

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Facultad 2

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

**Módulo de administración, configuración y seguridad del Sistema
de Gestión para la Ingeniería Clínica y Electromedicina v2.0**

AUTORES

Ana Fé del Toro Fonseca

Yordan Peña Dieguez

TUTORES

Ing. Ranniel Rivero Sevilla

MSc. Zoila Esther Morales Tabares

La Habana, Junio de 2014

“Año 56 de la Revolución”



“...el futuro de nuestra Patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia...”

Fidel Castro Ruz

DATOS DE CONTACTO DE LOS TUTORES

Ing. Ranniel Rivero Sevilla: Ingeniero en Ciencias Informáticas y profesor asistente graduado del curso 2007-2008. Fue líder de desarrollo del proyecto SACCEM durante 2 años, luego pasó a ser jefe de proyecto del mismo. Ha impartido las asignaturas de Práctica Profesional (PP), Inteligencia Artificial (IA), Historia de la Informática (HI), Programación Web (P4) y asignaturas del Segundo Perfil (SP). En varias ocasiones ha participado en diferentes eventos de carácter nacional e internacional. En el curso 2008-2009 formó parte de un tribunal en 6 tesis de grado y ha tutorado más de 10 tesis de grado incluyendo una en la CUJAE. Se desempeñó como jefe de desarrollo por 2 años en un proyecto desarrollado para el Centro Nacional de Electromedicina perteneciente al Departamento Sistema de Apoyo a la Salud del Centro de Informática Médica (CESIM) que posteriormente pasó a ser del departamento de Sistemas para la Salud (SS) en el que actualmente ocupa el rol de jefe de proyecto del Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM).

Correo Electrónico: rrivero@uci.cu

MSc. Zoila Esther Morales Tabares: Graduada de Ingeniera en Ciencias Informáticas en el curso 2007-2008. Profesora de la Facultad 2, adjunta a la producción, perteneciente al Departamento Sistemas Especializados en Salud, laborando en el proyecto: Sistema de Gestión para Ingeniería Clínica y Electromedicina como planificadora. Imparte clases de Administración de Empresas y Contabilidad y Finanzas. Ha sido tutora de 3 tesis en Opción del Título de Técnico en Informática en la Enseñanza Media: Politécnico “Simón Bolívar de Las Tunas”, las cuales obtuvieron la calificación máxima. Se desempeñó como cotutora en el curso 2010-2011 de una tesis en la cual se alcanzó la máxima calificación y oponente de una tesis, que de igual forma obtuvo excelentes resultados. Realizó en el mismo curso la discusión de Tesis de Maestría en Informática Aplicada, provincia Camagüey, obteniendo la calificación de Excelente, así como excelentes resultados en todos los postgrados cursados en la Maestría. Ha participado en numerosos eventos tanto nacionales como internacionales en los que ha tenido importantes publicaciones.

Correo Electrónico: zemorales@uci.cu

AGRADECIMIENTOS

Ana Fé Del Toro Fonseca:

Agradezco la realización de este trabajo a:

A mi Mamá y hermanos: Liosbel Liens Fonseca y Lisbet Liens Fonseca porque confiaron en mí y me han dado la fuerza para seguir adelante cada día.

Mi hermana y amiga Marien Díaz Ruíz por su dedicación, cariño y comprensión, que en conjunto con sus padres Longina Ruíz y Félix Díaz conformaron una linda familia en la que recibí todo tipo de apoyo y en la me pude regocijar al estar lejos de los míos. Gracias a los tres por existir.

A mis dos tutores y en especial a Ranniel Rivero Sevilla porque su ayuda y apoyo incondicional estuvieron presentes desde el primer momento en que me dieron el tema de tesis. Mejores tutores que ellos no puedo pedirlos. Gracias de veras.

A este tribunal y a los de los primeros cortes, porque sus críticas constructivas posibilitaron que obtuviera la máxima calificación.

Y por último y no menos importante: A mis compañeros de la universidad, por su ayuda y presencia.

Yordan Peña Dieguez:

Agradezco la realización de este trabajo a:

Mi familia, por su apoyo incondicional durante los cinco años de la carrera universitaria.

Mis tutores, por su ayuda y aliento para que cada día me supere más como profesional.

El tribunal y oponente, porque sus críticas constructivas hicieron posible la correcta realización de este trabajo.

A mis amistades de la universidad, por su ayuda y presencia.

Por último, y no menos importante, mi compañera de tesis y amiga Ana Fé Del Toro Fonseca, por su comprensión y paciencia cada día de trabajo.

DEDICATORIA

Ana Fé Del Toro Fonseca:

Dedico este trabajo primero que nada a mi madre querida Virtudes Fonseca Martínez porque es la luz de mis ojos y la razón de mi vivir, y sé que nadie más que ella quería que yo lograra esta meta.

A mis hermanos varones porque más que ellos han sido padres y aunque no están presentes sé que están orgullosos de su hermanita.

Y

A Félix Díaz Quiñónes porque considero que es mi estrella de la suerte. Gracias por dejarme llamarte Papá y más que eso: gracias por serlo.

Yordan Peña Dieguez:

Primero, dedico este trabajo a mis padres por la educación y motivación que siempre me dieron, apoyándome y dando de sí sus máximos esfuerzos para que pudiera alcanzar mis metas, siendo una de ellas: Ingeniero en Ciencias Informáticas.

A mi hermanita querida que siempre me alentó aprender cada día más y a obtener buenos resultados.

A mí amada esposa que siempre ha estado a mi lado, en los buenos y malos momentos, dándome una pisca de felicidad cada día.

RESUMEN

El Sistema de Gestión para la Ingeniería Clínica y Electromedicina brinda actualmente la administración y seguridad del sistema con un nivel de detalle alto, pero no es lo suficientemente configurable para la navegación del usuario y además, los *frameworks* con que fue desarrollada su primera versión: Symfony 1.4.8 y Extjs 2.2, ya no contarán con soporte técnico, hecho que atenta contra la escalabilidad del proyecto. Debido a estos problemas se identificó la necesidad de implementar un nuevo núcleo para la arquitectura de SIGICEM. Con la nueva arquitectura se hizo necesario desarrollar el Módulo de administración, configuración y seguridad para el Sistema de Gestión para Ingeniería Clínica y Electromedicina v2.0, de manera tal que queden creadas las bases del software, para luego incorporarle los demás módulos.

De acuerdo a lo anterior, se asimilaron las tecnologías, herramientas y metodología definidas por el proyecto SIGICEM v2.0, entre las que se encuentran: como metodología, el Proceso Unificado Rational, apoyado en el Lenguaje Unificado de Modelado 2.1, sustentado por el Visual Paradigm 6.4 como herramientas CASE. Además, de la utilización de los *frameworks* Symfony 2 y Ext JS 4, soportado sobre una plataforma LAMP con la utilización de Linux como Sistema Operativo en su distribución Ubuntu 12.04, Servidor Web Apache en su versión 2.2, Gestor de Bases de Datos MySQL en su versión 5.3 y lenguaje de programación: PHP en su versión 5.3.10. La puesta en práctica de este módulo permitirá una correcta administración, configuración y seguridad de los usuarios, roles y permisos del sistema.

Palabras clave: administración, configuración, seguridad, sistema

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1. Conceptos fundamentales asociados a la investigación.....	5
1.2. Análisis de sistemas informáticos web para la administración, configuración y seguridad a nivel nacional e internacional	7
1.3. Análisis comparativo entre sistemas informáticos web analizados para la administración, configuración y seguridad a nivel nacional e internacional	9
1.3.1. Argumentación de la comparativa efectuada en la tabla 1	9
1.4. Metodologías, tecnologías y herramientas	11
1.4.1. Metodología de Desarrollo: Rational Unified Process (RUP)	11
1.4.2. Lenguaje Unificado de Modelado: Unified Modeling Language 2.1	11
1.4.3. Herramienta CASE	12
1.4.4. Symfony 2	12
1.4.5. ExtJS 4.....	13
1.4.6. Entorno Integrado de Desarrollo	13
1.4.7. Plataforma de Desarrollo: LAMP	14
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	16
2.1. Modelo de Dominio	16
2.1.1. Descripción del Modelo de Dominio	16
2.1.2. Definición de clases del Modelo de Dominio	17
2.1.3. Relación entre clases.....	17
2.2. Propuesta de solución.....	18
2.3. Análisis de requisitos de software	18

2.3.1.	Requisitos Funcionales.....	18
2.3.2.	Requisitos no Funcionales.....	19
2.4.	Actores del Sistema	20
2.5.	Diagrama de caso de uso del Sistema.....	21
2.6.	Descripción de casos de uso del Sistema (CUS).....	23
2.6.1.	Registrar usuario.....	23
2.6.2.	Eliminar Usuario.....	25
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....		27
3.1.	Modelo de análisis	27
3.1.1.	Diagramas de Clases del Análisis.....	28
3.1.2.	Diagramas de Interacción. Colaboración	29
3.2.	Modelo de Diseño	30
3.2.1.	Diagrama de Clase del Diseño.....	31
3.2.2.	Descripción de las Clases del Diseño.....	33
3.3.	Patrón arquitectónico	34
3.4.	Patrones de Diseño.....	35
3.5.	Modelo de Datos	39
3.5.1.	Descripción de las tablas del modelo de Datos	40
CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA.....		45
4.1.	Diagrama de despliegue	45
4.2.	Diagrama de componentes.....	46
4.3.	Tratamiento de errores.....	48
4.4.	Seguridad	50

CONCLUSIONES GENERALES.....	55
RECOMENDACIONES.....	56
REFERENCIAS BIBLIOGRÁFICAS	57
ANEXOS	64

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) han impulsado el desarrollo de la sociedad, permitiendo una evolución en la transmisión e intercambio de información, aportando conocimientos a la sociedad actual, a los cuales se pueden acceder sin importar barreras geográficas o limitaciones del tiempo. Estos elementos como el espacio y el tiempo condicionan la comunicación y vienen aparejado de la capacidad de interacción en el proceso. En los últimos años, la mayoría de los países han implementado proyectos, políticas o estrategias para promover el uso de las TIC y aprovechar los beneficios que estas ofrecen.

Cuba, a pesar de no ser un país desarrollado, ha dedicado una importante parte de sus recursos al desarrollo de la informática en diferentes esferas de la sociedad, específicamente en el sector de la salud (1). Una de las instituciones que contribuye a este desarrollo gracias a las TIC, es la Universidad de las Ciencias Informáticas (UCI), que además de ser un centro de estudios universitarios, tiene como uno de sus objetivos la informatización del país, por lo que trata de desarrollar al máximo la industria del software en la mayoría de las ramas de la sociedad, contribuyendo de esta forma con el avance económico de Cuba. Uno de los sectores en el que se tiene gran experiencia en el desarrollo y cooperación para el diseño de aplicaciones, es el sector de la Salud.

En la UCI, se cuenta con un Centro Informática Médica (CESIM), que se encarga de desarrollar productos y soluciones informáticas para el mejoramiento de la calidad de la salud pública cubana, lo que le permite estar muy ligada al Sistema Nacional de Salud (SNS). Uno de los proyectos que se desarrolla en dicho centro lleva por nombre Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM), solución eficaz, encargada de la informatización de todos los procesos de gestión tecnológica llevados a cabo en la red nacional de electromedicina del país, con el objetivo de garantizar la fiabilidad y actualización de la información referente al equipamiento médico instalado, para contribuir así, a elevar la calidad y eficiencia de los servicios técnicos dentro del SNS mediante la organización, planificación, control y análisis de las acciones que se ejecuten.

SIGICEM fue desarrollado en su primera versión con los frameworks Symfony 1.4.8 y Extjs 2.2, el mismo cuenta con una acorde administración y seguridad, pero no es lo suficiente configurable para el trabajo del usuario con el sistema. Por tal motivo, aunque el núcleo de SIGICEM es una solución viable, el equipo de proyecto decide desarrollar una segunda versión, actualizando los frameworks anteriormente utilizados a

Symfony 2 y Extjs 4, de manera que se puedan aprovechar las nuevas prestaciones y seguir el ciclo de desarrollo, actualización y soporte. Por tales motivos se determina que la nueva arquitectura definida para el proyecto, no cuenta con un mecanismo que garantice la administración, configuración y seguridad del sistema, teniendo en cuenta los nuevos requerimientos para la mejora de las prestaciones con las que se contaba anteriormente; que permita implementarle los módulos ya definidos y luego incorporarle los restantes según los nuevos procesos de negocio que se pretenden informatizar.

De acuerdo a la problemática planteada anteriormente se define como **problema a resolver**: ¿Cómo lograr que el Sistema de Gestión para la Ingeniería Clínica y Electromedicina en su versión 2.0 garantice la administración, configuración y seguridad?

Se tiene como **objeto de estudio**: Proceso de administración, configuración y seguridad en aplicaciones web; el cual delimita el **campo de acción**: Gestión de la administración, configuración y seguridad en el Sistema de Gestión para la Ingeniería Clínica y Electromedicina.

Se plantea como **objetivo general**: Desarrollar un módulo de administración, configuración y seguridad para el Sistema de Gestión para Ingeniería Clínica y Electromedicina v2.0.

Para dar cumplimiento al objetivo del presente trabajo, como guía se definieron las **siguientes tareas de la investigación**:

1. Determinación de los principales referentes teóricos del proceso de administración, configuración y seguridad en aplicaciones web, estableciendo similitudes con la investigación en curso.
2. Análisis de las particularidades y necesidades del proceso administración, configuración y seguridad del Sistema de Gestión para Ingeniería Clínica y Electromedicina v2.0, para identificar los principales referentes teóricos metodológicos del sistema.
3. Diseño de los artefactos correspondientes a las Disciplinas de Trabajo: Modelamiento del Negocio, Requisitos, Análisis y Diseño e Implementación, guiados por el Proceso Unificado de Desarrollo que permita la obtención de la documentación necesaria para el Módulo administración, configuración y seguridad del Sistema de Gestión para Ingeniería Clínica y Electromedicina v2.0.
4. Implementación de las funcionalidades identificadas en el proceso, según lo determinado en la especificación de requisitos de software.

En el desarrollo de la investigación se utilizaron un conjunto de métodos científicos, todos bajo la concepción dialéctico – materialista como método general.

El método científico de la investigación proporciona la orientación y dirección adecuada al trabajo del investigador, se convierte en el camino más corto para alcanzar los resultados esperados, y condiciona los nuevos conocimientos (3). Como parte de los **métodos teóricos** utilizados se encuentran:

- **Histórico – Lógico:** Utilizado para la determinación de antecedentes, tendencias y regularidades del objeto de estudio y el campo de acción.
- **Analítico – Sintético:** Empleado para la determinación de las generalidades y especificidades en el objeto de estudio y el campo de acción; así como en la fundamentación teórica y elaboración del diseño del Módulo administración, configuración y seguridad del SIGICEM v2.0.
- **Inductivo – Deductivo:** Permite adquirir conocimientos generalizados acerca de los posibles elementos de gestión, configuración y seguridad, así como determinar aquellos procesos más específicos implicados en los cambios que se necesitan para la nueva versión del SIGICEM.
- **Modelación:** Empleado durante el desarrollo del Módulo administración, configuración y seguridad del SIGICEM v2.0, específicamente en la etapa de análisis y diseño del software.

Por otra parte, como **métodos empíricos** se utilizaron:

- **Entrevista:** Método utilizado para obtener criterios importantes acerca de la administración, configuración y seguridad en el SIGICEM. Para ello fue necesario entrevistar a sus principales desarrolladores, con el objetivo de asimilar las características y los problemas de la actual investigación. Se realiza de forma no estructurada porque no se lleva a cabo en ella un cuestionario estricto y pueden variar las personas entrevistadas.
- **Observación científica:** Método utilizado en distintos momentos de la investigación en cuestión, para recopilar información necesaria de conceptos y funcionalidades, de forma tal que permita caracterizar el estado de la administración, configuración y seguridad en el SIGICEM v1.1, logrando así una visión general del sistema.

El Módulo de administración, configuración y seguridad para el Sistema de Gestión para la Ingeniería Clínica y Electromedicina v2.0 proveerá los siguientes beneficios:

- La gestión eficiente de las exigencias de seguridad de los sistemas de autenticación que den solución a problemas en el SIGICEM permitirá que se lleve a cabo el control total de los usuarios, roles y permisos del sistema.
- La protección a la información tratada en todo el SIGICEM no permitirá debilidad, corrupción e inconsistencia de la misma.
- La correcta gestión de la administración, configuración y seguridad en el módulo, posibilitará el fortalecimiento y control del SIGICEM v2.0.
- Elevado control y calidad de los procesos de gestión de usuarios y asignación de privilegios.

El documento de la investigación está estructurado de la siguiente forma:

Capítulo 1. Fundamentación teórica. Se efectúa un análisis de los principales referentes teóricos existentes a nivel internacional y nacional relacionados con la investigación en curso. Se muestra una breve descripción de los conceptos fundamentales asociados al tema de investigación. Asimismo, se describen las herramientas, tecnologías y metodologías empleadas en la solución propuesta.

Capítulo 2. Características del sistema. En el capítulo se especifican los requisitos funcionales y no funcionales asociados al Módulo de administración, configuración y seguridad del Sistema de Gestión para la Ingeniería Clínica y Electromedicina v2.0, quedando de forma explícita y detallada una representación y descripción de los casos de usos que tendrá el sistema. Además, se hace una descripción del modelo de dominio definido, así como se describe la solución propuesta.

Capítulo 3. Análisis y Diseño del Sistema. Como parte de la solución, se explican y modelan para cada caso de uso definido en el capítulo anterior los diagramas de clases y de interacción correspondientes a los artefactos del flujo de trabajo de la fase de análisis y diseño; obteniéndose de esta forma la construcción de la estructura de la solución propuesta. También, se presenta el modelo de datos como base para la implementación del Módulo administración, configuración y seguridad del sistema de Gestión para la Ingeniería Clínica y Electromedicina v2.0.

Capítulo 4. Implementación. A partir del modelado de la propuesta en el capítulo 3 se representa el diagrama de despliegue y de componentes en correspondencia con la implementación del sistema. Además, se exponen algunos de los criterios seguidos para el tratamiento de errores, la gestión de la seguridad y las estrategias de codificación y se explican los estándares y estilos utilizados en esta fase de desarrollo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el capítulo se efectúa un análisis de los principales referentes teóricos existentes a nivel internacional y nacional relacionados con la investigación en curso. Se muestra una breve descripción de los conceptos fundamentales asociados al tema de investigación. Asimismo, se describen las herramientas, tecnologías y metodologías empleadas en la solución propuesta.

1.1. Conceptos fundamentales asociados a la investigación

Administración:

La administración es el proceso de planificar, organizar, dirigir y controlar el uso de los recursos y las actividades de trabajo con el propósito de lograr los objetivos o metas de la organización de manera eficiente y eficaz. (3)

La administración procura siempre el máximo aprovechamiento de los recursos, mediante su utilización eficiente y sus funciones incluyen la planeación, organización, dirección y el control.

Administración de la Informática:

Mecanismo particular con que se debe dotar a la empresa para lograr establecer con toda claridad y en todo momento el alcance y acercamiento de la misión de la función informática, diseñada para esa empresa en particular, a través de su ciclo de crecimiento competitivo, en función de los objetivos de negocio de ella. (4)

Por otra parte, los autores del trabajo consideran que se entiende por administración de la informática: un mecanismo que permite la transformación de los recursos de una empresa de manera automatizada para facilitar de forma sencilla y ágil la ejecución de los procesos desempeñados por las personas que integran la organización. Posibilitando el control y la seguridad del o los sistemas, de manera tal que todo funcione armónicamente.

Módulo de Administración en un sistema informático:

El módulo de administración debe ser desarrollado pensando en facilitarle el trabajo al usuario, a los restantes módulos, debe ser a su vez común, que todos puedan utilizarlo y lo más importante debe ser totalmente dinámico. (5)

Es de gran importancia para todo sistema informático, contar con módulo de administración que sea el encargado de realizar funcionalidades para el mejoramiento, el control, la rapidez, la flexibilidad, la configuración, y otras que hacen que un sistema informático actual sea amigable y seguro, de modo que el usuario final se sienta satisfecho con la labor desempeñada por el equipo que desarrollo el software.

Configuración:

En informática la **configuración** es un conjunto de datos que se cargan, típico de las aplicaciones empresariales, como son programas de aplicación que provee recursos de hardware/software. La configuración caracteriza el valor de diferentes elementos de un sistema cómputo y puede modificarse en cualquier momento. Es decir, se constituye por el conjunto de programas que conforman a un sistema informático.

Enfocado el concepto en un marco del desarrollo de un software, los autores del presente trabajo consideran que la **configuración informática** es la manera de adaptar una aplicación de software a los demás componentes del entorno de desarrollo y a su vez a las necesidades del usuario. Una aplicación configurable es aquella desarrollada de forma que el usuario pueda configurar muchas de sus opciones.

Seguridad informática:

Es la disciplina que se ocupa de diseñar las normas, procedimientos, métodos y técnicas destinados a conseguir un sistema de información seguro y confiable.

El término seguridad comprende distintos aspectos esenciales para la seguridad de la información:

- Confidencialidad: la información sólo puede ser accedida por quién esté autorizado.
- Integridad: los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.
- Disponibilidad: la información tiene que estar disponible siempre que sea necesario, evitando por tanto, ataques externos que puedan reducir esta disponibilidad o incluso una caída del servicio. (6)

La seguridad informática comprende el conjunto de medidas administrativas, organizativas, físicas, técnicas, legales y educativas dirigidas a prevenir, detectar y responder a las acciones que ponen en riesgo la confidencialidad, integridad y disponibilidad de la información que se procesa, intercambia, reproduce y conserva por medio de las tecnologías de información (7).

1.2. Análisis de sistemas informáticos web para la administración, configuración y seguridad a nivel nacional e internacional

Algunas alternativas para resolver la problemática planteada fue el análisis de diversos sistemas web para la administración, configuración y seguridad existentes a nivel internacional y nacional, que se ajustarán a las necesidades que se tienen y tengan similitud con la arquitectura que conforma el SIGICEM. Dentro de los sistemas analizados se tiene:

SureGX: Software de seguridad y auditoría creado en Perú que permite gestionar eficientemente los accesos, controles y permisos de todos los usuarios para aplicaciones Web desarrolladas con GeneXus. Es la aplicación integradora de las soluciones instaladas en las empresas. Brinda beneficios como el buen control de auditoría y parametrización para un óptimo seguimiento de la administración de las aplicaciones. (8)

Contiene módulos como: Grupo de usuarios, el cual asigna automatizaciones y los permisos que tendrá sobre los sistemas de la empresa a los usuarios y grupos, Usuarios del sistema, que realiza una completa administración del usuario, Programas del sistema, que realiza el mantenimiento de los programas que maneja el sistema de forma integrada, Auditoría, que hace una completa revisión de las actividades y tareas realizadas en el sistema, Parámetros, que controla los niveles de acceso y/o permisos, así como el dinamismo de los sistemas desarrollados. Las tecnologías soportadas son: interfaz 100% Web, desarrollado en GeneXus, arquitectura tres capas, generación en lenguaje Java, .Net, sistemas gestores de bases de datos soportados: SQL server, Oracle, DB2, Informix, MySQL, PostgreSQL, servidor de aplicaciones Internet Information Server 5.0 (IIS) o superior, Tomcat Apache. (8)

Componente de Seguridad para SAS: Creado en La Habana en junio 2008, para aplicaciones del Área Temática Sistema de Apoyo a la Salud (SAS), de la UCI. Permite una gestión eficiente de usuarios, asignación de roles y privilegios de acceso a todos los sistemas externos que consuman sus servicios. Brinda un eficiente proceso de trazabilidad y auditoría. Permite además mediante los procesos de administración proporcionados, una gestión eficiente de todos los requerimientos de seguridad lográndose de esta forma la reutilización del código y evitando que se realicen acciones innecesarias fuera de cada negocio.

Su desarrollo está basado en tecnologías libres, multiplataformas, sobre una arquitectura en capas, PHP 5 como lenguaje de programación, patrón de arquitectura Modelo Vista Controlador (MVC), Sistema Gestor

de Base de Datos PostgreSQL 8.2, el conjunto de tecnologías AJAX (*Asynchronous JavaScript And XML*), librería YUI (*Yahoo User Interface*), estándares XML, Protocolo de transporte HTTPS. Su creación fue con la intención de sustituir los mecanismos de seguridad actuales de cada proyecto de esta área entre sus funcionalidades previstas se encuentran la administración eficiente de los usuarios y la asignación de sus privilegios en dependencia de su nivel, así como la optimización de la auditoría. (9)

SIGICEM: Módulo administración v1.1: El Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM), que se encarga de la informatización de todos los procesos de gestión tecnológica llevados a cabo en la red nacional de electromedicina de Cuba. Dentro de sus funcionalidades SIGICEM cuenta con un módulo que facilita la administración del sistema, dicho módulo permite la gestión eficiente de usuario, roles y permisos, además de la seguridad con un nivel de detalles alto.

Para su desarrollo se utilizó Rational Unified Process (RUP), Herramienta CASE (*Computer Aided Software Engineering*) Visual Paradigm, DB Designer 4 Fork, Zend Studio, frameworks: Symfony 1.4.8 y Extjs 2.2, soportado sobre la plataforma LAMP: Linux (L): Ubuntu, servidor web: Apache (A), Gestor de Base de Datos: MySQL (M) y lenguaje de programación: PHP (P). Garantiza la fiabilidad y actualización de la información referente al equipamiento médico instalado en el país, mediante la organización, planificación control y análisis de las acciones que se ejecuten.

Módulo de Seguridad del EBMS DSerp: Creado en La Habana, 2013. Sistema estratégico para la gestión avanzada de procesos y recursos que se desarrolla con el objetivo de dar soporte a la gestión de información empresarial. Permite administrar varias empresas que comparten dependencias funcionales sustanciales.

Su arquitectura de EBMS (Sistema Electrónico de Gestión de Empresas o *Electronic Business Management System* por sus siglas en inglés) le ofrece adaptabilidad para desplegarse en cualquier ambiente corporativo, con un modelo centralizado, flexible e interactivo, que le permite integrar y administrar procesos claves de una organización, garantizando la automatización de las actividades generadas por un conjunto de empresas, creando un dominio de negocio para la gestión de información en un entorno corporativo.

Se desarrolló incidiendo directamente en diez valores: accesibilidad, integridad, confidencialidad, información, sencillez, organización, fiabilidad, rapidez, eficiencia y eficacia. Para el desarrollo del módulo se utilizó la metodología XP (Extreme Programming), lenguaje de modelado UML (Unified Modeling

Language), entorno de desarrollo NetBeans, Herramienta CASE Visual Paradigm, Frameworks ExtJS y Symfony, Servidor web Apache bajo Linux, PostgreSQL y EMS SQL Management como una herramienta de alto rendimiento de administración de Bases de Datos. (10)

1.3. Análisis comparativo entre sistemas informáticos web analizados para la administración, configuración y seguridad a nivel nacional e internacional

Una vez descritos los sistemas que los autores consideraron referentes importantes como guía para dar solución a la problemática planteada, se realizó un análisis comparativo para una posible integración de los mismos con SIGICEM. Análisis mostrado en la siguiente tabla:

Tabla1. Comparación de los sistemas informáticos web para la administración, configuración y seguridad.

	SureGX	Componente de Seguridad para SAS	EBMS DSerp: Módulo de Seguridad	SIGICEM: Módulo administración v1.1	Módulo administración , configuración y seguridad
Perteneciente	GeneXus. Perú	Universidad de las Ciencias Informáticas (UCI). Cuba	Universidad de las Ciencias Informáticas (UCI). Cuba	Universidad de las Ciencias Informáticas (UCI). Cuba	Universidad de las Ciencias Informáticas (UCI). Cuba
Software y Licencia	Propietario y Privada	Libre y GPL	Libre y GPL	Libre y GPL	Libre y GPL
Lenguaje	Java, .Net	PHP 5	PHP 5	PHP 5	PHP 5.3.10
Frameworks	No	No	Extjs y Symfony	Extjs 2.2 y Symfony 1.4.8	Extjs 4 y Symfony 2
Correcta Administración	Sí	Sí	Sí	Sí	Sí
Correcta Configuración	Sí	Sí	No	No	Sí
Correcta Seguridad	Sí	No	Sí	Sí	Sí

1.3.1. Argumentación de la comparativa efectuada en la tabla 1

Después del análisis de los sistemas existentes relacionados con la administración, configuración y seguridad de aplicaciones web, para ver si pudieran ser integrados con el SIGICEM v2.0, se evidencia la

necesidad de desarrollar la solución propuesta, debido a que los sistemas analizados presentan inconsistencias que no le permite integrarse con la aplicación.

El sistema internacional SureGX brinda una completa administración, configuración, seguridad y permite la conexión con diferentes sistemas gestores de bases de datos, pero tiene como desventaja que es un software propietario, de elevado costo para la adquisición de su licencia, y solamente es integrable con sistemas desarrollados sobre las tecnologías Java y .Net.

El Componente de seguridad SAS a pesar que fue diseñado a la medida para aplicaciones en el sector de la salud y consume servicios web para todo lo referente a la seguridad, auditoría, autenticación y autorización; se ha demostrado que estos servicios no son factibles porque su funcionamiento no es estable y no se ajusta a los requerimientos del SIGICEM, ya que no fue desarrollado con el framework Symfony a pesar de las ventajas y funcionalidades que el mismo proporciona, donde se utilizarán en el módulo propuesto mecanismos de seguridad definidos por el propio framework, los cuales están probados y respaldados por su reconocimiento y por la comunidad de expertos por el que fue desarrollado.

Los sistemas web que utilizan servicios generalmente tienen como desventajas que al apoyarse en HTTP (*Hyper Text Transfer Protocol*), pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear:

- La comunicación entre programas.
- Existe poca información de servicios web para algunos lenguajes de programación.
- No es la solución a todos los problemas.
- Relativamente nuevo, algunos estándares no definidos.
- Dependencia de la disponibilidad de servidores y comunicaciones.

Sin embargo, los sistemas restantes son de la misma procedencia y se asemejan bastante, pero presentan problemas con la configuración y no es lo que se requiere en el desarrollo de la propuesta. El módulo de seguridad del EBMS DSerp tiene una deficiente configuración de la seguridad y no permite definir roles y políticas de acceso personalizadas. Igualmente, la primera versión del SIGICEM no es lo suficientemente configurable para el trabajo del usuario con el sistema y no existe migración de los frameworks con el que fue desarrollado a los actuales, lo cual atenta contra la escalabilidad del sistema.

Todo el análisis efectuado permitió a los autores ratificar la necesidad de desarrollar un nuevo módulo para el SIGICEM v2.0 que permita la administración, configuración y seguridad dentro del mismo.

1.4. Metodologías, tecnologías y herramientas

Para la construcción de un software uno de los pasos más importantes es la correcta selección de las herramientas y tecnologías a utilizar. En el epígrafe se describen las características y funcionalidades de las herramientas, tecnologías y metodología establecidas por el proyecto Sistema de Gestión para Ingeniería Clínica y Electromedicina y el Centro de Informática Médica que además, serán las utilizadas en el desarrollo del Módulo administración, configuración y seguridad del SIGICEM v2.0.

1.4.1. Metodología de Desarrollo: Rational Unified Process (RUP)

RUP, proceso de desarrollo de software, que junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización (11). Es dirigido por casos de usos, centrado en la arquitectura y su desarrollo es iterativo e incremental. La arquitectura provee la estructura sobre la cual guiar el trabajo en iteraciones, mientras que los casos de uso definen las metas y dirigen el trabajo en cada iteración (12). De este modo los desarrolladores obtendrán una buena organización del trabajo y una visión precisada.

1.4.2. Lenguaje Unificado de Modelado: Unified Modeling Language 2.1

Lenguaje de modelado de sistemas de software utilizado por la metodología RUP, del inglés *Rational Unified Process*, en la creación de diseños que capturen las ideas de los desarrolladores de forma convencional y de fácil comprensión, para un mejor entendimiento y comunicación con otras personas. A través de este lenguaje gráfico se podrá visualizar, especificar, construir y documentar el módulo propuesto en esta investigación, permitiendo además la descripción de aspectos concretos relacionados con el lenguaje de programación utilizado, los esquemas de la base de datos, así como los componentes reutilizables. (13)

Este lenguaje de modelado permite crear un nivel de comprensión y entendimiento entre los analistas, desarrolladores o cualquier personal involucrado y hace más fácil la comunicación existente entre ellos. (14)

1.4.3. Herramienta CASE

Las herramientas de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés) son software destinados a incrementar la productividad en el proceso de desarrollo de software, reduciendo los costos de tiempo y recursos. Entre sus principales características se encuentran las facilidades que brindan para la realización de prototipos y el desarrollo conjunto de aplicaciones, facilita la reutilización de componentes de software y mejora y estandariza la documentación. (15)

Visual Paradigm for UML 6.4

Herramienta de modelado visual para todo tipo de diagramas UML (Lenguaje Unificado de Modelado), desarrollada por la empresa Visual Paradigm Internacional. Esta herramienta está diseñada para el desarrollo de software orientado a objetos brindando soporte para todo su ciclo de vida, así como muchas facilidades para el trabajo colaborativo, además de la integración para el trabajo con modelos relacionales de Bases de Datos. (16)

1.4.4. Symfony 2

Symfony, Framework para desarrollar aplicaciones web en PHP con buena reputación, fue adoptado rápidamente por los profesionales que trabajan en este campo después de su lanzamiento en 2005. Entorno estable bien conocido y reconocido internacionalmente. Hoy en día sigue siendo la herramienta diaria utilizada por sus propios equipos para desarrollar proyectos de clientes. Diseñado por profesionales para profesionales. Se distribuye bajo la licencia Open Source MIT (Instituto Tecnológico de Massachusetts o *Massachusetts Institute of Technology*, por sus siglas), que no impone restricciones y permite el desarrollo de código abierto, así como las aplicaciones propietarias. Es todo lo que se espera de un marco: la velocidad, la flexibilidad, los componentes reutilizables. Ha desarrollado un hábito de sacudir el orden establecido y siempre se esfuerza por la excelencia, la búsqueda de la mejora continua en la productividad de los desarrolladores. (17)

➤ FOSUserBundle 1.1.3

Componente de seguridad de Symfony 2 que proporciona una plataforma flexible y permite cargar usuarios desde la configuración. Por tanto, si un sistema necesita persistir y recuperar usuarios hacia y desde una base de datos, entonces es mejor usar FOSUserBundle. Entre sus características permite

registrar nuevos usuarios, enviar un mensaje de confirmación a los usuarios recién creados, editar el perfil de usuario, encriptación de contraseñas y agrega un campo en la base de datos para poder guardar roles.

1.4.5. ExtJS 4

Gran paso adelante para los frameworks web basándose en Ext JS 3.3, la última versión añade más de 350 nuevas aplicaciones, 50 nuevas clases, y el 65% más documentación. Un nuevo paquete entero de datos equipa a los desarrolladores aprovechar las características como Red infinita *Scrolling* para construir un nuevo nivel de interactividad a las aplicaciones web. No hay plugins, no hay problema. Ofrece las capacidades de gráficos más avanzados de cualquier marco de JavaScript, sin depender de plugins, ofreciendo una visualización perfecta de píxeles en cualquier navegador en cualquier sistema operativo. Aprovechando SVG (Gráficos Vectoriales Redimensionables o *Scalable Vector Graphics*, por sus siglas en inglés) y VML (*Vector Markup Language*), su paquete de gráficos permite a los desarrolladores diseñar y programar sus gráficas una vez para todos los navegadores. Trae una nueva forma de construir aplicaciones cliente, mediante la introducción del patrón modelo-vista-controlador (MVC), por sus siglas, popular para Ext JS. (18)

Al permitir la separación de la gestión de datos, la lógica y elementos de la interfaz, Ext JS 4 hace que sea más fácil para los equipos de desarrollo, incluso de gran tamaño para trabajar de forma independiente. Se entrega con una guía útil MVC para empezar. Permite a los desarrolladores entregar una increíble variedad de navegadores. Ya sea que esté utilizando Ext JS incorporado en los componentes de interfaz de usuario, utilizando el paquete de gráficos, o la terminalización de la aplicación, hace que sea fácil de construir una aplicación que brinda el poder de la web, independientemente de qué navegador utiliza su cliente. (18)

1.4.6. Entorno Integrado de Desarrollo

Un Entorno Integrado de Desarrollo (*IDE*, por sus siglas en inglés) representa una herramienta utilizada por los programadores para la implementación del producto deseado. Con su utilización es posible crear la interfaz gráfica de usuario y escribir el código de lo que hará la aplicación de una manera más amena y en una interfaz amigable. (19)

Consiste en un editor de código, un compilador, un depurador y constructor de interfaz gráfica.

NetBeans 7.3

NetBeans, proyecto de código abierto dedicado a proveer productos de desarrollo de software de roca sólida (NetBeans *IDE* y la Plataforma NetBeans) que se ocupan de las necesidades de los desarrolladores, los usuarios y las empresas que dependen de NetBeans como base para sus productos; sobre todo, para permitir que se desarrollen estos productos de forma rápida, eficaz y sencilla mediante el aprovechamiento de los puntos fuertes de la plataforma Java y otras normas relevantes de la industria. El IDE NetBeans es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++. Este proyecto está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos, así como una amplia selección de complementos de terceros. (20)

NetBeans, además de ser de código abierto y totalmente gratuito, su última versión ofrece un asistente para crear un proyecto PHP utilizando el framework Symfony y la integración con este permite desarrollar aplicaciones de forma más sencilla y productiva.

1.4.7. Plataforma de Desarrollo: LAMP

El Módulo de administración, configuración y seguridad constituye el primer módulo del Sistema de Gestión para Ingeniería Clínica y Electromedicina v2.0, por lo que se utiliza su misma plataforma de desarrollo: LAMP.

La misma está compuesta por un conjunto de aplicaciones de código abierto que trabajando entre sí, permiten disponer de un servidor web, ya sea para desarrollo o para producción. La plataforma, está integrada por:

- Linux (L) con su distribución Ubuntu 12.04 como sistema operativo.
- Apache (A) como servidor web en su versión 2.2.
- MySQL (M) en su versión 5.5, como sistema gestor de base datos.
- PHP (P) en su versión 5.3.10 como lenguaje de programación. (21)

En el presente capítulo, el estudio y descripción de los principales sistemas a nivel nacional e internacional, aportaron nuevas ideas para definir las necesidades funcionales en el desarrollo del Módulo administración, configuración y seguridad, así como en los diferentes conceptos relacionados con la investigación que permitieron un mejor entendimiento del entorno de desarrollo. Además, se describieron las herramientas, tecnologías y metodología definidas para la nueva versión de SIGICEM v2.0, las cuales serán las aplicadas en el desarrollo de la solución propuesta, ya que ofrecen ventajas para su implementación porque se integran fácilmente. En el caso de la metodología seleccionada representa una garantía para la continuidad del trabajo del proyecto, permitiendo así, guiar el proceso de desarrollo del módulo a partir del uso de las herramientas y tecnologías definidas, las cuales a su vez permiten desarrollar un producto de calidad, que facilite la comunicación entre los desarrolladores y ayuda a soportar la fase de implementación del desarrollo del software.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el capítulo se especifican los requisitos funcionales y no funcionales asociados al Módulo de administración, configuración y seguridad del Sistema de Gestión para la Ingeniería Clínica y Electromedicina v2.0, quedando de forma explícita y detallada una representación y descripción de los casos de usos que tendrá el sistema. Además, se hace una descripción del modelo de dominio definido, así como se muestra una descripción detallada de la solución propuesta.

2.1. Modelo de Dominio

En aras de ayudar a comprender los conceptos claves del problema planteado en la investigación se decide representar un modelo de dominio; a partir que este artefacto permite disminuir la brecha de representación entre cómo ven los clientes el problema y la representación utilizando un modelado orientado a objetos en la solución, en este caso en el Módulo de administración, configuración y seguridad del SIGICEM v2.0.

Un modelo de dominio se considera un artefacto de la disciplina de análisis, construido con las reglas de UML, presentado como uno o más diagramas de clases que contiene, la realidad física del sistema. Pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. El modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.

(22)

2.1.1. Descripción del Modelo de Dominio

El Modelo de dominio se describe mediante diagramas de clases del dominio, que muestran las clases del dominio y sus respectivas asociaciones. En la figura 1 se representa el diagrama de clase del modelo de dominio realizado.

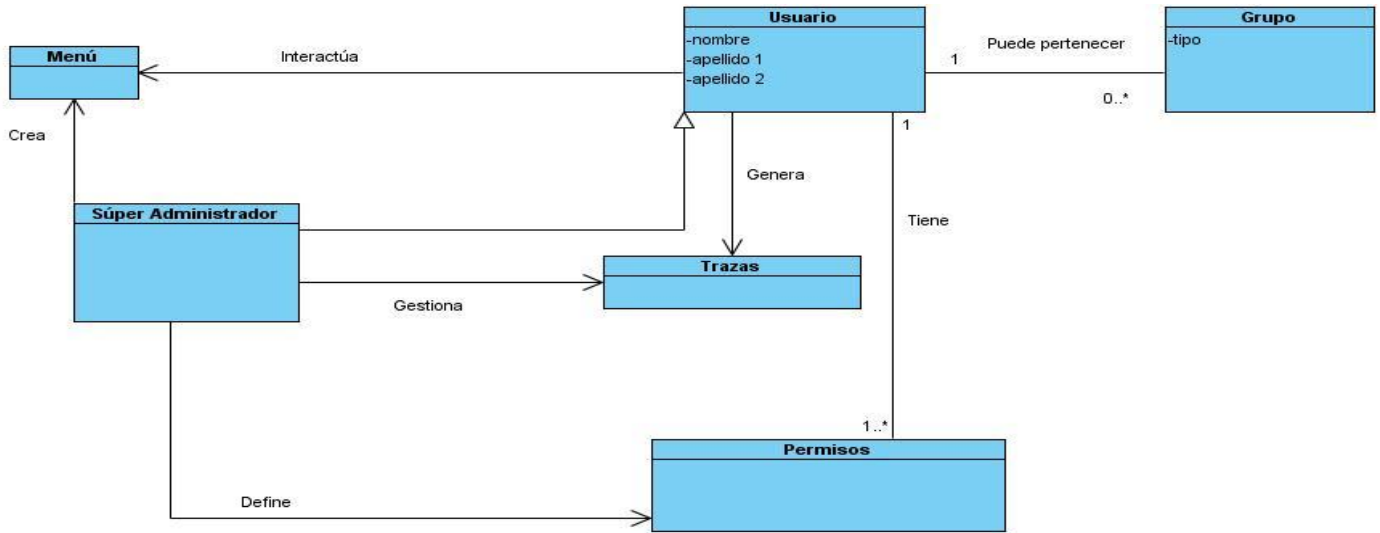


Figura 1. Diagrama de clases del dominio.

2.1.2. Definición de clases del Modelo de Dominio

Súper Administrador: responsable de gestionar los usuarios y grupos de la aplicación, además de definir los permisos que pueden tener los mismos y crear los menús correspondientes.

Usuario: encargados de interactuar con la aplicación de acuerdo a los permisos asignados.

Grupo: conjuntos de usuarios con permisos específicos.

Menú: estructura de acceso para la navegación a las funcionalidades que tiene la aplicación.

Permisos: contiene los tipos de acceso que se pueden tener sobre los diferentes módulos.

Trazas: representa todo el registro de trazas generadas por los usuarios del sistema.

2.1.3. Relación entre clases

El Súper Administrador gestiona los usuarios que pueden o no pertenecer a uno o varios grupos, teniendo ambos permisos definidos por este. Por otra parte el Súper Administrador puede crear el menú para la navegación en el sistema y el usuario interactúa con este.

Los usuarios interactúan con los módulos de la aplicación ejecutando acciones de inserción, modificación o eliminación de la información registrada en la base de datos. Al mismo tiempo las acciones del usuario serán guardadas en un historial de trazas para la posible visualización de las mismas.

2.2. Propuesta de solución

Para darle solución al problema planteado se propone desarrollar un módulo en el SIGICEM v2.0 que permita la correcta administración, configuración y seguridad, de modo que garantice el control total de los usuarios, roles y permiso, además que propicie la correcta navegación del usuario, que impida las brechas de seguridad y la inconsistencia de la información. El módulo contará con restricciones de permisos a los usuarios o grupos en dependencia del rol y los privilegios asignados. Existirán dos actores del sistema, súper administrador y los usuarios. El sistema debe permitir además el control de trazas para poder gestionar todas las acciones que se realizan sobre la aplicación una vez que ocurra algún problema ocasionado por un error del usuario que provoque cualquier alteración de la información.

2.3. Análisis de requisitos de software

El análisis de requisitos es una de las tareas más importantes en el ciclo de vida del desarrollo de software. En cualquier proyecto software los requisitos son las necesidades del producto que se debe desarrollar. Por ello, en la fase de análisis de requisitos se deben identificar claramente estas necesidades y documentarlas. Como resultado de esta fase se debe producir un documento de especificación de requisitos en el que se describa lo que el futuro sistema debe hacer. (23)

Los requisitos se pueden clasificar en funcionales, considerando las funciones que el sistema debe cumplir, y no funcionales, teniendo en cuenta las cualidades o propiedades que debe poseer el producto.

2.3.1. Requisitos Funcionales

Tabla 2. Requisitos Funcionales

Requisitos Funcionales	
RF1 Visualizar datos del usuario	RF2 Registrar usuario
RF3 Modificar datos del usuario	RF4 Eliminar usuario
RF5 Visualizar grupos	RF6 Registrar grupo
RF7 Modificar datos del grupo	RF8 Eliminar grupo
RF9 Agregar usuario a grupo	RF10 Cambiar contraseña
RF11 Agregar grupo a usuario	RF12 Autenticar usuario
RF13 Crear traza de modificación	RF14 Crear traza de eliminación

RF15 Crear traza de Inserción	RF 16 Gestionar traza
RF16 Asignar permisos a grupo	RF17 Asignar permisos a usuario
RF18 Personalizar accesos directos	RF19 Personalizar entorno escritorio

2.3.2. Requisitos no Funcionales

RNF Usabilidad

- Los usuarios que interactúan con el sistema no requieren de grandes conocimientos de informática.
- La aplicación web deberá facilitar la interacción usuario – sistema con el objetivo de evitar rechazo en el uso de la misma, puesto que se realiza una aplicación sencilla con un ambiente similar al de Windows que le permitirá al usuario final una mejor familiarización con la misma.
- El sistema brindará al usuario mensajes descriptivos para orientarlo en cada acción realizada.

RNF Confiabilidad

- El sistema cerrará la sesión del usuario autenticado después de quince minutos de inactividad.
- Al ocurrir un error el sistema muestra un mensaje personalizado indicando una mala operación o funcionamiento del sistema.

RNF Soporte

- Una vez puesta en marcha la aplicación y siendo usada por los usuarios finales, se recogerá toda la información referente a los defectos y/o no conformidades existentes, para incorporar las mejoras sugeridas al sistema.
- La implementación del módulo 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.

RNF Restricciones de Diseño

- El sistema estará desarrollado en base a las políticas del software libre que fueron ajustadas al Sistema Nacional de Salud como: uso de servidores GNU/Linux.

- Se utilizará como gestor de base de datos MySQL, Symfony 2 como framework de desarrollo de aplicaciones web, el cual utiliza PHP como lenguaje de programación del lado del servidor, y NetBeans como IDE de desarrollo.
- Para el diseño de interfaces se utilizará EXT JS 4, librería javascript para el desarrollo de aplicaciones RIA (Rich Internet Application).
- Se utilizará Visual Paradigm como herramienta para el modelado de los artefactos generados por la metodología RUP.

RNF Requisitos para la documentación de usuarios en línea y ayuda del sistema

- Para informar a los usuarios todo lo referente a cómo trabaja el sistema se incorporará una ayuda que le permitirá orientarse en cada acción a realizar.

RNF Interfaz

- 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 del sistema operativo Windows XP 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 e informada claramente 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 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.

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. (24)

Tabla 3. Actores del Sistema

Actor	Descripción
Usuario	Usuario común que interactúa con el sistema de acuerdo a los privilegios de que le han sido asignados.
Súper Administrador	Es el encargado de asignar permisos a usuarios y grupos del sistema. Además, crea el Menú correspondiente a cada usuario.

2.5. Diagrama de caso de uso del Sistema

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar (25).

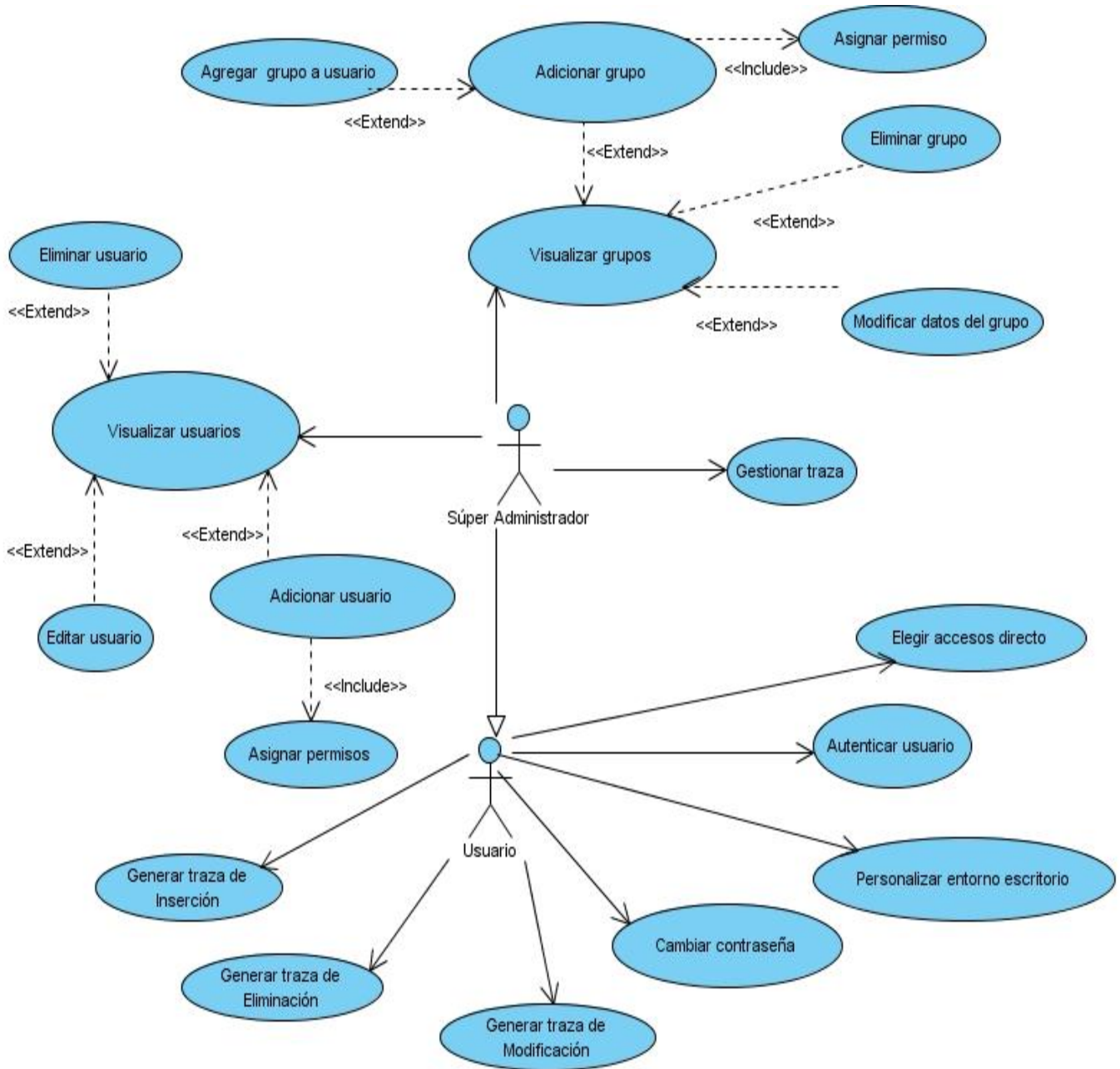


Figura 2. Diagrama de Casos de Uso del Sistema (CUS)

2.6. Descripción de casos de uso del Sistema (CUS)

2.6.1. Registrar usuario

Tabla 4.CU Registrar Usuario

Objetivo	Adicionar un usuario al sistema.	
Actores	Súper Administrador	
Resumen	El caso de uso inicia cuando el actor accede a la opción adicionar usuario en la interfaz de visualizar usuarios, donde el sistema da la posibilidad de seleccionar los datos del usuario a insertar, el actor introduce los datos del usuario, el sistema adiciona el usuario, el caso de uso termina.	
Complejidad	Alta	
Prioridad	Crítica	
Precondiciones	El usuario que se desea adicionar no puede estar registrado en el sistema, y los datos insertados deben estar correctos.	
Postcondiciones	Se confirmó la correcta inserción del usuario.	
Flujo de eventos		
Flujo básico Adicionar Usuario		
	Actor	Sistema
1.	Accede a la opción "Adicionar" de la interfaz Usuarios.	
2.		Brinda la posibilidad de seleccionar los datos del usuario: <ul style="list-style-type: none"> • Activo • Súper Administrador • Permisos • Grupo Permite introducir: <ul style="list-style-type: none"> • Usuario

Capítulo 2: Características del sistema

		<ul style="list-style-type: none"> • Contraseña • Repita Contraseña Asignar un usuario al grupo <ul style="list-style-type: none"> • Disponibles
3.	Introduce los datos del usuario y escoge la opción Guardar.	Valida los datos. Si hay datos incompletos, ver Alternativa 1 : “Existen campos vacíos”. Si hay datos incorrectos, ver Alternativa 2 : “Existen campos incorrectos.” Si los datos son correctos se adiciona el usuario.
4.		Muestra el mensaje de información “Se adicionó el usuario correctamente.”
5.		Termina el caso de uso
Flujos alternos		
1. Existen campos vacíos		
	Actor	Sistema
1.		Muestra el mensaje de error “Existen campos vacíos”
2.		Muestra un indicador al lado de los campos vacíos.
Flujos alternos		
2. Existen campos incorrectos		
	Actor	Sistema
1.		Muestra el mensaje de error “Existen campos incorrectos”
2.		Muestra un indicador sobre los campos incorrectos.

Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

2.6.2. Eliminar Usuario

Tabla 5.CU Eliminar Usuario

Objetivo	Eliminar usuario.	
Actores	Súper Administrador	
Resumen	El caso de uso inicia cuando el actor selecciona un usuario de la lista de usuario y accede a la opción Eliminar usuario, el sistema elimina el usuario, el caso de uso termina.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Debe existir en el sistema el usuario que se desea eliminar.	
Postcondiciones	El usuario ha sido eliminado del sistema	
Flujo de eventos		
Flujo básico Listar Usuario		
	Actor	Sistema
1.	Accede a la opción "Eliminar", del usuario que desea eliminar.	Muestra el mensaje de confirmación "Esta seguro que desea eliminar el usuario seleccionado" y permite: <ul style="list-style-type: none"> • Aceptar • Cancelar Ver Alternativa 1: "Cancelar operación."

Capítulo 2: Características del sistema

2.	Selecciona la opción Aceptar.	Elimina el usuario
3.		Termina el caso de uso
Flujos alternos		
1. Cancelar operación.		
	Actor	Sistema
1.	Selecciona la opción "Cancelar".	Regresa a la vista anterior.
2.		El caso de uso termina.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

El presente capítulo permitió obtener una mejor comprensión del problema en cuestión, mediante la documentación obtenida a través de los flujos de trabajo: modelamiento del negocio y requerimientos. Además, el modelamiento de los artefactos del flujo de trabajo de análisis y diseño sirvió de guía para la posterior implementación del Módulo administración, configuración y seguridad. Finalmente, se obtuvo una visión detallada y entendible de la solución propuesta.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

El presente capítulo constituye la base para la implementación del Módulo administración, configuración y seguridad porque es donde se construye su estructura mediante la comprensión de los aspectos relacionados con los requisitos funcionales definidos en el capítulo 2. También, se presenta el modelo de datos diseñado a partir de la metodología definida.

3.1. Modelo de análisis

Descrito con el lenguaje de desarrollador, estructurado por clases y paquetes estereotipados. Utilizado fundamentalmente por los desarrolladores para comprender cómo debería ser diseñado e implementado el sistema. No debería contener redundancia e inconsistencia entre requisitos. Esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa para la arquitectura; sirve como una primera aproximación al diseño. (26)

Modelo conceptual de objetos que ayuda a refinar los requerimientos y permite a los desarrolladores describir la estructura interna del sistema; jerarquía de paquetes de análisis que agregan clases de análisis y realizaciones de casos de uso. En el mismo se describen las clases de análisis bajo sus tres estereotipos conceptuales, estos son: (27)



Interfaz: Toda la funcionalidad especificada en las descripciones de los casos de uso que depende directamente de los aspectos externos del sistema se ubica en los objetos de interfaces. Se utilizan para modelar la interacción entre el sistema y sus actores.



Entidad: Las Clases Entidad (Entity) son usadas para modelar la información que tiene permanencia en el tiempo y es persistente. Además modelan la información y el comportamiento asociado de algún concepto como una persona, evento u objeto del mundo real.



Control: Las clases “control” representan la coordinación, secuencia, gestión de transacciones y control de otros objetos. Usualmente se utilizan para encapsular el control relacionado con un caso de uso específico.

3.1.1. Diagramas de Clases del Análisis

El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. (28)

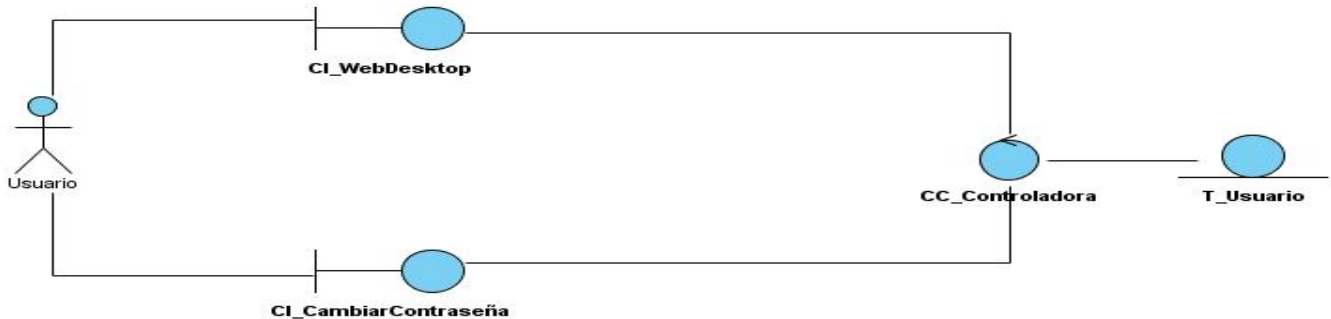


Figura 3. Diagrama de clase del análisis CU: Cambiar Contraseña.

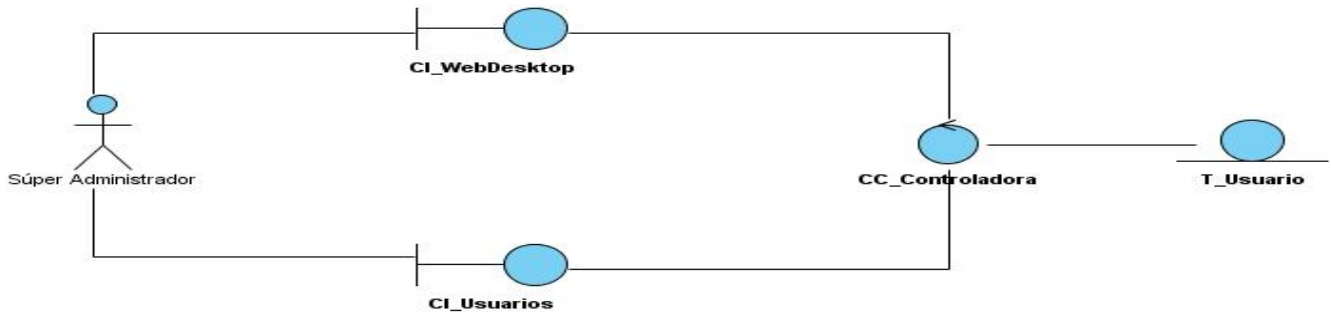


Figura 4. Diagrama de clase del análisis CU: Visualizar Usuario.

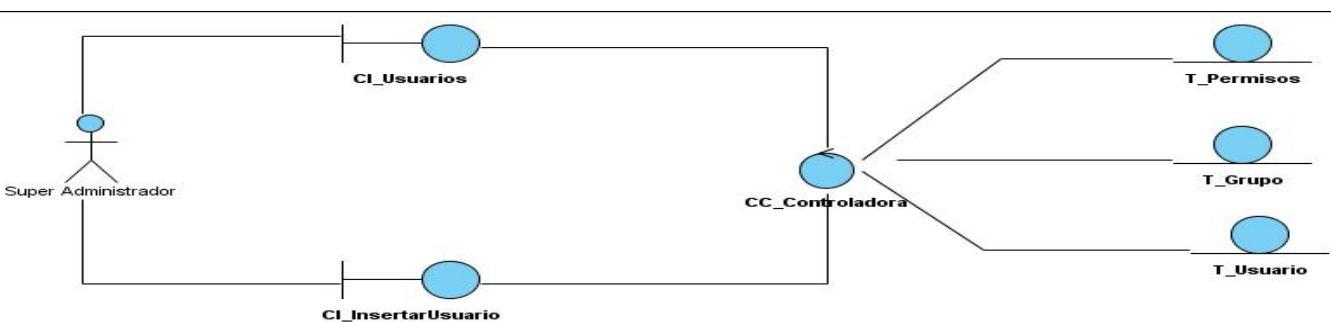


Figura 5. Diagrama de clase del análisis CU: Insertar Usuario.

3.1.2. Diagramas de Interacción. Colaboración

El Diagrama de Colaboración presenta una alternativa al diagrama de secuencia para modelar interacciones entre objetos en el sistema. Mientras que el diagrama de secuencia se centra en la secuencia cronológica del escenario que estamos modelando, el diagrama de colaboración se centra en estudiar todos los efectos de un objeto dado durante un escenario. Los objetos se conectan por medio de enlaces y cada uno de estos representa una instancia de una asociación entre las clases implicadas. El enlace muestra los mensajes enviados entre los objetos, el tipo de mensaje (sincrónico, asíncrono, simple, blanking, y 'time-out'), y la visibilidad de un objeto con respecto a los otros. (28)

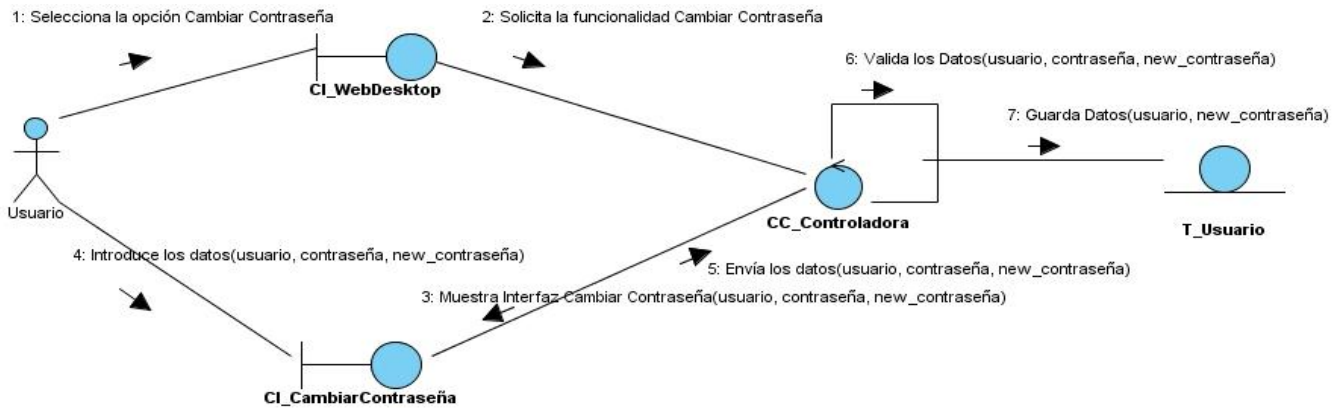


Figura 5. Diagramas de Colaboración del CU: Cambiar Contraseña.

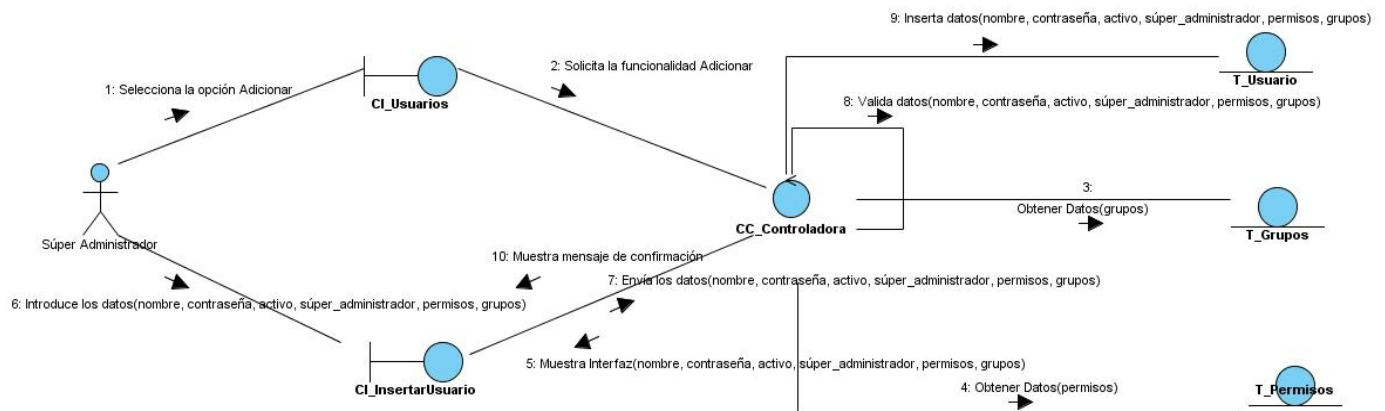


Figura 6. Diagramas de Colaboración del CU: Insertar Usuario.

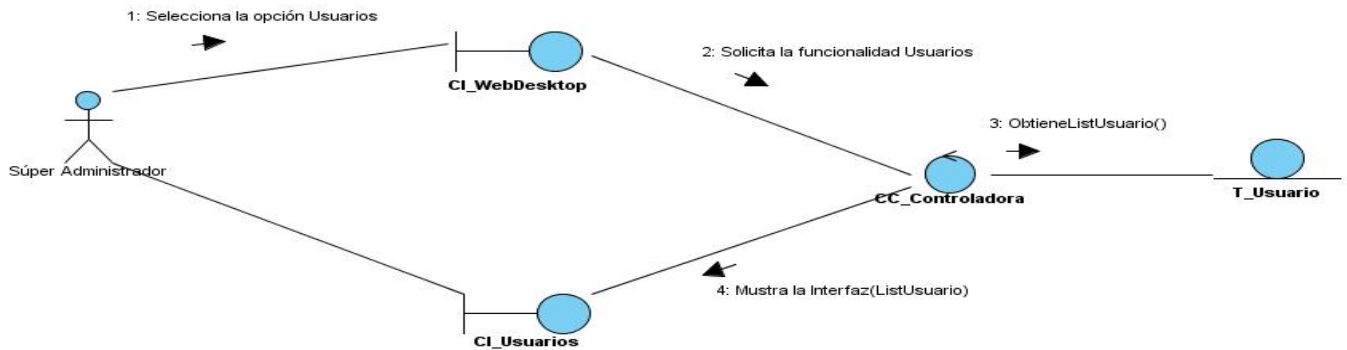


Figura 6. Diagramas de Colaboración del CU: Visualizar Usuario.

3.2. Modelo de Diseño

Es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a implementación. Representa a los casos de uso en el dominio de la solución. Puede contener: los diagramas, las clases, paquetes, subsistemas, capsulas, protocolos, interfaces, relaciones, colaboraciones, atributos, las realizaciones de los casos de uso, entre otros que se puedan considerar para el sistema en desarrollo. (29)

Para la representación de elementos específicos del diseño de aplicaciones web son utilizados los estereotipos web. La extensión UML para web, establece como elementos significativos, tres clases estereotipadas de la siguiente manera: (30)

- 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. Diagrama de Clase del Diseño

Durante el diseño, el diagrama de clases definido en el análisis se usa y se modifica para satisfacer los detalles de las implementaciones. (28)

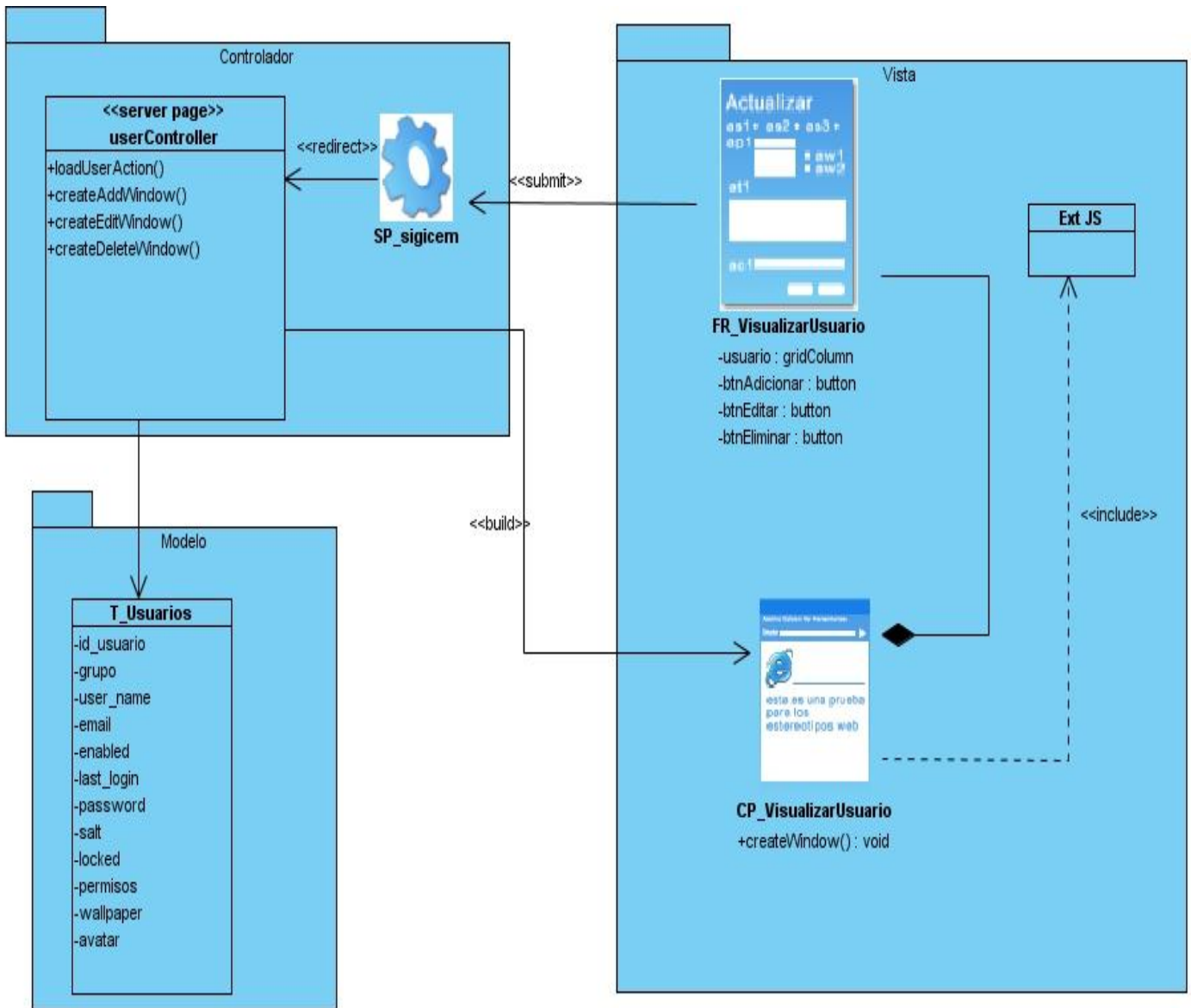


Figura 7. Diagramas de clase del diseño CU: Visualizar Usuario.

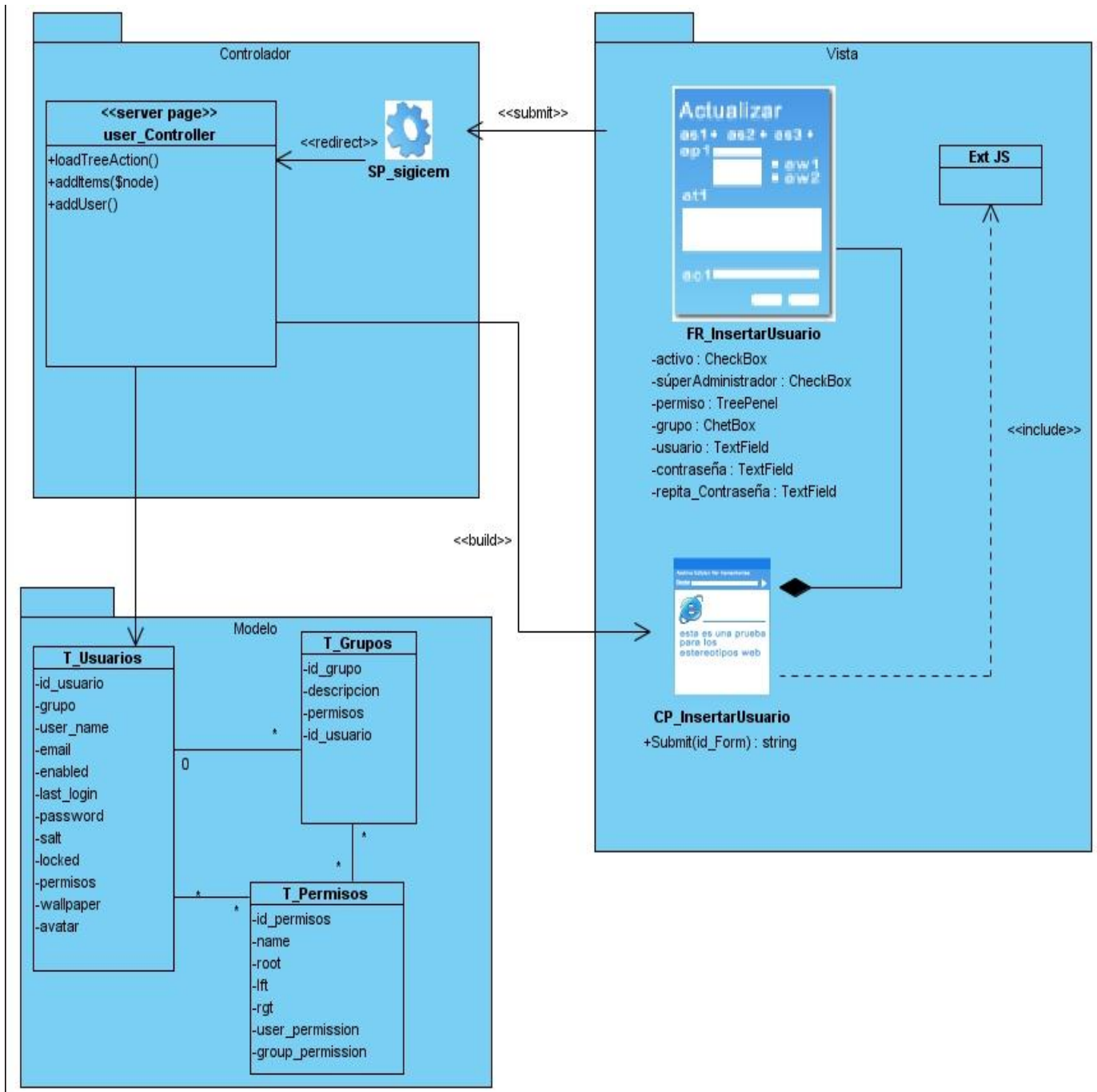


Figura 8. Diagramas de clase del diseño CU: Insertar Usuario.

3.2.2. Descripción de las Clases del Diseño

Tabla 6. Descripción de las clases del diseño.

Clases	Descripción
SP_sigicem	La página servidora sigicem es el punto de entrada único de toda la aplicación denominada comúnmente controlador frontal. Todas las peticiones realizadas son gestionadas por esta. 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.
actions_Usuario	Representa una clase controladora, en ella se encuentra la lógica de la aplicación del CU Usuario, está compuestos por métodos y acciones (métodos especiales) asociadas a esta funcionalidad, que a su vez utiliza el modelo y define las variables necesarias para la vista.
actions_Grupo	Representa una clase controladora, en ella se encuentra la lógica de la aplicación del CU Grupo, está compuestos por métodos y acciones (métodos especiales) asociadas a esta funcionalidad, que a su vez utiliza el modelo y define las variables necesarias para la vista.
CP_VisualizarUsuario	Representa el acceso y la vista del usuario en la Interfaz Usuario. Se encarga de atender todas las selecciones y peticiones que puedan realizar un usuario determinado. Además, contiene todo el código necesario para crear un formulario, mediante el cual el usuario interactúa.
CP_InsertarUsuario	Representa el acceso del usuario en la Interfaz Insertar Usuario. Se encarga de atender todas las selecciones y peticiones que puedan realizar un usuario determinado.
FR_VisualizarUsuario	Representa la entrada y salida de datos en el sistema. Tiene todos los elementos necesarios para recoger la información de un usuario, la misma es

	enviada a la clase controladora dónde se procesa esta información. A través de esta se podrá visualizar los datos de los usuarios del sistema.
FR_InsetarUsuario	Representa la entrada y salida de datos en el sistema, a través de esta, el súper Administrador podrá asignar usuarios al sistema. Tiene todos los elementos necesarios para recoger la información de un usuario, la misma es enviada a la clase controladora dónde se procesa esta información.
T_Usuarios	Es la 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 los usuarios. Es decir representa a la tabla de los Usuarios de la base de datos. Contiene todos los atributos necesarios de los usuarios del sistema.
T_Grupos	Es la 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 los grupos. Es decir representa a la tabla de los Grupos de la base de datos. Contiene todos los atributos necesarios de los grupos.

3.3. Patrón arquitectónico

El Patrón Arquitectónico que se utilizó en la realización de las clases del diseño para del sistema es el Modelo Vista Controlador (MVC), definido por los directivos del proyecto Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM) para el desarrollo de sus productos. El mismo incrementa la reutilización y la flexibilidad del sistema y separa la lógica de negocio de la Interfaz de usuario por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

Symfony está basado en este patrón, clásico del diseño web, formado por tres niveles claramente definidos: (31)

- El Modelo: representa la información con la que trabaja la aplicación, es decir, su lógica de negocio y se encarga de acceder a los datos.
- La Vista: transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador: se encarga de procesar las interacciones del usuario y transformar sus peticiones

en operaciones sobre el modelo y la vista.

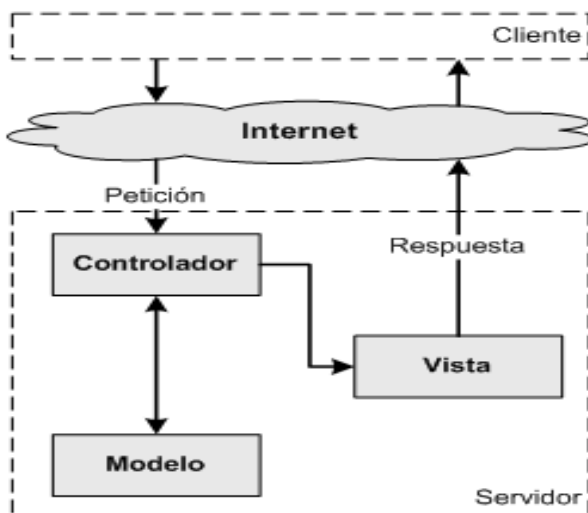


Figura 10. El patrón MVC

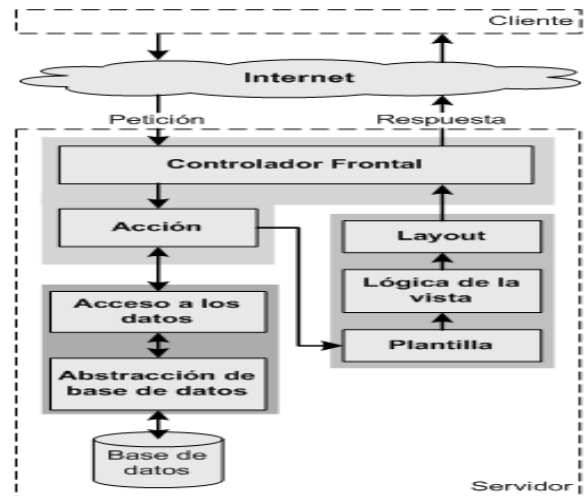


Figura 11. El flujo de trabajo de Symfony

3.4 Patrones de Diseño

Facilitan la reutilización de las clases y del propio diseño. Son descripciones de clases cuyas instancias colaboran entre sí, que deben ser adaptados para resolver problemas de diseño generales en un contexto particular. Un patrón de diseño identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades. (32)

Los patrones de diseño tienen una importante utilidad en el diseño realizado, se dividen en dos grandes grupos: Asignación General de Responsabilidades (GRASP, por sus siglas en inglés) y Banda de los cuatros (GOF, por sus siglas en inglés). En la realización del capítulo 1 del presente trabajo de investigación se decidió utilizar el framework de desarrollo Symfony, definido por los directivos del proyecto SIGICEM, el mismo se encuentra concebido de tal manera que puede afirmarse que obliga a la utilización de varios patrones. Dentro de estos los que han sido considerados a utilizar se presentan y describen a continuación:

GRASP: Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (33).

Experto: Al incluir Doctrine como Object-Relational Mapping (ORM, por sus siglas en inglés), que permite generar el modelo de clases, Symfony 2 crea entidades con responsabilidades debidamente asignadas para manipular la información de la base de datos. Estas entidades contienen la información precisa de la tabla que representan, al mismo tiempo que cuentan con las funcionalidades necesarias para el trabajo con los datos. A continuación se presenta un ejemplo de la clase T_Usuarios, que cuenta con la información necesaria de los usuarios del sistema, para cumplir cualquier responsabilidad que tenga que ver con estos, lo que evidencia la solución que propone el patrón Experto.

T_Usuarios
-id_usuario -grupo -user_name -email -password -avatar
+getIdUsuario() : int +getGrupo() : int +getUser_Name() : string +getEmail() : string +getPassword() : string +setUserName() : void +setEmail() : void +setPassword() : void +setAvatar() : void

Figura12. Ejemplo del patrón Experto.

Creador: Responde al problema de quién debería ser el responsable de crear nuevas instancias de una clase. La creación de instancias es una actividad común en los sistemas de software, por lo que es ventajoso contar con un responsable para la asignación de las responsabilidades de creación, lo que puede proporcionar mayor claridad, encapsulación, reutilización y brinda soporte a un bajo acoplamiento.(34)

Su utilización se evidencia mediante el uso del contenedor de servicios de Symfony 2, el cual es un objeto PHP que gestiona la creación de instancias de objetos.

Bajo Acoplamiento: Se manifiesta mediante la inyección de dependencias que provee Symfony2. Esta consiste en pasar a las clases todos los objetos que necesitan ya creados y configurados, logrando de esta manera reducir las dependencias entre las clases de la aplicación. (35) Un ejemplo de su empleo se

manifiesta en la figura que aparece a continuación:

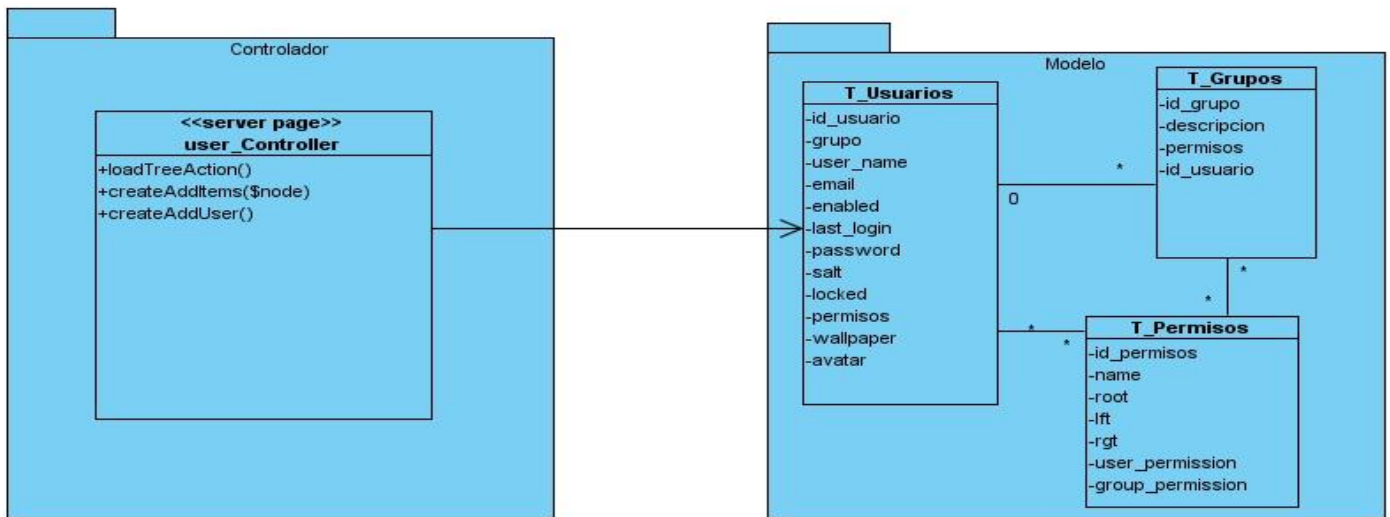


Figura13. Ejemplo del patrón Bajo Acoplamiento.

Alta Cohesión: El framework Symfony permite la organización del trabajo en cuanto a la estructura de un proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase `user_Controller`, la cual está formada por varias funcionalidades altamente relacionadas entre sí. Su utilización mejora la claridad, la facilidad para comprender el diseño y las mejoras de funcionalidad. Además, proporciona un fácil mantenimiento y soporta mayor capacidad de reutilización.

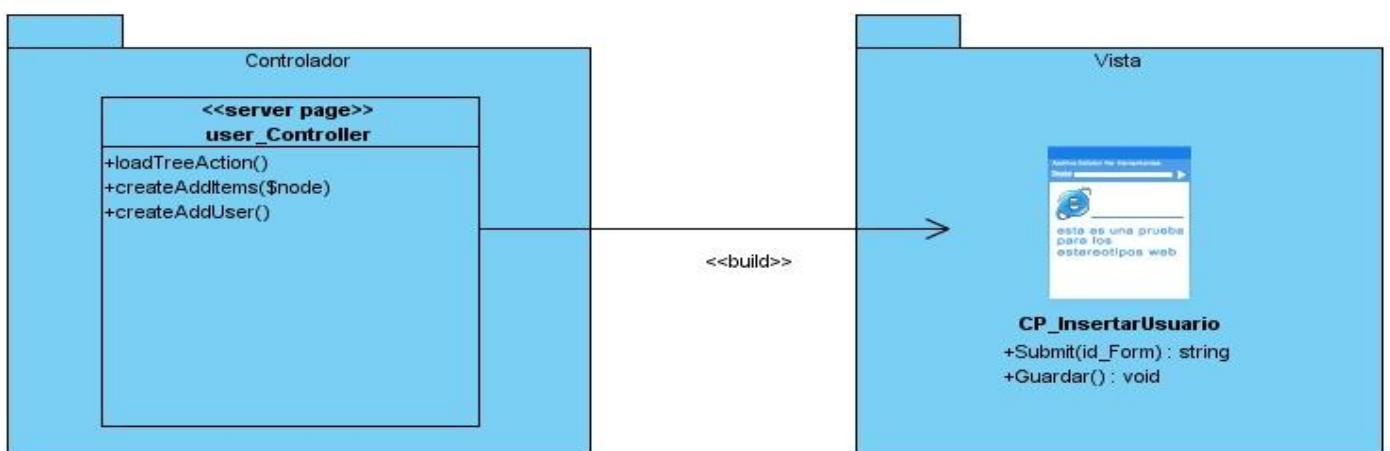


Figura 14. Ejemplo del patrón Alta Cohesión.

Controlador: Las peticiones web son manipuladas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación, ya que gestiona todas las peticiones realizadas por el usuario. Ejemplo básico que evidencia de forma clara, su utilización en Symfony. Este maneja las peticiones del usuario, la seguridad y carga la configuración de la aplicación. En busca de disminuir un poco la carga que este posee se cuentan con las acciones que contienen las especificaciones de cada página.

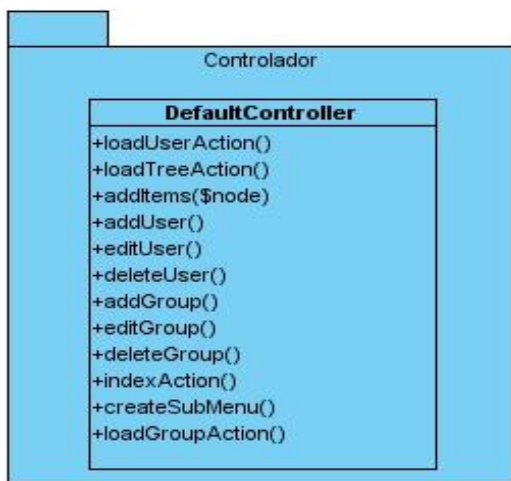


Figura15. Ejemplo del patrón Controlador.

GOF: Se descubren a raíz del libro "Design Patterns", obra pionera en la que se presentan 23 patrones de gran utilidad durante el desarrollo de software orientado a objetos. Son conocidos por este nombre ya que el libro fue escrito por cuatro autores; según el propósito con el que son utilizados se clasifican en tres categorías: creacionales, estructurales y de comportamiento. (35)

Envoltorio (Decorador): Es un patrón estructural que sirve para diseñar las interconexiones entre los objetos, a la vez que estudia cómo estos se relacionan en tiempos de ejecución (35). Su utilización se evidencia en el desarrollo del módulo debido a que el framework Symfony presenta un archivo denominado layout.php o también conocido como plantilla global. Este guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. De esta forma el contenido de cada interfaz se integra a la plantilla global, obteniéndose la interfaz final, evidenciándose así una implementación del patrón Decorador.

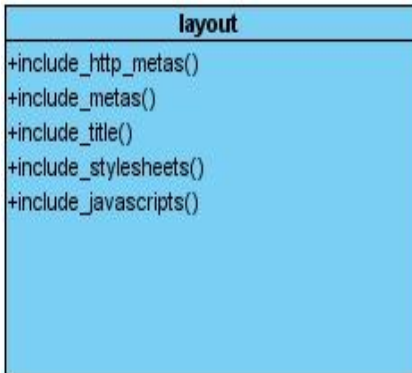


Figura 16. Ejemplo del patrón Decorador.

3.5 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 (36).

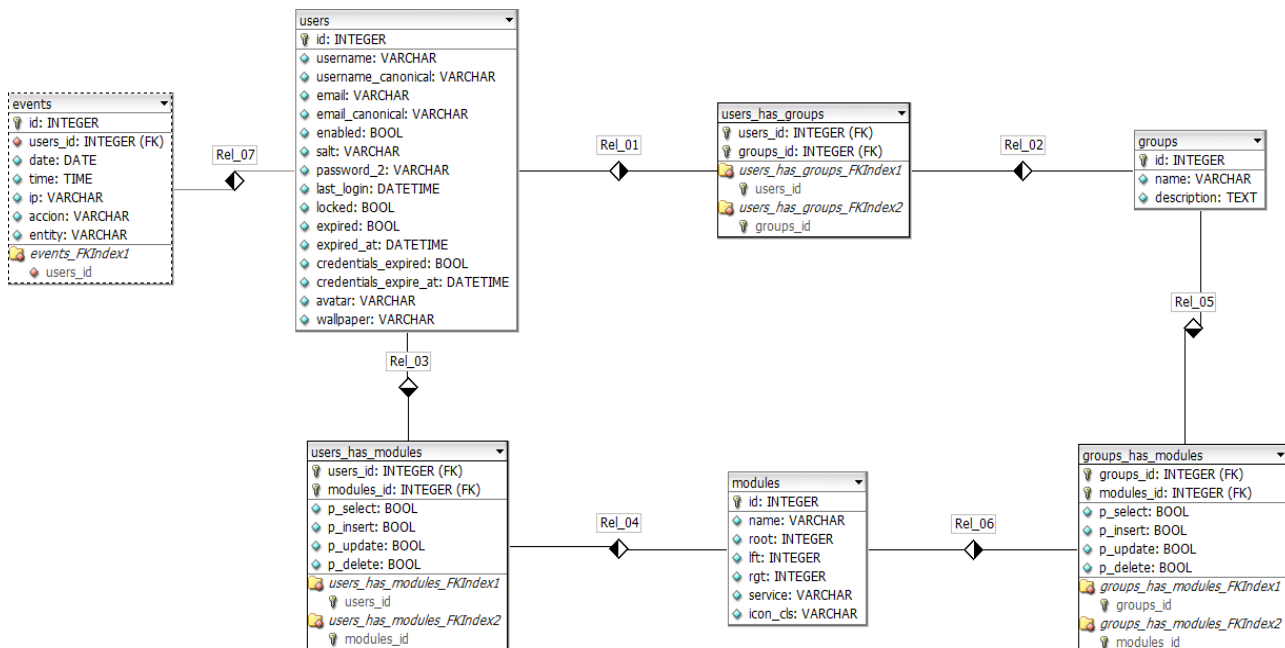


Figura 17. Modelo de datos del Módulo administración, configuración y seguridad.

3.5.1 Descripción de las tablas del modelo de Datos

A continuación se muestra, para cada una de las entidades del Módulo administración, configuración y seguridad, la descripción de sus atributos, dónde se utilizan los términos llave primaria (PK, por sus siglas en inglés) y llave foránea (FK, por sus siglas en inglés).

Tabla 7. Descripción de la tabla Usuarios.

Nombre: users		
Descripción: Almacena información referente a todos los usuarios del sistema.		
Atributo	Tipo	Descripción
Id(PK)	integer	Identificador único de la entidad.
username	varchar	Atributo único que identifica el usuario.
username_canonical	varchar	Nombre de usuario de forma
email	varchar	Dirección de correo del usuario
email_canonical	varchar	Dirección de correo del usuario de forma
enabled	bool	Describe si el usuario está actualmente en uso.
salt	varchar	Se usa de ayuda para codificar la contraseña.
password	varchar	Describe la contraseña encriptado del usuario.
lastLogin	datetime	Describe la última vez que el usuario inicio sección.
expired	bool	Verifica si el usuario ya expiró
locked	bool	Verifica si el usuario está bloqueado.
expired_at	datetime	Verifica si la contraseña del usuario ya expiró.
credentials_expired	bool	Describe si el credencial del usuario ya expiró.
credentials_expired_at	datetime	Muestra la fecha en que expira el credencial del usuario.

avatar	varchar	Muestra la imagen que el usuario elija.
wallpaper	varchar	Describe el fondo de escritorio.

Tabla 8. Descripción de la tabla Grupos.

Nombre: groups		
Descripción: Almacena información referente a todos los grupos existente en el sistema.		
Atributo	Tipo	Descripción
Id(PK)	integer	Identificador único de la entidad.
name	varchar	Atributo único de la entidad que identifica el grupo.
description	text	Describe las funcionalidades del grupo.

Tabla 9. Descripción de la tabla Usuarios y Grupos.

Nombre: users_has_groups		
Descripción: Almacena información referente a todos los usuarios que pertenecen a un grupo y viceversa.		
Atributo	Tipo	Descripción
users_id(FK)	integer	Identificador asociado a los usuarios.
group_id(FK)	integer	Identificador asociado a los grupos.

Tabla 10. Descripción de la tabla Módulos.

Nombre: modules		
Descripción: Almacena información referente a los módulos existentes en la aplicación.		
Atributo	Tipo	Descripción
Id(PK)	integer	Identificador único de la entidad.

name	varchar	Atributo único de la entidad que identifica al módulo.
root	integer	Atributo que especifica quien es el padre del elemento que lo contenga.
lft	integer	Rango mínimo que ayuda a definir quienes son los hijos del objeto creado por la entidad.
rgt	integer	Rango máximo que ayuda a definir quiénes son los hijos del objeto creado por la entidad.
service	varchar	Define el servicio de la entidad para usar dentro de la aplicación
icon_ds	varchar	Propiedad del Ext JS que define un icono a la entidad.

Tabla 11. Descripción de la tabla Usuarios y Módulos.

Nombre: users_has_modules		
Descripción: Almacena información referente a los permisos asignados a los diferentes usuarios del sistema sobre cada módulo.		
Atributo	Tipo	Descripción
users_id(FK)	integer	Identificador asociado a los usuarios.
modules_id(FK)	integer	Identificador asociado a los módulos.
p_select	bool	Define si el usuario tiene permisos de seleccionar sobre el módulo especificado.
p_insert	bool	Define si el usuario tiene permisos de insertar sobre el módulo especificado.
p_update	bool	Define si el usuario tiene permisos de actualizar sobre el módulo especificado.
p_delete	bool	Define si el usuario tiene permisos de eliminar sobre el módulo especificado.

Tabla 12. Descripción de la tabla Grupos y Módulos.

Nombre: groups_has_modules		
Almacena información referente a los permisos asignados a los diferentes grupos del sistema sobre cada módulo.		
Atributo	Tipo	Descripción
group_id(FK)	integer	Identificador asociado a los grupos.
modules_id(FK)	integer	Identificador asociado a los módulos.
p_select	bool	Define si el grupo tiene permisos de seleccionar sobre el módulo especificado.
p_insert	bool	Define si el grupo tiene permisos de insertar sobre el módulo especificado.
p_update	bool	Define si el grupo tiene permisos de actualizar sobre el módulo especificado.
p_delete	bool	Define si el grupo tiene permisos de eliminar sobre el módulo especificado.

Tabla 13. Descripción de la tabla Eventos.

Nombre: events		
Almacena información referente a los eventos realizados por el los usuarios del sistema.		
Atributo	Tipo	Descripción
id(PK)	integer	Identificador único de la entidad.
users_id(FK)	integer	Identificador asociado a los usuarios.
accion	varchar	Atributo que especifica que acción realiza el usuario.
entity	varchar	Describe sobre qué entidad el usuario realiza la

		acción.
element	integer	Describe el elemento sobre el que se realiza la acción.

La realización del presente capítulo demostró los resultados obtenidos durante la etapa de implementación del Módulo administración, configuración y seguridad del SIGICEM v2.0. La identificación de los estándares y estilos de codificación a utilizar, permitieron la correcta implementación de los componentes definidos y las opciones a efectuar para llevar a cabo el tratamiento de errores y la seguridad de acceso. Además, quedó definida la distribución física de los nodos que componen al sistema a través del diagrama de despliegue.

CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA

A partir del modelado de la propuesta en el capítulo anterior se representa el diagrama de despliegue y de componentes en correspondencia con la implementación del sistema. Además, se exponen algunos de los criterios seguidos para el tratamiento de errores, la gestión de la seguridad y las estrategias de codificación y se explican los estándares y estilos utilizados en esta fase de desarrollo.

4.1. Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Es un grafo de nodos unidos por conexiones de comunicación. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. (37)

Para la estructuración física del sistema se utilizó un nodo PC_Cliente que representa la computadora que contiene las funcionalidades ofrecidas al usuario. La misma se conecta mediante el protocolo de Transferencia de Hipertexto (*http* por sus siglas en inglés) al nodo Servidor Web, en el cual se encontrará el Sistema de Gestión para la Ingeniería Clínica y Electromedicina v2.0. Desde este 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, en la estación PC_Cliente se podrá conectar una impresora mediante el puerto USB.

En la figura 22 se muestra el diagrama de despliegue antes descrito.

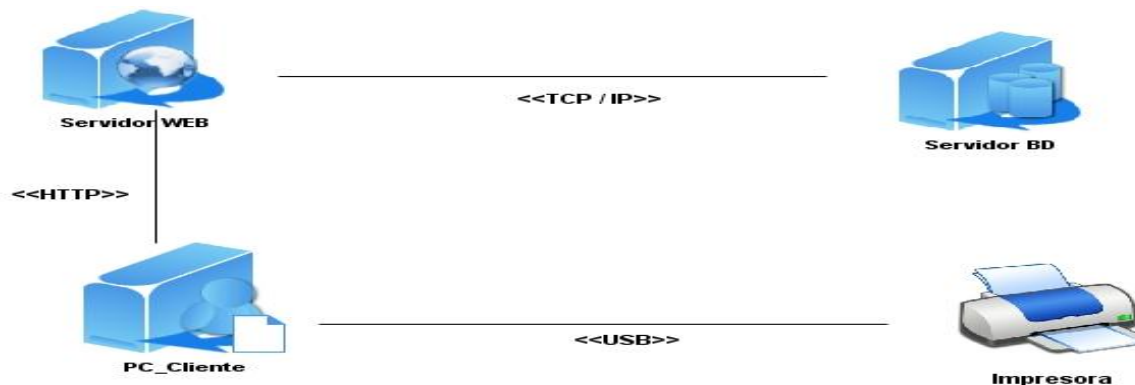


Figura 22. Diagrama de despliegue para el Módulo administración, configuración y seguridad.

Requerimientos de Hardware

➤ **Nodo PC Cliente**

Computador Pentium a 333 MHz o superior, 512 MB RAM o superior, MODEM o red con TCP-IP para conexión al servidor.

➤ **Nodo Servidor Web**

Computador Pentium a 2.8 GHz o superior, 1 GB RAM o superior, 1 GB de espacio libre en Disco Duro como mínimo.

➤ **Nodo Servidor de Base de Datos**

Computador Pentium a 2.8 GHz o superior, 2 GB RAM o superior, 10 GB de espacio libre en Disco Duro como mínimo.

4.2. Diagrama de componentes

Los diagramas de componentes representan la estructura física del código, asignan la vista lógica de las clases del proyecto a los archivos que contienen el código fuente en el que se implementa la lógica (38). A continuación se muestra el diagrama de componentes del módulo administración, configuración y seguridad del SIGICEM v2.0, conformado por los paquetes Vista, Controlador y Modelo, donde se agrupan otros componentes más pequeños en dependencia de las funciones que realizan.

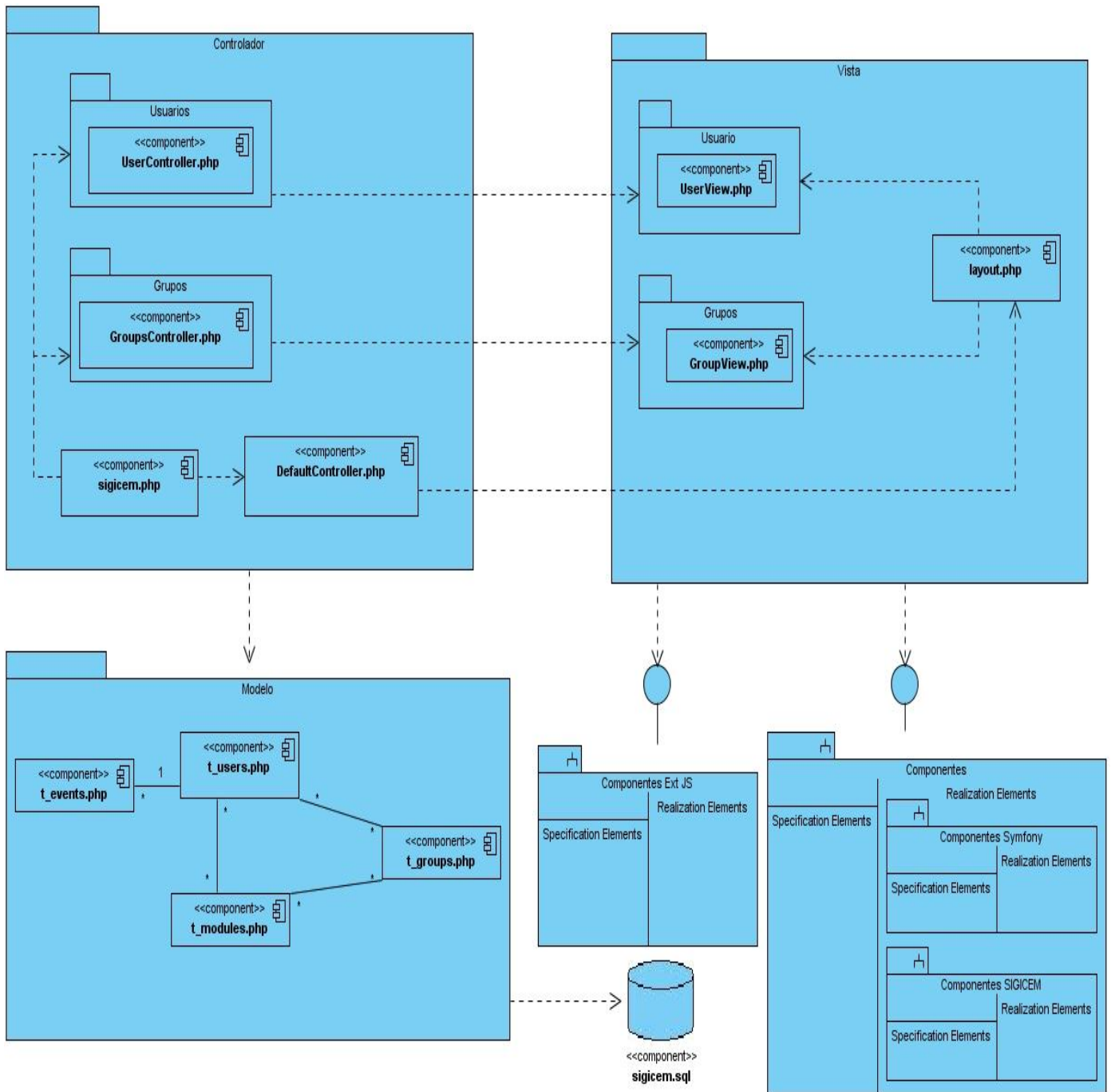



Figura 23. Diagrama de componentes.

4.3. Tratamiento de errores

El tratamiento de errores es de vital importancia en un sistema para evitar que sucedan situaciones adversas que se conviertan luego en problemas mayores que afecten la funcionalidad del mismo. Además garantizar la integridad y confidencialidad de la información son aspectos fundamentales que los sistemas deben de cumplir para que posean una correcta elaboración. A continuación se muestran algunas imágenes del tratamiento de errores en el módulo de Administración, Configuración y Seguridad del SIGICEM v2.0.



The image shows a web application window titled "Insertar Usuario" (Add User). The window has a blue header with a user icon and standard window controls (minimize, maximize, close). Below the header is a form with several fields and options:

- Datos** (collapsible section):
 - Usuario:** ypdieguez
 - Contraseña:** masked with dots
 - Repetir:** masked with dots
 - Correo:** not_email (highlighted with a red border and a red exclamation mark icon, indicating an error)
 - Expira:** empty field with a calendar icon
 - Estado:** Activo, Expiración de credenciales, Super-Admin
- Permisos** (collapsible section with a plus icon)
- Grupos** (collapsible section with a plus icon)

At the bottom of the form are two buttons: "Guardar" (Save) and "Cancelar" (Cancel).

Figura 24. Ejemplo relacionado con el tratamiento de errores.



Figura 25. Ejemplo relacionado con el tratamiento de errores.



Figura 26. Ejemplo relacionado con el tratamiento de errores.

4.4. Seguridad

Uno de los aspectos más importantes en el desarrollo de cualquier aplicación es la seguridad de acceso, para ello Symfony 2 dispone de una moderna librería que se encarga de las validaciones de acceso y seguridad. La autenticación y autorización son los elementos fundamentales de la seguridad en Symfony, el primero se encarga de verificar si el usuario en cuestión está Autenticado (logueado) y se le conoce como “Firewall”, el segundo verifica si el usuario tiene los permisos o “roles” necesarios y se le conoce como “access_control” (39) .

En el Módulo de administración, configuración y seguridad se registrarán los múltiples usuarios que interactúan con la aplicación SIGICEM v2.0. Debido a esto, se utiliza el FOSUserBundle en conjunto con la librería de Symfony 2, que facilitan de forma segura la gestión de usuario y la seguridad de acceso en la aplicación. Mediante su utilización los usuarios tendrán que iniciar sesión para acceder al sistema y a partir de este momento el Módulo administración, configuración y seguridad garantizará, en dependencia de los permisos que tenga cada uno asignado, quienes accederán a las funcionalidades correspondientes del mismo. También en dicho módulo 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: inserción, modificación y eliminación de un atributo perteneciente a una entidad.

4.5 Estándares y estilos de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, es necesario establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente. (40)

A continuación se describen los estándares y estilos utilizados en el Módulo administración, configuración y seguridad del SIGICEM v2.0.

Tabla 14. Estándares y estilos de codificación.

Indentación	
Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.	
Inicio y fin de bloque	El código debe usar cuatro espacios desde la instrucción anterior para el inicio y fin de bloque {}.
Aspectos generales	El código debe usar cuatro espacios para la indentación en vez de usar el tabulado. Esto minimiza problemas con otras herramientas de desarrollo.
Comentarios, separadores, líneas, espacios en blanco y márgenes.	
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada bloque de código. Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro).
Líneas en blanco	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.
Espacios en blanco	Entre operadores lógicos y aritméticos. Se recomienda usar espacios en blanco entre estos

	operadores para una mayor legibilidad en el código. Ejemplo: usuario = nomusuario.
Aspectos generales	<ul style="list-style-type: none"> ➤ Sobre las líneas. <p>Las líneas podrían tener 80 caracteres o menos evitando tener más de 120 caracteres.</p> <ul style="list-style-type: none"> ➤ Sobre el comentario. <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> <ul style="list-style-type: none"> ➤ Sobre los espacios en blanco. <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>
Variables y constantes	
Apariencia de variables	<p>El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCase, sin subrayado para variables. Ejemplo: sNombrePaciente.</p>
Apariencia de las contantes	<p>La declaración de las constantes en una clase debe usar letras</p>

	mayúsculas con subrayado como separador.
Aspectos generales	El nombre empleado en las variables y constantes, debe permitir que con sólo leerlo se conozca el propósito de la misma.
Clases y Objeto	
Apariencia de clases y objetos.	Los nombres de clases deben ser declarados con estilo StudlyCaps . Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula.
Apariencia de atributos	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.
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.
Aspectos generales	Se debe usar el formato de camelCase, sin subrayados para variables, funciones y nombres de métodos y sus argumentos. El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.

Los resultados obtenidos durante la etapa de implementación del Módulo administración, configuración y seguridad para SIGICEM v2.0 fueron satisfactorios. Además, los estándares y estilos de codificación definidos a utilizar en la implementación de los componentes, más las estrategias para llevar a cabo un buen tratamiento de errores quedaron explicados detalladamente en el capítulo, contribuyendo así a garantizar una correcta seguridad, claridad y limpieza del código. Finalmente, se confeccionó el diagrama de componentes y se definió la distribución física de los nodos que componen al sistema a través de la representación del diagrama de despliegue.

CONCLUSIONES GENERALES

Una vez finalizada la investigación para el desarrollo de la solución propuesta, se arriba a las siguientes conclusiones:

- El análisis de los principales referentes teóricos acerca de la administración, configuración y seguridad en aplicaciones web, demostró que existen inconsistencias en los sistemas similares a la investigación a nivel nacional e internacional, que no les permiten integrarse a la aplicación SIGICEM v2.0, por lo que se ratifica el desarrollo del Módulo administración, configuración y seguridad como solución a la problemática planteada.
- Las entrevistas realizadas al grupo de desarrollo, y el estudio de la primera versión del SIGICEM permitieron identificar las particularidades y necesidades del proceso administración, configuración y seguridad del Sistema de Gestión para Ingeniería Clínica y Electromedicina v2.0, lo que permitió obtener una guía para el diseño e implementación de la solución propuesta.
- La correcta asimilación de las herramientas, tecnologías y metodología definidas para el SIGICEM v2.0, permitió la identificación de nuevas bondades principalmente en los frameworks utilizados, lo que posibilitaron la obtención de una aplicación con las características que se esperaban según los requisitos definidos a partir de la incorporación de técnicas actuales y novedosas.
- La confección de los artefactos pertenecientes a las disciplinas de trabajo definidas según el Proceso Unificado de Desarrollo, dieron como resultado la documentación necesaria de para el desarrollo exitoso de la solución.
- Se desarrolló un módulo que permitirá una correcta administración, configuración y seguridad para el Sistema de Gestión de Ingeniería Clínica y Electromedicina v2.0, lo que representa la máxima fortaleza de dicha aplicación y permitiendo la integración de los demás módulos del sistema de forma rápida y sencilla.

RECOMENDACIONES

Para futuras investigaciones se recomienda:

- Integrar el Módulo administración, configuración y seguridad al nuevo núcleo para la arquitectura del Sistema de Gestión para la Ingeniería Clínica y Electromedicina, para su posterior utilización.
- Desarrollar una funcionalidad para el Sistema de Gestión para Ingeniería Clínica y Electromedicina v2.0 que permita recuperar la información que se haya eliminado o modificado de acuerdo a un mal manejo por parte del usuario.
- Permitir que al cerrarse el sistema por inactividad del usuario se salve el estado en que se encontraba la sesión.

REFERENCIAS BIBLIOGRÁFICAS

1. **Velmour Muñoz Casals, Ernesto Rodolfo Castañeda Squires.**SLD069 Sistema para el manejo de órdenes de servicio por los especialistas del Centro de Ingeniería Clínica y Electromedicina. Ciudad de La Habana : Velmour, 2013. isbn: 978-959-7213-02-4 .
2. **Thompson, Ivan.**Definición de Administración. Mexico : s.n., 2008.
3. **Dr.Carlos M. Alvarez de Zayas, Dra Virginia M. Sierra Lombardía.** *Metodología de la Investigación Científica.* La Habana: Editorial Pablo y Educación, 1999.
4. **Pablin Gutiérrez.** Administración de la Informática. S.l.: Blogger, 2009.
5. **Geothesis.** [En Línea] Agosto 01, 2008. [Citado: febrero 21, 2014.] <http://www.geothesis.com/index.php>.
6. **JIMENEZ, Carlos VILLARUBIA.** Seguridad y alta Disponibilidad, Adopción de pautas de seguridad informática. España: Universidad de Castilla-La Mancha.
7. **Pfleeger, Charles P. 2006.** Security in computing. 2006.
8. **Empresa Distribuidora de GeneXus Perú.** GeneXus. SureGX. [En línea] AB&AB, 1992. [Citado: 02 22, 2014.] <http://www.proemsasoftware.com/administracion-de-seguridad-y-auditoria/>.
9. **Carina Centeno Díaz, Danisbel Rojas Rios, Héctor Manuel Solís Mulet.** Componente de Seguridad para aplicaciones del Área Temática Sistemas de Apoyo a la Salud. La Habana: UCI, Facultad 7, 2008.
10. Méndez, Yoan Hernández. Módulo de seguridad para el sistema de negocio EBMS DSerp. La Habana, 2013: UCI, Diseño e implementación de un módulo de seguridad para el sistema de negocio EBMS DSerp.
11. **Ingeniería de Software. RUP (Rational Unified Process).** [En línea] [Citado el: 28 de enero de 2013.] <http://nextech.pe/areas/ingenieria-de-software>.
12. **El Proceso Unificado de Desarrollo de Software.** [En línea] [Citado el: 28 de enero de 2013.] <http://yaqui.mxl.uabc.mx/~molguin/as/RUP.htm>.
13. **B. Mahadevan, David F. Pyke, Moritz Fleischmann.** “Periodic review, push inventory policies for remanufacturing”. [En línea] [Citado el: 20 de Noviembre de 2012.] <http://www.sciencedirect.com/science>.
14. **Aragón, Ismaray Morera.** Módulo de Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina. Sistemas de Apoyo a la Salud, Universidad de las Ciencias Informáticas. La Habana: s.n., 2011. pág. 18, Tesis de pregrado.

15. **Collado Cabeza, Eduardo y Díaz Berenguer, Angi.** [En línea] Madritel, 2003. <http://web.madritel.es/personales3/edcollado/index.html>.
16. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** El Lenguaje Unificado de Modelado. s.l: Pearson Addison-Wesley, 1999, 84-7829-028-1.
17. **SensioLabs.** Symfony. [En línea] 2005. <http://symfony.com/six-good-reasons#>.
18. **Sencha.** Sencha Ext JS JavaScript Framework for Rich Desktop Apps. [En línea] 2014. [Citado: 02 24, 2014.] <http://www.sencha.com/products/complete/>.
19. **Romero, Nelson Martínez.** "Implementación del módulo de Mantenimiento para el Sistema de Gestión para Ingeniería Clínica y Electromedicina". La Habana: UCI, 2013.
20. **NetBeans.** NetBeans Community. [En línea] 2010. [Citado: 02 24, 2014.] https://netbeans.org/community/releases/69/index_es.html.
21. **Gloria Madelaine Fernández Pérez, Daymer Rodríguez Fillad.** Módulo de gestión tecnológica para el sistema de gestión para ingeniería clínica y electromedicina v2.0. La Habana: UCI, 2013.
22. **TECNOLOGÍA y SYNERGIX.** [En línea] [Citado: marzo 25, 2014.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
23. **Agut, Raúl Monferrer.** Especificación de Requisitos Software según. S.l.: E78. INGENIERÍA DEL SOFTWARE, 2000-2001.
24. **Ceria, Santiago.** Casos de Uso. Un Método Práctico para Explorar Requerimientos. [ed.] Cátedra de Ingeniería del Software I. págs. 3-4.
25. **Cáceres, Jesús Tello.** Diagramas de Casos de Uso. [En línea] [Citado el: 01 de abril de 2014.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.
26. **Quintín, Saul Cuzcano.** Scribd. Modelo de Análisis. [En línea] noviembre 13, 2008. [Citado: abril 1, 2014.] <http://es.scribd.com/doc/7930106/Modelo-de-Analisis-Saul-Cuzcano-Quintin>.
27. **Pantigozo, Ing. Antonio Arque.** Modelado del Análisis del Sistema. (UAP), Universidad Alas Peruanas: s.n., 2012.
28. **Yolanda, Ing. Lizana Puelles Esther.** El Rinconcito Informático. [En línea] 06 25, 2000. [Citado: abril 02, 2014.] <http://www.elrinconcito.com/articulos/modeladoUML/modeladoUML.html>.
29. **Armadillo, Integración Tecnológica.** MeRinde. [En línea] Centro Nacional de Tecnologías de Información, 2014. [Citado: abril 04, 2014.] http://merinde.net/index.php?option=com_content&task=view&id=92&Itemid=296.

30. **Rational Unified Process** Extended Help for RUP. 2003.
24. **LIBROSWEB**. Diseño y Programación Web. [En línea] Agosto 2013, 2013. [Citado: Abril 4, 2014.] http://librosweb.es/symfony_1_2/capitulo_2/el_patron_mvc.html.
32. **Joaquín Nicolás Ros, Patrones de diseño**. Departamento de informática y Sistemas. DIS. [En línea] [Citado: abril 5, 2014.] <http://dis.um.es/~jnicolas/CSW/transparencias/Capitulo03.pdf>.
33. **LAMAN, CRAIG**. 1999. UML y Patrones. Introducción al análisis y diseño orientado a objetos. [Traducido.] Luz Maria Henández Rodriguez. México: PRENTICE HALL, 1999. ISBN: 970-17-0261-1.
34. **Marcello Visconti y Hernán Astudillo**. Fundamentos de Ingeniería de software. Departamento de Informática. Universidad Técnica Federico Santa María. [En línea] [Citado: Abril 28, 2014.] www.inf.utfsm.cl/~visconti/ili236/Documentos/01-IntroISw.pdf.
35. **Susana Beatriz Borrero-Arias, Joyce Suárez-Fabre, Adrian Gainza-Huepp, Enrique Machado-Hanze, Dainiel Viltre-Guillen**. Módulo Inventario de Minerales Sólidos para el Sistema de Gestión de Datos Geológicos v2.0.. No.12, UCI, La Habana, Cuba : Ediciones Futuro, Diciembre, 2013, Vol. 6. ISSN: 2306-2495 | RNPS: 2343.
36. **Acuña, Kareenny Brito**. Desarrollo para Aplicaciones Web. ISBN-13: 978-84-692-6641-0.
37. **Hugo Michael, Marca Huallpara**. Diagrama de Despliegue. <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc>
38. **Altova**. Diagramas de componentes UML. [En línea] 2005-2014. [Citado: Abril 28, 2014.] <http://www.altova.com/es/umodel/uml-component-diagrams.html>. ATU66731218.
39. **Christian Van Der Henst**. Maestros del Web. Symfony Framework PHP orientado a objetos. [En línea] Julio 19, 1997. <http://www.maestrosdelweb.com/editorial/curso-symfony2-seguridad-de-acceso>.
40. **Giovanny Fernández**. el Guille, la Web del Visual Basic, C#, .NET y más. Estándar codificación DOTNET. [En línea] Abril 30, 2005. [Citado: Abril 29, 2014.] http://www.elguille.info/colabora/NET2005/giovannyfernandez_EstandarCodificacionNET.htm.

BIBLIOGRAFÍA

- Abarca, Roger.** Scribd. Scribd. [En línea] 03 de Octubre de 2012. [Citado el: 16 de Febrero de 2014.] <http://es.scribd.com/doc/108816828/El-proceso-de-investigacion-cientifica>.
- Acuña, Karenny Brito.** Desarrollo para Aplicaciones Web. ISBN-13: 978-84-692-6641-0.
- Agut, Raúl Monferrer.** Especificación de Requisitos Software según. S.I.: E78. INGENIERÍA DEL SOFTWARE, 2000-2001.
- Altova.** Diagramas de componentes UML. [En línea] 2005-2014. [Citado el: Abril 28, 2014.] <http://www.altova.com/es/umodel/uml-component-diagrams.html>. ATU66731218.
- Aragón, Ismaray Morera.** Módulo de Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina. Sistemas de Apoyo a la Salud, Universidad de las Ciencias Informáticas. La Habana: s.n., 2011. pág. 18, Tesis de pregrado.
- Armadillo, Integración Tecnológica.** MeRinde. [En línea] Centro Nacional de Tecnologías de Información, 2014. [Citado el: 04 de abril de 2014.] http://merinde.net/index.php?option=com_content&task=view&id=92&Itemid=296.
- B. Mahadevan, David F. Pyke, Moritz Fleischmann.** “Periodic review, push inventory policies for remanufacturing”. [En línea] [Citado el: 20 de Noviembre de 2012.] <http://www.sciencedirect.com/science>.
- Cáceres, Jesús Tello.** Diagramas de Casos de Uso. [En línea] [Citado el: 01 de abril de 2014.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.
- Carina Centeno Díaz, Danisbel Rojas Rios, Hector Manuel Solís Mulet.** Componente de Seguridad para aplicaciones del Área Temática Sistemas de Apoyo a la Salud. La Habana : UCI, Facultad 7, 2008.
- Ceria, Santiago.** Casos de Uso. Un Método Práctico para Explorar Requerimientos. [ed.] Cátedra de Ingeniería del Software I. págs. 3-4.
- Christian Van Der Henst.** Maestros del Web. Symfony Framework PHP orientado a objetos. [En línea] Julio 19, 1997. <http://www.maestrosdelweb.com/editorial/curso-symfony2-seguridad-de-acceso/>.
- Collado Cabeza, Eduardo y Díaz Berenguer, Angi.** [En línea] Madritel, 2003. <http://web.madritel.es/personales3/edcollado/index.html>.
- Dr.Carlos M. Alvarez de Zayas, Dra Virginia M. Sierra Lombardía.** *Metodología de la Investigación Científica*. La Habana : Pablo y Educación, 1999.
- El Proceso Unificado de Desarrollo de Software.** [En línea] [Citado el: 28 de enero de 2013.] <http://yaqui.mx/uabc.mx/~molguin/as/RUP.htm>.

- Empresa Distribuidora de GeneXus Perú.** GeneXus. SureGX. [En línea] AB&AB, 1992. [Citado el: 22 de 02 de 2014.] <http://www.proemasoftware.com/administracion-de-seguridad-y-auditoria/>.
- Geothesis.** [En línea], 01 de Agosto de 2008. [Citado el: 21 de febrero de 2014.] <http://www.geothesis.com/index.php>.
- Giovanny Fernández.** el Guille, la Web del Visual Basic, C#, .NET y más. Estándar codificación DOTNET. [En línea], Abril 30, 2005. [Citado el: Abril 29, 2014.] http://www.elguille.info/colabora/NET2005/giovannyfernandez_EstandarCodificacionNET.htm.
- Gloria Madeleine Fernández Pérez, Daymer Rodríguez Fillad.** Módulo de gestión tecnológica para el sistema de gestión para ingeniería clínica y electromedicina. La Habana: UCI, 2013. http://www.elguille.info/colabora/NET2005/giovannyfernandez_EstandarCodificacionNET.htm.
- Hugo Michael, Marca Huallpara.** Diagrama de Despliegue. <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc>
- Ing. Antonio Arqque Pantigozo.** Modelado del Análisis del Sistema. (UAP), Universidad Alas Peruanas : s.n., 2012.
- Ingeniería de Software.** RUP (Rational Unified Process). [En línea] [Citado el: 28 de enero de 2013.] <http://nextech.pe/areas/ingenieria-de-software>.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** El Lenguaje Unificado de Modelado. s.l: Pearson Addison-Wesley, 1999. 84-7829-028-1.
- JIMENEZ, Carlos VILLARUBIA.** Seguridad y alta Disponibilidad, Adopción de pautas de seguridad informática. España: Universidad de Castilla-La Mancha.
- Joaquín Nicolás Ros, Patrones de diseño.** Departamento de informática y Sistemas. DIS. [En línea] [Citado: abril 5, 2014.] <http://dis.um.es/~jnicolas/CSW/transparencias/Capitulo03.pdf>.
- LAMAN, CRAIG.** 1999. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Luz María Hernández Rodríguez. México: PRENTICE HALL, 1999. ISBN: 970-17-0261-1.
- Larman, Craig.** UML y Patrones Segunda Edición. Segunda Edición. Canadá : s.n. pág. 172.
- LIBROSWEB.** Diseño y Programación Web. [En línea] 2013 de Agosto de 2013. [Citado el: 4 de Abril de 2014.] http://librosweb.es/symfony_1_2/capitulo_2/el_patron_mvc.html.
- M. en C. Roberto Hernández Sampieri, DR. Carlos Fernández Collado, Dra Pilar Baptista Lucia.** Metodología de la Investigación. Mexico : Editorial Mexicana, 1998. BN 968-422-931-3.

- Marcello Visconti y Hernán Astudillo.** Fundamentos de Ingeniería de software. Departamento de Informática. Universidad Técnica Federico Santa María. [En línea] [Citado el: Abril 28, 2014.] www.inf.utfsm.cl/~visconti/ili236/Documentos/01-IntroISw.pdf.
- NetBeans.** NetBeans Community. [En línea] 2010. [Citado el: 24 de 02 de 2014.] https://netbeans.org/community/releases/69/index_es.html.
- Pablin Gutiérrez.** Administración de la Informática. s.l. : Blogger, 2009.
- Pfleeger,** Charles P. 2006. Security in computing. 2006.
- Rational Unified Process,** Extended Help for RUP. 2003.
- Rolando Alfredo Hernández León, Sayda Caello González.** EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. La Habana : Editorial Universitaria, 2002. ISBN: 959-16-0343-6 .
- Romero, Nelson Martínez.** Implementación del módulo de Mantenimiento para el Sistema de Gestión para Ingeniería Clínica y Electromedicina. La Habana : UCI, 2013.
- Saul Cuzcano Quintin.** Scribd. Modelo de Analisis. [En línea] 13 de noviembre de 2008. [Citado el: 1 de abril de 2014.] <http://es.scribd.com/doc/7930106/Modelo-de-Analisis-Saul-Cuzcano-Quintin>.
- Sencha.** Sencha Ext JS JavaScript Framework for Rich Desktop Apps. [En línea] 2014. [Citado el: 24 de 02 de 2014.] <http://www.sencha.com/products/complete/>.
- SensioLabs.** Symfony. [En línea] 2005. <http://symfony.com/six-good-reasons#>.
- Susana Beatriz Borrero-Arias, Joyce Suárez-Fabre, Adrian Gainza-Huepp, Enrique Machado-Hanze, Dainiel Viltre-Guillen.** Módulo Inventario de Minerales Sólidos para el Sistema de Gestión de Datos Geológicos v2.0. No.12, UCI, La Habana, Cuba : Ediciones Futuro, Diciembre, 2013, Vol. 6. ISSN: 2306-2495 | RNPS: 2343.
- TECNOLOGÍA y SYNERGIX.** [En línea] [Citado el: 25 de 03 de 2014.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
- Tello, Jesús Cáceres.** Diagramas de Casos de Uso. [En línea] [Citado el: 01 de Abril de 2014.]
- Thompson, Ivan.** Definición de Administración. Mexico : s.n., 2008.
- Velmour Muñoz Casals, Ernesto Rodolfo Castañeda Squires.** SLD069 Sistema para el manejo de órdenes de servicio por los especialistas del Centro de Ingeniería Clínica y Electromedicina. Ciudad de La Habana : Velmour, 2013. isbn: 978-959-7213-02-4 .

Yoan Hernández Méndez. Módulo de seguridad para el sistema de negocio EBMS DSerp. La Habana, 2013 : UCI, Diseño e implementación de un módulo de seguridad para el sistema de negocio EBMS DSerp.

Yolanda, Ing. Lizana Puelles Esther. El Rinconcito Informático. [En línea] 25 de 06 de 2000. [Citado el: 02 de abril de 2014.] <http://www.elrinconcito.com/articulos/modeladoUML/modeladoUML.html>.

ANEXOS

Anexo 1. Guía de Observación. Aspectos a tener en cuenta.

Objetivo: Caracterizar el estado de la administración, configuración y seguridad en el SIGICEM v1.1.

1. Administración y seguridad del Módulo administración del SIGICEM v1.1.
2. Configuración en el SIGICEM v1.1 para la navegación personalizada del usuario en el nuevo módulo.
3. Funcionalidades en el SIGICEM v1.1 que le permitan a los usuarios definir accesos directos, iconografía y entorno escritorio.
4. Dependencias en la gestión de trazas del Módulo administración del SIGICEM v1.1.
5. Incorporación de módulos al sistema.
6. Estrategias de integración de los frameworks en el SIGICEM v1.1.
7. **Anexo 2.** Guía con aspectos a considerar en las entrevistas.

Objetivo: Obtener criterios acerca de la administración, configuración y seguridad para la nueva versión del SIGICEM.

1. Características del Módulo administración del SIGICEM v1.1 que deben prevalecer en la nueva versión.
2. Características que no se encuentran presentes en la primera versión del SIGICEM y sean necesarias agregarlas a la segunda.
3. Aspectos de la configuración que se deben manejar en el sistema.
4. Perfeccionamiento de la gestión de trazas en el nuevo módulo.
5. Otros elementos que se desean considerar acerca de la administración, configuración y seguridad para SIGICEM v2.0.
6. Deseos de estrategias de integración de los frameworks basada en componentes, para una mayor seguridad, integridad y escalabilidad del sistema.

Anexo 3. Prototipo de interfaz de usuario CU Autenticar usuario.

Logo de XAVIA SIGICEM. Sistema de Gestión para Ingeniería Clínica y Electromedicina.

Usuario:

Contraseña:

Entrar

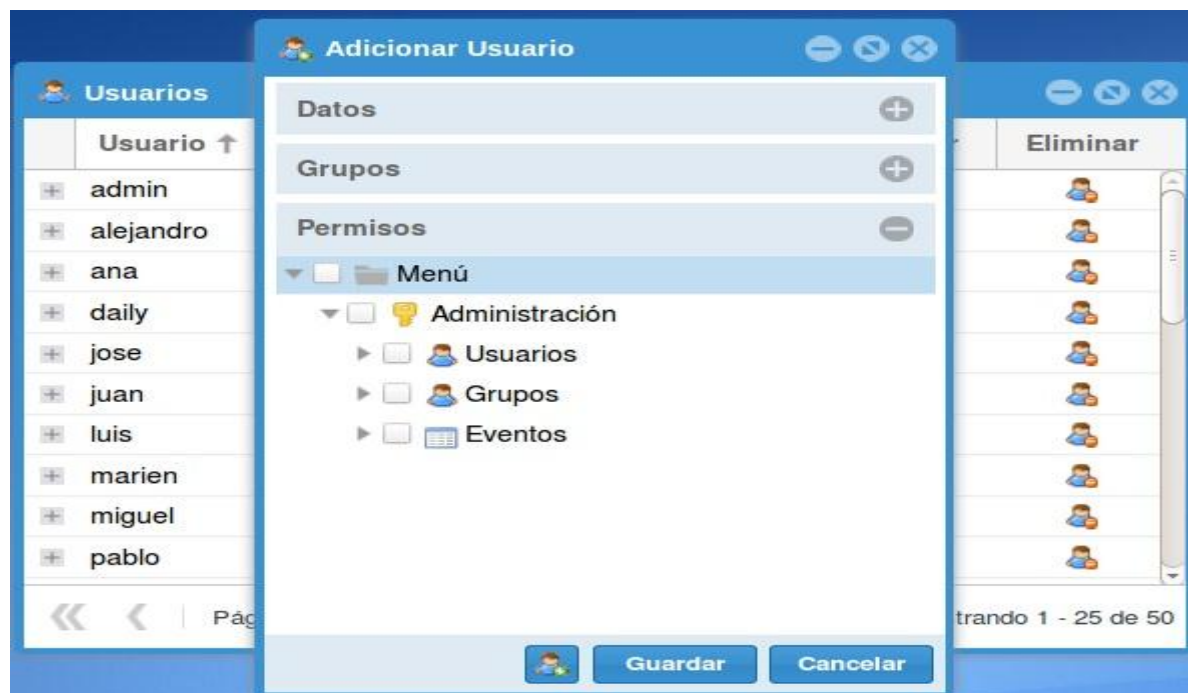
Anexo 4. Prototipo de interfaz de usuario CU Visualizar usuarios.

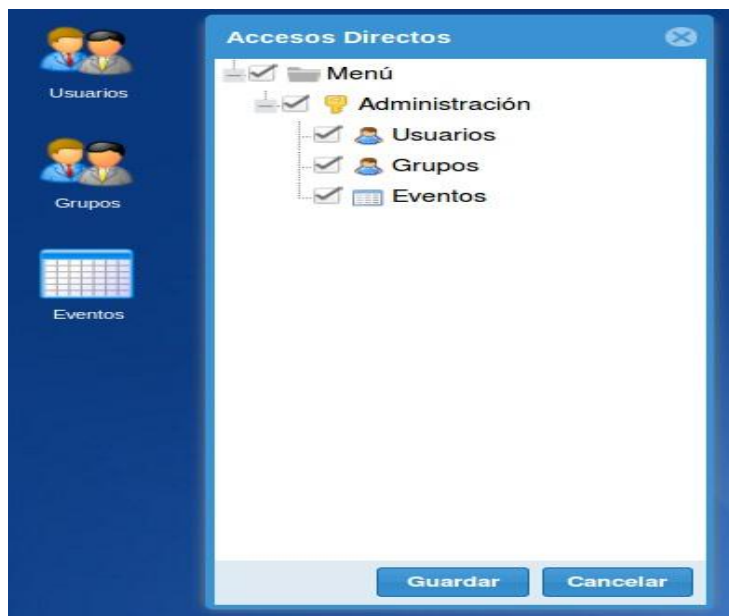
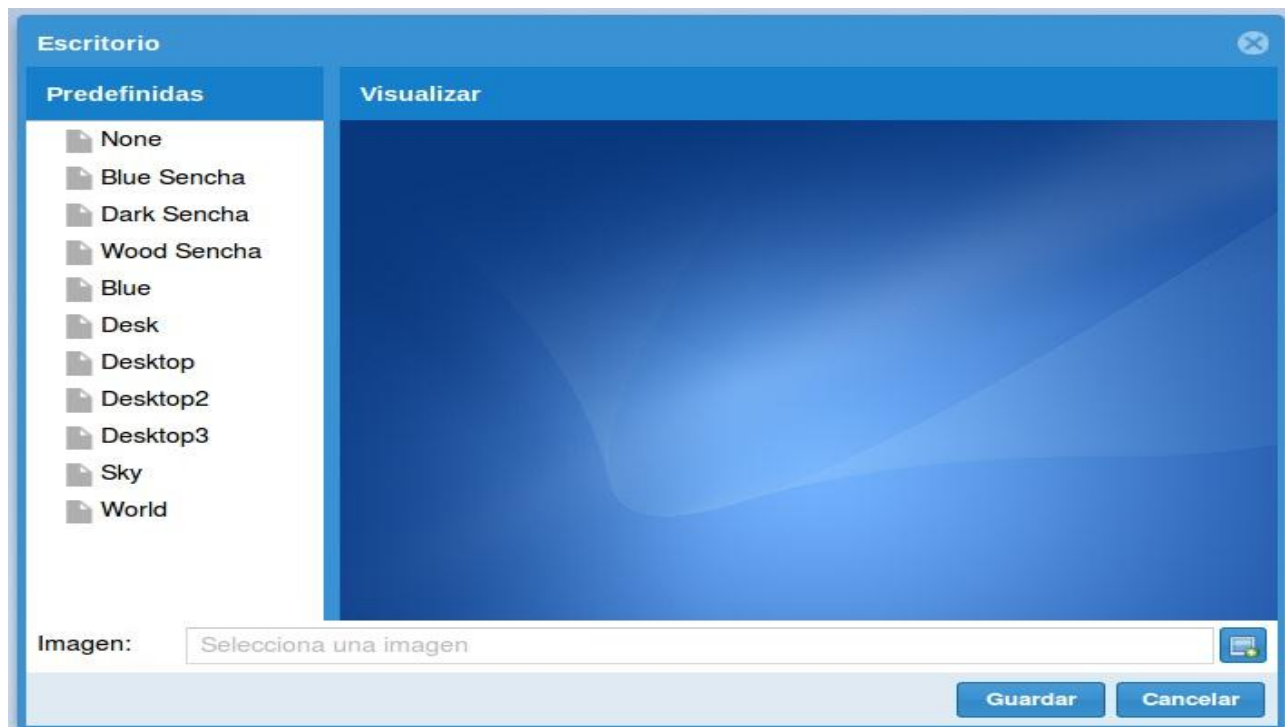
Usuarios

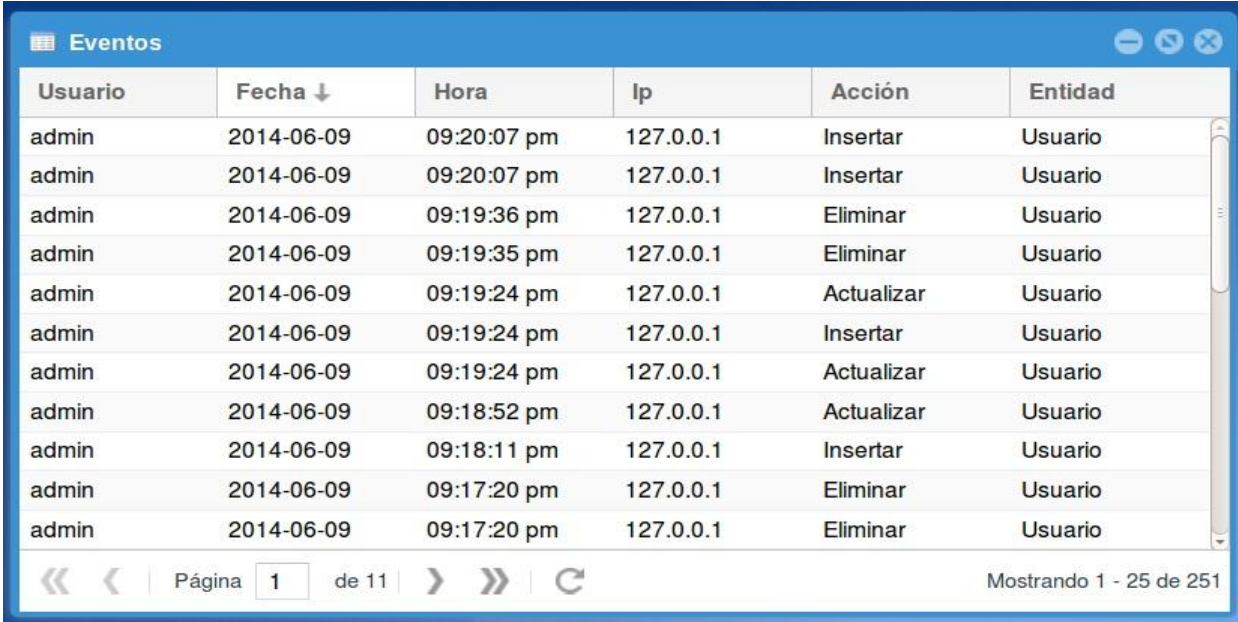
Usuario ↑	Editar	Eliminar
admin		
alejandro		
ana		
daily		
jose		
juan		
luis		
marien		
miguel		
pablo		

« « | Página 1 de 2 | » » | ↻

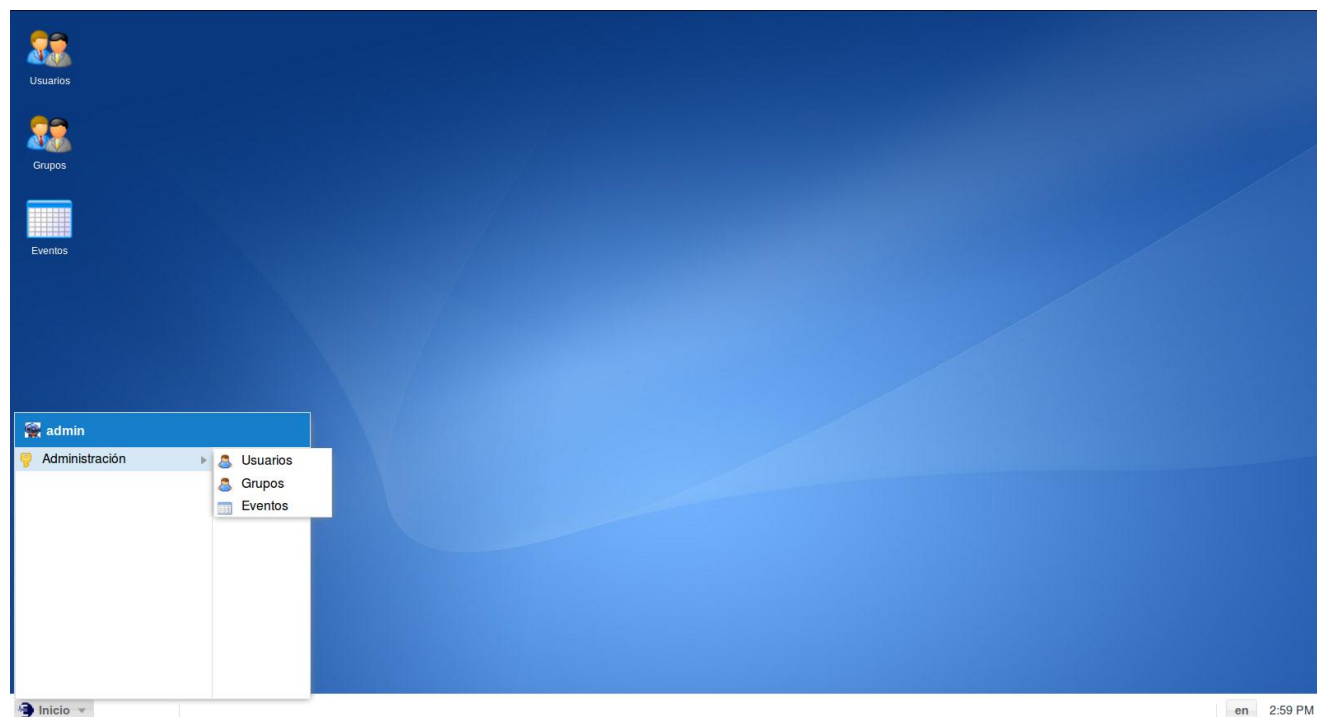
Mostrando 1 - 25 de 50

Anexo 5. Prototipo de interfaz de usuario CU Insertar usuario.**Anexo 6.** Prototipo de interfaz de usuario CU Asignar permisos a usuario.

Anexo 7. Prototipo de interfaz de usuario CU Personalizar accesos directo.**Anexo 8.** Prototipo de interfaz de usuario CU Personalizar fondo escritorio.

Anexo 10. Prototipo de interfaz de usuario CU Gestión de trazas.

Usuario	Fecha ↓	Hora	Ip	Acción	Entidad
admin	2014-06-09	09:20:07 pm	127.0.0.1	Insertar	Usuario
admin	2014-06-09	09:20:07 pm	127.0.0.1	Insertar	Usuario
admin	2014-06-09	09:19:36 pm	127.0.0.1	Eliminar	Usuario
admin	2014-06-09	09:19:35 pm	127.0.0.1	Eliminar	Usuario
admin	2014-06-09	09:19:24 pm	127.0.0.1	Actualizar	Usuario
admin	2014-06-09	09:19:24 pm	127.0.0.1	Insertar	Usuario
admin	2014-06-09	09:19:24 pm	127.0.0.1	Actualizar	Usuario
admin	2014-06-09	09:18:52 pm	127.0.0.1	Actualizar	Usuario
admin	2014-06-09	09:18:11 pm	127.0.0.1	Insertar	Usuario
admin	2014-06-09	09:17:20 pm	127.0.0.1	Eliminar	Usuario
admin	2014-06-09	09:17:20 pm	127.0.0.1	Eliminar	Usuario

Anexo 9. Prototipo de interfaz de usuario CU Pantalla principal.

GLOSARIO DE TÉRMINOS

- **camelCase:** es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello. El nombre CamelCase se podría traducir como Mayúsculas/Minúsculas Camello.
- **notación CamelCasing:** Define que el primer carácter de todas las palabras, excepto la primera palabra se expresa en mayúscula y el resto de los caracteres en minúscula, por ejemplo: customerOrder
- **StudyCaps:** es una forma de notación de texto en el que la capitalización de letras varía según algún patrón, o arbitrariamente, por lo general también la omisión de los espacios entre las palabras y muchas veces omitiendo algunas letras, por ejemplo: StUdLyCaPs o STuDLyCaPS. Estos patrones son identificados por muchos usuarios, de forma ambigua, como camel case. La alternativa típica es simplemente reemplazar los espacios por guiones bajos (como en el caso de la serpiente).
- **PascalCasing:** Establece que el primer carácter de todas las palabras se expresa en mayúscula y el resto de los caracteres en minúscula, por ejemplo: CustomerOrder
- **Software:** conjunto de los programas de cómputo, procedimientos, regla, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.