el usuario final. Los gestores de plantillas se incluyen dentro de esta.
- ◆ **Controlador:** se enfoca en realizar las llamadas necesarias al modelo para capturar los datos y devolverlos a la vista para que en esta sean mostrados al usuario. Este responde a eventos,

usualmente acciones del usuario e invoca cambios en el modelo. En resumen, se encarga de asimilar las peticiones del usuario y realizar los cambios apropiados en el modelo o en la vista.

En la siguiente figura se observa la interacción que se efectúa entre las distintas capas, específicamente en el framework Symfony que es el que se emplea en la presente investigación:

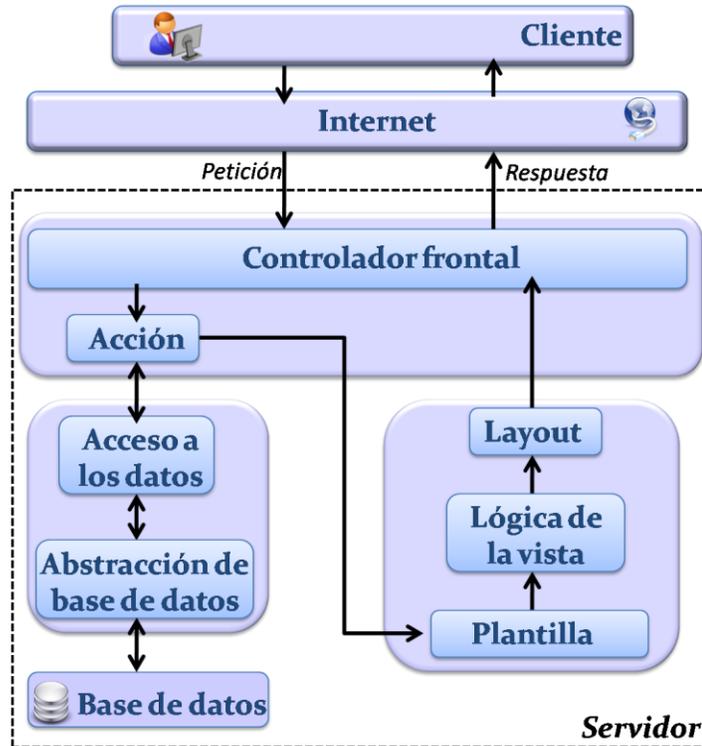


Figura 21 Comportamiento del MVC en Symfony.

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.

Propel, como ORM, se encarga de generar de forma automática las clases de la capa del modelo en dependencia de la estructura de datos de la aplicación, creando métodos especiales para acceder y modificar los datos. La abstracción de la base de datos se gestiona de manera transparente para el programador, con la utilización Objetos de Datos de PHP (PDO, PHP Data Objects), razón por la cual es posible migrar de sistema gestor de bases de datos. Por su parte la lógica de la vista se puede transformar en un archivo de configuración sencillo, sin necesidad de que esto implique grandes cambios, debido a que generalmente, el layout es global en toda la aplicación o en un grupo de páginas. La plantilla sólo se encarga de dar salida a las variables definidas en el controlador.

El controlador tiene como principal objetivo el manejo de las peticiones del usuario, de la seguridad, cargar la configuración de la aplicación y otras tareas. Este se genera de forma automática, dividiéndose, normalmente en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página. Este ofrece un punto de entrada único para toda la aplicación, por lo que en caso de ser necesario impedir el acceso a la aplicación, solamente se debe editar el script correspondiente al mismo. El controlador frontal y el layout, generalmente son comunes para todas las acciones de la aplicación. (32)

3.5 Patrones de diseño empleados

Los patrones de diseño no son más que la descripción de un problema y la solución del mismo, de forma que se pueda utilizar en diferentes contextos dando respuesta a interrogantes comunes. Los patrones *GRASP* (Patrones de Software para la Asignación General de Responsabilidades) tienen una importante utilidad en el diseño realizado. Los utilizados en la realización del presente trabajo son los siguientes:

- ◆ **Bajo Acoplamiento:** es el encargado de disminuir la dependencia de una clase con las demás. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño, estimulando asignar una responsabilidad de modo que su colocación no incremente el acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad.

- ◆ **Alta Cohesión:** es un principio que se debe tener presente en todas las decisiones de diseño: es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Este patrón evita asignar demasiadas responsabilidades a las clases. Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización.
- ◆ **Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. Capítulo 3: Diseño del Sistema.
- ◆ **Experto:** es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos, con este se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil manteniendo. El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clase "sencillas" y más cohesivas que son más fáciles de comprender y de mantener. Así se brinda soporte a una alta cohesión. (33)
- ◆ **Controlador:** Este se basa en la existencia de un intermediario entre las páginas clientes y el algoritmo que responde a las peticiones realizadas por estas. La existencia del controlador frontal, 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, siendo 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.

- ◆ **Decorador:** Symfony presenta el denominado archivo layout.php o también conocido como plantilla global, en la que convergen todos los elementos comunes a cada una de las páginas del sistema en construcción. Este fichero se complementa con las plantillas, decorándolas y obteniéndose la interfaz final que será mostrada al usuario. (34)

Diagramas de clases del diseño

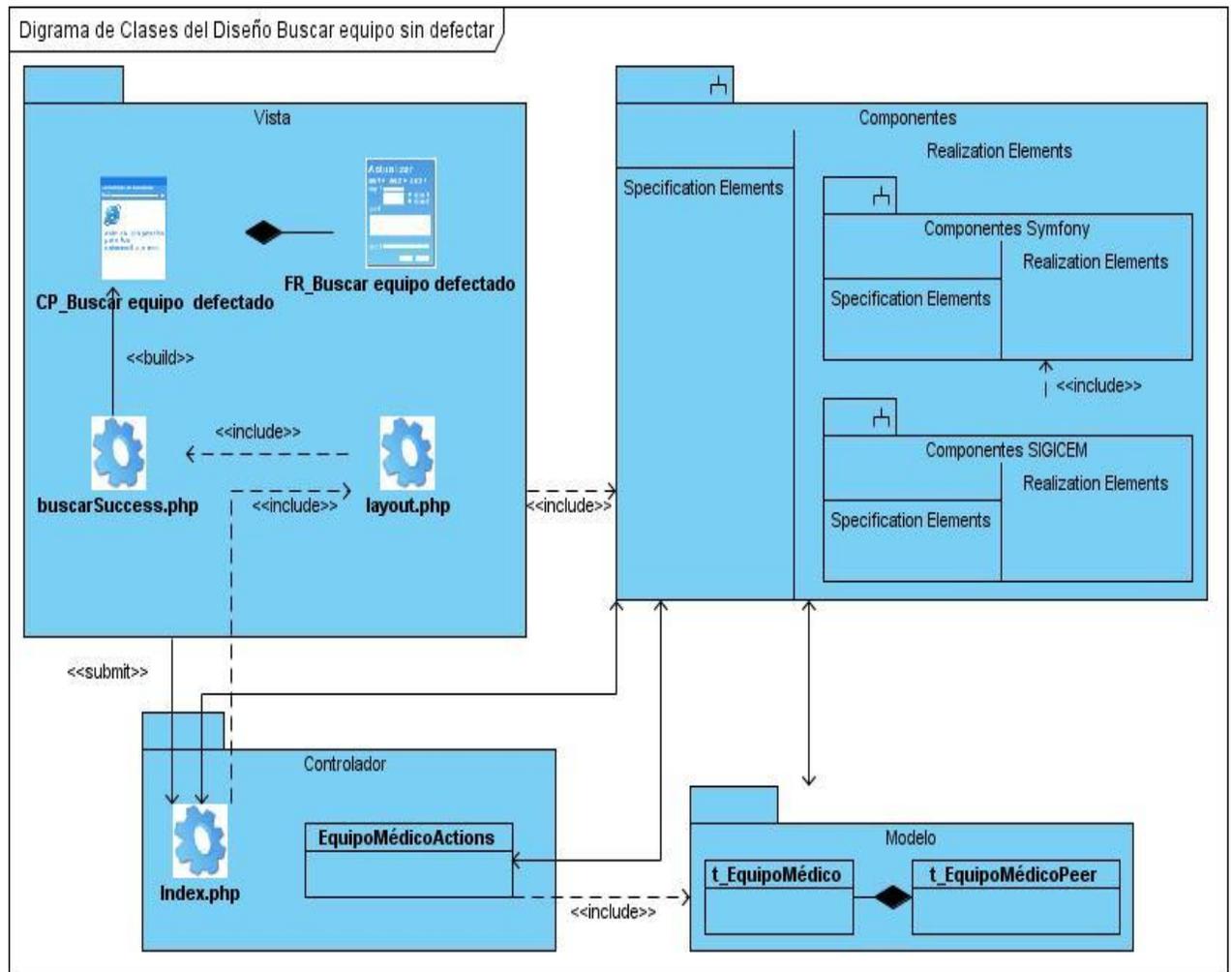


Figura 22 Diagrama de clases del diseño. Caso de uso: Buscar equipo sin defectar.

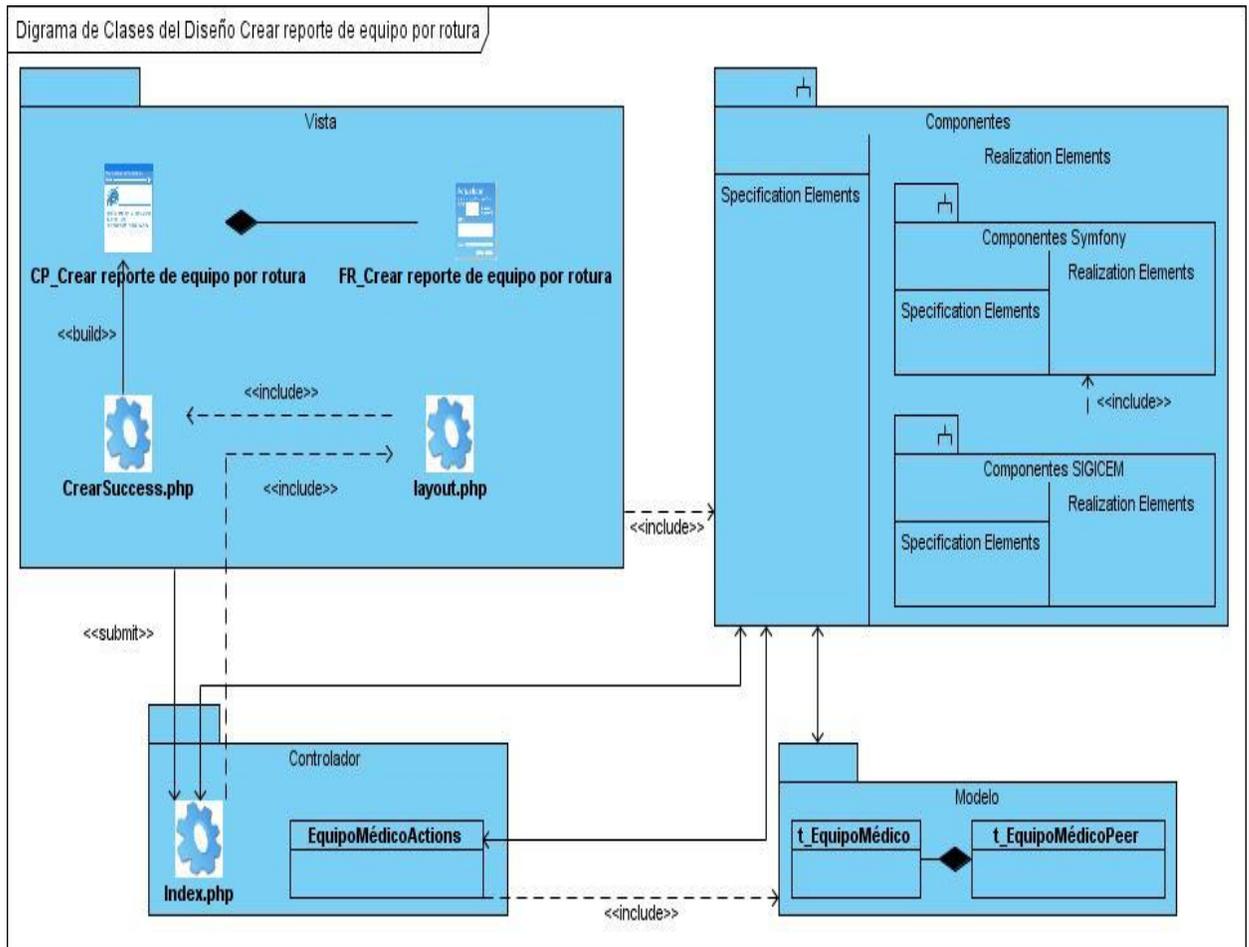


Figura 23 Diagrama de clases del diseño. Caso de uso: Crear reporte de equipo por rotura.

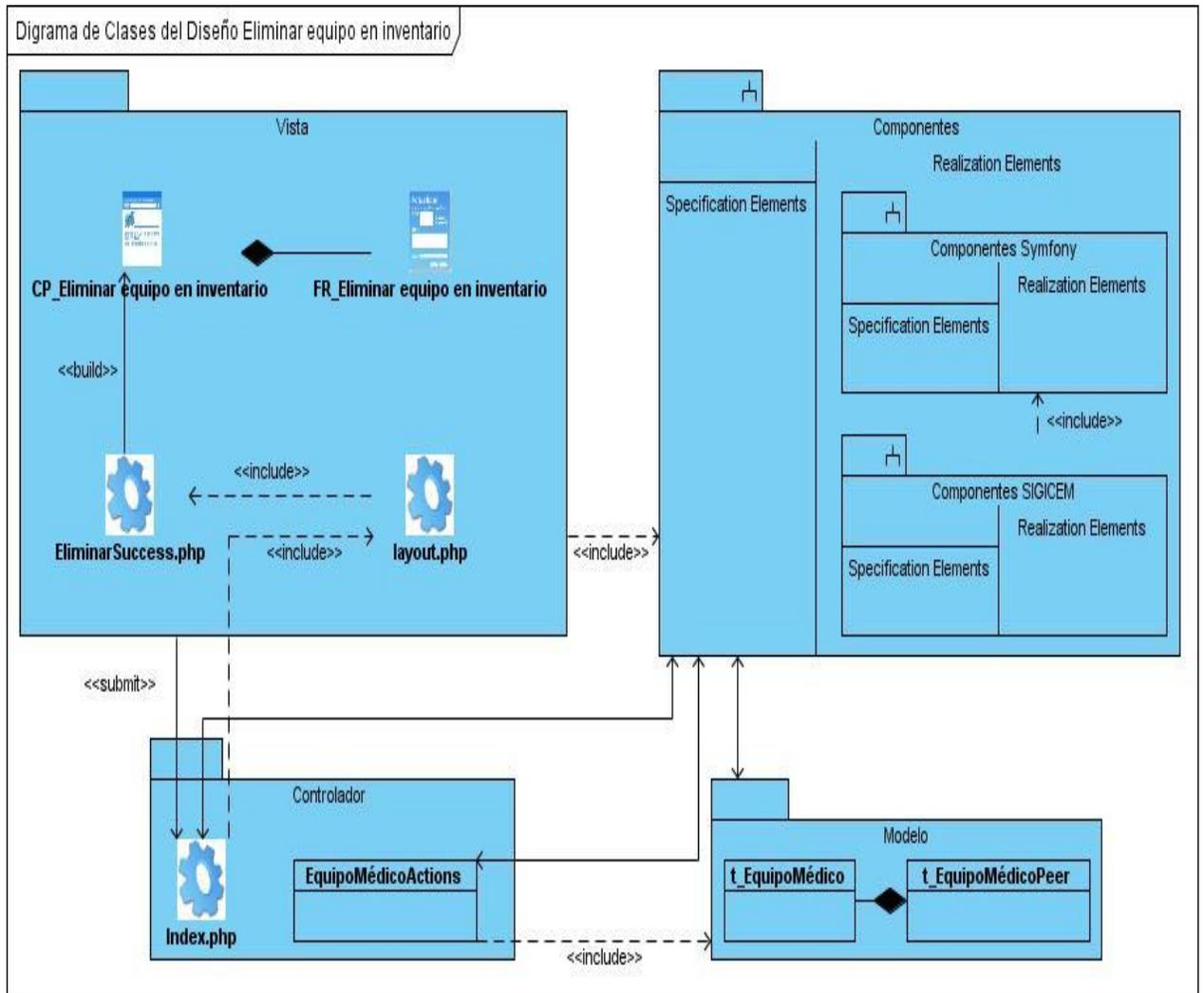


Figura 24 Diagrama de clases del diseño. Caso de uso: Eliminar equipo en inventario.

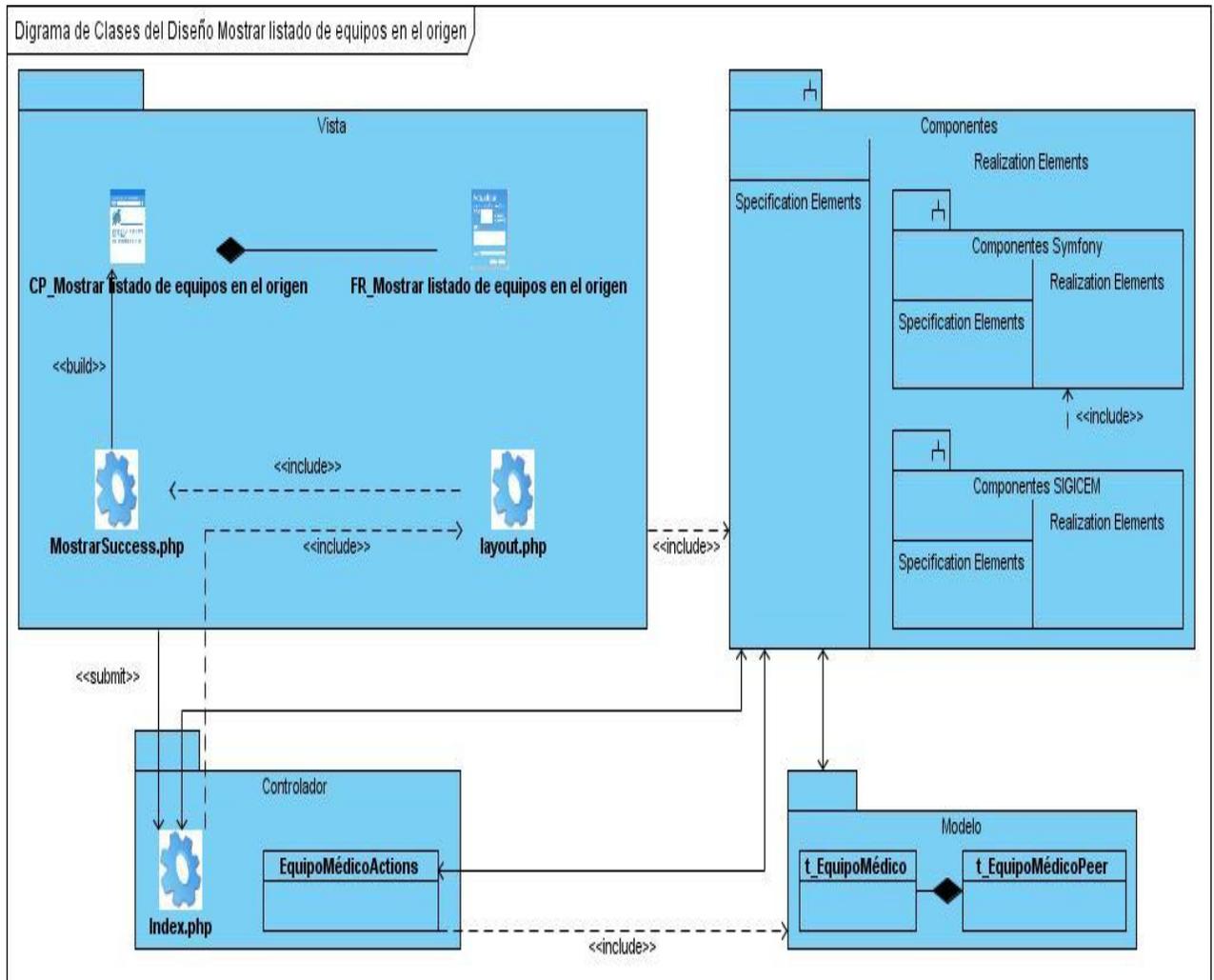


Figura 25 Diagrama de clases del diseño. Caso de uso: Mostrar listado de equipo en el origen.

Descripción de las clases del diseño

Nombre de la Clase:	t_EquipoMédicoPeer
Descripción:	Contiene los métodos estáticos para obtener registros de la base de datos. Se debe tener en cuenta que esta es una representación generalizada de cualquiera de las tablas de la base de datos.

Nombre de la Clase:	t_EquipoMédico
Descripción:	Representación Orientada a Objetos (OO) de los registros de la base de datos, a través de la cual es posible el tratamiento en forma de objeto de las entidades. Se debe tener en cuenta que esta es una representación generalizada de cualquiera de las tablas de la base de datos.
Nombre de la Clase:	EquipoMédicoActions
Descripción:	Modela el comportamiento específico, contiene la lógica de la aplicación del subsistema registro de equipos médicos, 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.
Nombre de la Clase:	indexSuccess.php
Descripción:	Contiene el código de presentación, siendo el resultado de una acción a la que previamente está asociada por el nombre seguido del resultado.
Nombre de la Clase:	layout.php
Descripción:	Contiene el código HTML común a todas las páginas de la aplicación para no tener que repetirlo en cada página. Esta es también conocida como plantilla global.
Nombre de la Clase:	editSuccess.php
Descripción:	Contiene el código de presentación, siendo el resultado de una acción a la que previamente está asociada por el nombre seguido del resultado.
Nombre de la Clase:	sigicem.php
Descripción:	Punto de entrada único de toda la aplicación, denominada controlador frontal, todas las peticiones realizadas son gestionadas por él. 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.

Nombre de la Clase:	CP_Eliminar_equipo_inventario
Descripción:	Representa el acceso del usuario a la interfaz de eliminar un equipo médico del inventario, esta se encarga de atender todas las selecciones y peticiones que pueda realizar un determinado usuario.
Nombre de la Clase:	fr_Eliminar_equipo_inventario
Descripción:	Representa la entrada y salida de datos en el sistema, a través de esta el usuario podrá introducir los datos necesarios para la posterior eliminación del equipo médico del inventario de la Unidad de Salud.

Tabla 13. Descripción de las clases empleadas en los diagramas de clases del diseño.

En el capítulo se realizó el modelado de análisis y diseño del Subsistema de Registro de Equipo Médicos a través de la representación de diagramas de clases, que facilitan la comprensión de la estructura del sistema a implementar. Asimismo, se identificaron y explicaron los patrones de diseño utilizados para el desarrollo del software.

Capítulo 4: Implementación del sistema

El trabajo en este capítulo parte del resultado del análisis y diseño obtenidos en el capítulo anterior y se procederá describir el modelo de implementación con el objetivo de definir la estructura y organización de la aplicación, además de implementar las clases y subsistemas encontrados durante el diseño representado en los diagramas de componentes y despliegue.

4.1 Modelo de datos

En el proceso y construcción de todo sistema informático, el diseño de la Base de Datos ocupa un lugar importante, definiéndose como un proceso relativamente independiente dentro del diseño del sistema y compuesto por una serie de etapas. Una de esas etapas es la formación del modelo Entidad Relación, definiéndose de la siguiente manera: el modelo Entidad-Relación puede ser usado como una base para una vista unificada de los datos, adoptando el enfoque más natural del mundo real que consiste en entidades e interrelaciones. Este modelo está basado en una percepción del mundo real que consta de un conjunto de objetos básicos llamados entidades con sus atributos y de las interrelaciones que existen entre estos objetos ([VER ANEXO 1](#)).

4.2 Modelo de implementación

El modelo de implementación es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros de código fuente, y otros tipos de ficheros necesarios para la implantación y despliegue del sistema. Describe también, como se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen de los componentes unos de otros. (34)

Diagrama de componentes

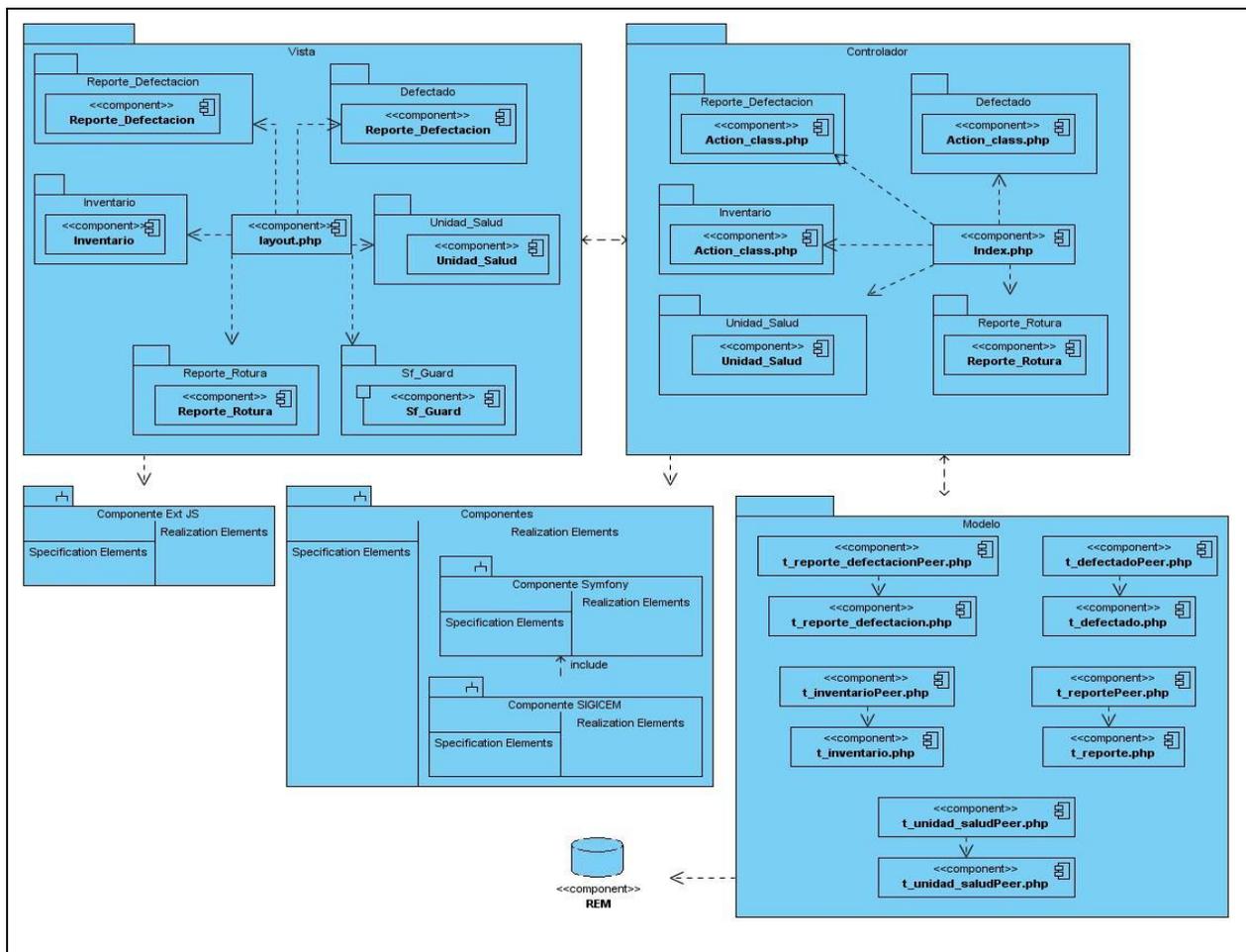


Figura 27 Diagrama de componentes

4.3 Tratamiento de errores

El tratamiento de errores es un paso fundamental para una buena elaboración del sistema, debido a que garantiza la integridad y confidencialidad de la información que se maneja en él.

Realizar este proceso permite evitar que estas situaciones indeseadas se conviertan en problemas mayores que afecten la funcionalidad de la aplicación. Con la validación de los datos enviados por los usuarios al servidor se evitan muchos de estos errores; la validación de formularios se puede realizar en el lado del servidor y en el lado del cliente. En el servidor es obligatoria para no corromper la base de datos con datos incorrectos y en el lado del cliente es opcional, pero mejora la experiencia de usuario.

Para minimizar estos errores se cuenta con cuadros de opción y menú de selección que facilita la entrada de datos por parte del usuario al sistema, así como los cargados por defecto. (35)

The screenshot shows a web application window titled "Modificar equipo en inventario". It contains several dropdown menus for location and equipment details, a table of programs, and "Guardar" and "Cancelar" buttons. The "Departamento" and "Servicios" dropdowns are highlighted with red boxes.

Programas	Eliminar
defecto1	X
defecto2	X
defecto3	X
defecto4	X

Figura 28 Ejemplo de cuadro de selección utilizado.

4.4 Seguridad

La seguridad e integridad del sistema se garantizan una vez que se creen los usuarios con los permisos necesarios para la gestión de la información en la base de datos del sistema.

Para lograr la seguridad total del subsistema propuesto se realizará la autenticación del mismo mediante el SAAA, de esta manera un usuario obtendrá los permisos necesarios para acceder al mismo. También el sistema de gestor de base datos debe mantenerse en un lugar restringido y asegurado.

4.5 Modelo de despliegue

Un diagrama de despliegue muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. A continuación se muestra una propuesta de despliegue que es como se distribuirá el subsistema. (36)

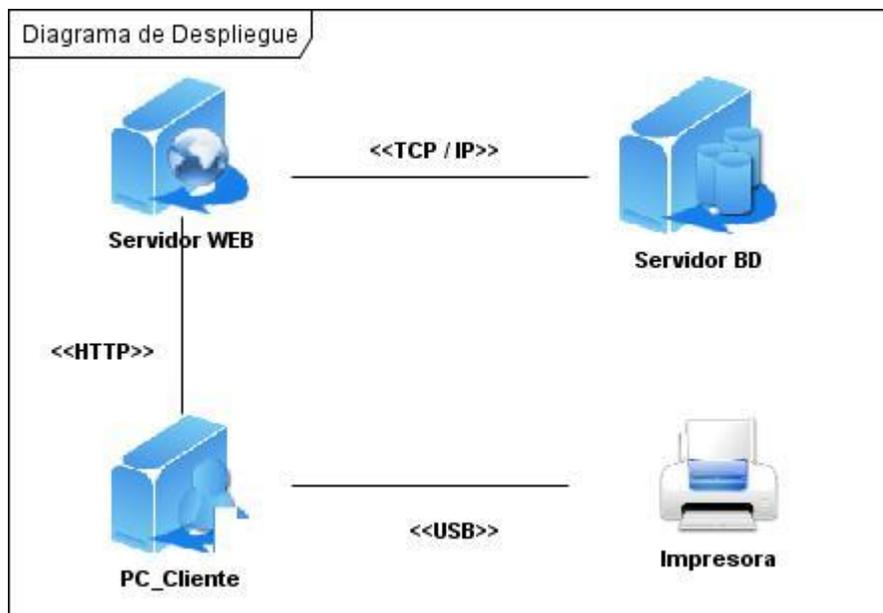


Figura 28 Diagrama de despliegue del sistema.

En el presente capítulo se ha desarrollado el proceso de implementación del Subsistema de registro de equipos médicos para el sistema informatizado de salud, obteniéndose un producto con las funcionalidades requeridas por el cliente. Se mostró la relación existente entre todos los nodos que

conforman el sistema, y su relación con la base de datos mediante el Diagrama de Componentes ya que este permite dividir el sistema en partes más entendibles por los programadores. De esta manera se brinda a los usuarios del sistema una solución funcional de fácil interacción para la visualización de la información.

Conclusiones

Al culminar el presente trabajo de diploma, se ha cumplido con el objetivo general y las tareas propuestas, por lo que se han obtenido los siguientes resultados:

- ◆ El estudio y análisis previo sobre el proceso de gestión de la información en el Sistema Informatizado de Salud, permitió resaltar la necesidad de llevar a cabo la investigación.
- ◆ Se aplicaron las herramientas y tecnologías propuestas por el Centro de Informática Médica (CESIM) y el Departamento Sistemas de Apoyo a la Salud (SAS) obteniéndose el ambiente de desarrollo y arquitectura del sistema.
- ◆ Se confeccionaron los artefactos pertenecientes a las Disciplinas de Trabajo: Modelado de Negocio, Gestión de Requerimientos, Análisis y Diseño, Implementación, siguiendo el Proceso Unificado de Desarrollo, lo que permitió conformar el expediente y grupo de plantillas tipo definidas por el Centro de Calidad para Soluciones Tecnológicas (Calisoft).
- ◆ Se diseñaron los prototipos de Interfaz de Usuario lo que permitió obtener y validar los requerimientos esenciales, manteniendo abiertas, las opciones de implementación.
- ◆ Se obtuvo el subsistema que es un producto que permite el registro de equipos médicos para el Sistema Informatizado de Salud

Recomendaciones

Al concluir la investigación se recomienda:

- ◆ Integrar al subsistema Registro de Equipos Médicos el consumo de datos de los servicios web pertenecientes al Sistema Informatizado de Salud.

Referencias bibliográficas

1. **Hernández, Rolando Alfredo y Coello, Zayda.** El paradigma cuantitativo de la investigación científica. *El paradigma cuantitativo de la investigación científica*. s.l. : Editorial Universitaria, 2002.
2. **Sánchez, E.Rodríguez M. y A. Miguel.** LA PRAXIS EN LA GESTIÓN DE EQUIPOS MÉDICOS. [En línea] [Citado el: 2 de 10 de 2010.] http://www.hab2003.sld.cu/Articles/T_0069.pdf.
3. Gestion-de-la-informacion. [En línea] [Citado el: 8 de 10 de 2010.] <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>.
4. Información para la gestión. [En línea] [Citado el: 15 de 10 de 2010.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.
5. **Carlos, Núñez Contreras y Reyes, David Castro.** Sistema informático de gestión del equipamiento médico mediante dispositivos móviles. [En línea] [Citado el: 25 de 10 de 2010.] <http://rosario2009.sabi.org.ar/uploadsarchivos/p64.pdf>.
6. Idem 4. [En línea]
7. Sistema de mantenimiento asistido por computadores. [En línea] [Citado el: 28 de 10 de 2010.] <http://antonio-miguel-cruz.qapacity.com/servicios-de-gestion/18528/sistema-de-mantenimiento-asistido-por-computadores/>.
8. <http://dspace.ups.edu.ec/bitstream/123456789/828/4/CAPITULO%204.pdf>. [En línea] [Citado el: 6 de 11 de 2010.] <http://dspace.ups.edu.ec/bitstream/123456789/828/4/CAPITULO%204.pdf>.
9. Grupo Industrial: Advantage Tecnología. [En línea] [Citado el: 15 de 11 de 2010.] http://www.nom-144.com.mx/equipos_control.html.
10. Idem 8. [En línea]
11. Sistema-automatizado-cubano-para-el-control-de-equipos-medicos. [En línea] [Citado el: 13 de 11 de 2010.] <http://informatica2009.sld.cu/Members/dhernandez00242/sacem-sistema-automatizado-cubano-para-el-control-de-equipos-medicos-1/>.

12. Idem **10**. [En línea]
13. El Proceso Unificado de Desarrollo de Software. [En línea] [Citado el: 29 de 11 de 2010.] http://www.wepapers.com/Papers/122324/EI_Proceso_Unificado_de_Desarrollo_de_Software_%28RUP%29.
14. Idem **12**. [En línea]
15. Modelado-sistemas-UML. [En línea] [Citado el: 30 de 11 de 2010.] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
16. Introduccion_a_uml. [En línea] [Citado el: 4 de 12 de 2010.] http://www.programacion.com/articulo/introduccion_a_uml_181.
17. UML-Lenguaje-Unificado-de-Modelado. [En línea] [Citado el: 5 de 12 de 2010.] http://ww.taringa.net/posts/info/6858927/UML_-Lenguaje-Unificado-de-Modelado.html.
18. sistemas-gestores-bases-datos. [En línea] [Citado el: 16 de 12 de 2010.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
19. Bases-datos-postgresql. [En línea] [Citado el: 17 de 12 de 2010.] <http://www.dataprix.com/bases-datos-postgresql>.
20. Visual Paradigm for UML. [En línea] [Citado el: 20 de 12 de 2010.] <http://translate.google.com/translate?hl=es&sl=en&u=http://www.visual-paradigm.com/&ei=hmpsTaLcAdGutwfs27zIBQ&sa=X&oi=translate&ct=result&resnum=1&ved=0CC4Q7gEwAA&prev=/search%3Fq%3DVisual>.
21. Visual-paradigm-for-uml. [En línea] [Citado el: 27 de 12 de 2010.] <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
22. Netbeans ya incluye soporte para symfony. [En línea] [Citado el: 16 de 12 de 2010.] <http://www.symfony.es/2009/10/05/netbeans-ya-incluye-soporte-para-symfony/>.
23. Idem **20**. [En línea]

24. symfony_en_pocas_palabras. [En línea] [Citado el: 26 de 11 de 2010.] http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.
25. Symfony. [En línea] [Citado el: 9 de 12 de 2010.] <http://www.buenastareas.com/ensayos/Symfony/1011314.html>.
26. Extjs-lo-bueno-lo-malo-y-lo-feo. [En línea] [Citado el: 22 de 11 de 2010.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
27. Idem **22**. [En línea]
28. Idem **13**. [En línea]
29. Idem **13**. [En línea]
30. Idem **13**. [En línea]
31. Idem **24**. [En línea]
32. Idem **24**. [En línea]
33. PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad).Parte II. [En línea] [Citado el: 28 de 11 de 2010.] <http://jorgesaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>.
34. Idem **25**. [En línea]
35. MODULO 2:Tema 12:Modelo de Implementación:Diagramas de Componentes y Despliegue. [En línea] [Citado el: 30 de 11 de 2010.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
36. Manejo de errores en capa de presentación web. [En línea] [Citado el: 3 de 12 de 2010.] <http://privmario.wordpress.com/2009/10/28/manejo-de-errore>.
37. Una Introducción al UML El Modelo Físico. [En línea] [Citado el: 5 de 12 de 2010.] http://www.craftware.net/es/descargas/modelo_fisico.pdf.

Bibliografía

Bases-datos-postgresql. [En línea] [Citado el: 17 de 12 de 2010.] <http://www.dataprix.com/bases-datos-postgresql>.

Carlos, Núñez Contreras y Reyes, David Castro. Sistema informático de gestión del equipamiento médico mediante dispositivos móviles. [En línea] [Citado el: 25 de 10 de 2010.] <http://rosario2009.sabi.org.ar/uploadsarchivos/p64.pdf>.

El Proceso Unificado de Desarrollo de Software. [En línea] [Citado el: 29 de 11 de 2010.] http://www.wepapers.com/Papers/122324/El_Proceso_Unificado_de_Desarrollo_de_Software_%28RUP%29.

Extjs-lo-bueno-lo-malo-y-lo-feo. [En línea] [Citado el: 22 de 11 de 2010.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.

Gestion-de-la-informacion. [En línea] [Citado el: 8 de 10 de 2010.] <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>.

Grupo Industrial: Advantage Tecnología. [En línea] [Citado el: 15 de 11 de 2010.] http://www.nom-144.com.mx/equipos_control.html.

Hernández, Rolando Alfredo y Coello, Zayda. El paradigma cuantitativo de la investigación científica. *El paradigma cuantitativo de la investigación científica*. s.l. : Editorial Universitaria, 2002.

Información para la gestión. [En línea] [Citado el: 15 de 10 de 2010.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.

Introduccion_a_uml. [En línea] [Citado el: 4 de 12 de 2010.] http://www.programacion.com/articulo/introduccion_a_uml_181.

Manejo de errores en capa de presentación web. [En línea] [Citado el: 3 de 12 de 2010.] <http://privmario.wordpress.com/2009/10/28/manejo-de-errore>.

Modelado-sistemas-UML. [En línea] [Citado el: 30 de 11 de 2010.] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.

MODULO 2:Tema 12:Modelo de Implementación:Diagramas de Componentes y Despliegue. [En línea] [Citado el: 30 de 11 de 2010.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.

Netbeans ya incluye soporte para symfony. [En línea] [Citado el: 16 de 12 de 2010.] <http://www.symfony.es/2009/10/05/netbeans-ya-incluye-soporte-para-symfony/>.

PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad).Parte II. [En línea] [Citado el: 28 de 11 de 2010.] <http://jorgesaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>.

Sánchez, E.Rodríguez M. y A. Miguel. LA PRAXIS EN LA GESTIÓN DE EQUIPOS MÉDICOS. [En línea] [Citado el: 2 de 10 de 2010.] http://www.hab2003.sld.cu/Articles/T_0069.pdf.

Sistema de mantenimiento asistido por computadores. [En línea] [Citado el: 28 de 10 de 2010.] <http://antonio-miguel-cruz.qapacity.com/servicios-de-gestion/18528/sistema-de-mantenimiento-asistido-por-computadores/>.

Sistema-automatizado-cubano-para-el-control-de-equipos-medicos. [En línea] [Citado el: 13 de 11 de 2010.] <http://informatica2009.sld.cu/Members/dhernandez00242/sacem-sistema-automatizado-cubano-para-el-control-de-equipos-medicos-1/>.

Sistemas-gestores-bases-datos. [En línea] [Citado el: 16 de 12 de 2010.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

Symfony. [En línea] [Citado el: 9 de 12 de 2010.] <http://www.buenastareas.com/ensayos/Symfony/1011314.html>.

Symfony_en_pocas_palabras. [En línea] [Citado el: 26 de 11 de 2010.] http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.

UML-Lenguaje-Unificado-de-Modelado. [En línea] [Citado el: 5 de 12 de 2010.] http://ww.taringa.net/posts/info/6858927/UML_-Lenguaje-Unificado-de-Modelado.html.

Una Introducción al UML El Modelo Físico. [En línea] [Citado el: 5 de 12 de 2010.] http://www.craftware.net/es/descargas/modelo_fisico.pdf.

Visual Paradigm for UML. [En línea] [Citado el: 20 de 12 de 2010.]
<http://translate.google.com/translate?hl=es&sl=en&u=http://www.visual-paradigm.com/&ei=hmpsTaLcAdGutwfs27zIBQ&sa=X&oi=translate&ct=result&resnum=1&ved=0CC4Q7gEwAA&prev=/search%3Fq%3DVisual>.

Visual-paradigm-for-uml. [En línea] [Citado el: 27 de 12 de 2010.]
<http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.

Glosario de términos

BD: Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

CESIM: Centro de Informática Médica.

CUN: Caso de uso del negocio.

CUS: Caso de uso del sistema.

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.

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).

HTTP: (Hypertext Transfer Protocol). Se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que se conoce como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

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

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).

PostgreSQL: Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD.

SACCEM: Sistema Automatizado Cubano para el Control de Equipos Médicos.

SAS: Departamento de Sistemas de Apoyo a la Salud perteneciente al CESIM.

SGBD: Gestores de Bases de Datos. Permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

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

SNS: Sistema Nacional de Salud.

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

SYSMANCOR: Sistema de mantenimiento asistido por computador.

RF: Requerimientos 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).

TIC: Tecnologías de la Informática y las Comunicaciones.

UCI: Universidad de Ciencias Informáticas.

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

Anexo1

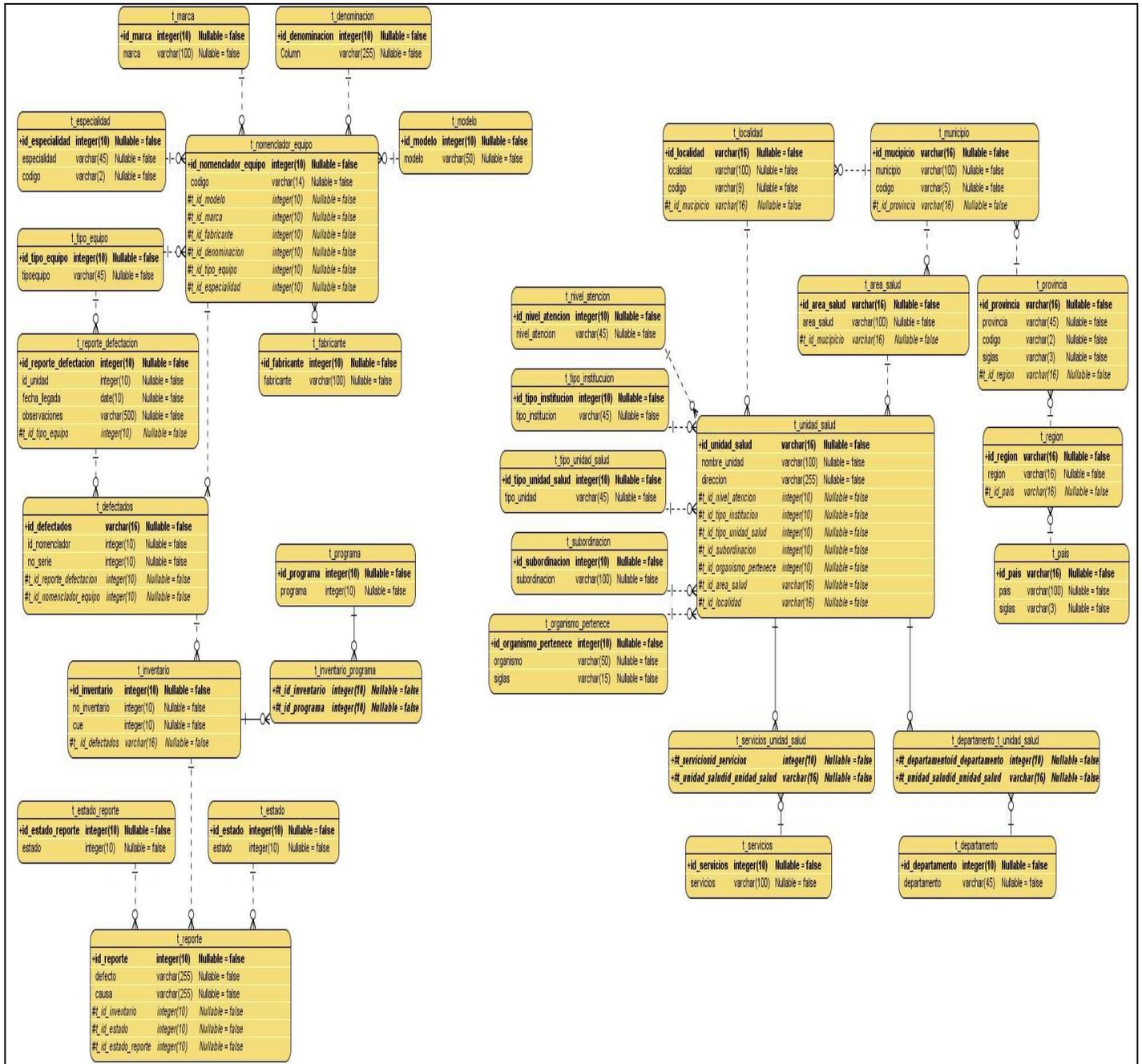


Figura 26 Modelo de datos.