

**Universidad de las Ciencias Informáticas**

**FACULTAD 7**



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

**Rediseño e implementación del Portal de la  
Sociedad Cubana de Informática Médica Versión 1.0**

**Autor:** Javier López González

**Tutor:** Ing. Renier Ramos Oliva

**Cotutor:** Ing. Dannier Flores Ramos

La Habana, julio 2011

“Año del 53 de la Revolución”

## DATOS DE CONTACTO.

**Renier Ramos Oliva:** Graduado de Ingeniería en Ciencias Informáticas en el año 2008 con Título de Oro. Actualmente labora en la Universidad de Ciencias Informáticas (UCI), desempeñándose como profesor de la Facultad No. 7. Ha participado en diferentes eventos científicos del centro obteniendo resultados relevantes. Presenta la categoría docente de Instructor. Actualmente es parte del grupo de mercadotecnia del Centro de Informática Médica (CESIM) donde se desempeña como diseñador. Ha cursado varios cursos de postgrado y publicaciones nacionales e internacionales como parte de su superación profesional. **Correo electrónico: rramos@uci.cu**

**Dannier Flores Ramos:** Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2009. Profesor adiestrado del departamento de Sistemas de Apoyo a la Salud del Centro de Informática Médica, CESIM. **Correo electrónico: dflores@uci.cu**

## AGRADECIMIENTOS.

Con estas palabras inicio mis agradecimientos "le pido a dios que me alcance la vida y me tiempo para regresar aunque sea tan solo un poco de lo mucho que me das, le pido a dios que me alcance la vida para decirte todo lo que siento es gracias a tu amor". La fuente de mi inspiración de agradecimiento está a nombre de mis almas mis razones de vivir: mi mamá y mi abuela que no se encuentra hoy con nosotros. Quiero darles las gracias por quererme, confiar en mí y sobre todo por apoyarme. Otras personas no menos importantes son mis familiares sobre todo mi tío Pacolo, mi abuelo, en fin toda mi familia. Mis amistades no puedo dejar de mencionarlas son muchas dentro fuera de la UCI, Nuris, Cundi, Andy, Exabie, Sisi, Lino, Yuley, Oniel, Dany, Adonis, Franklin, Jose Carlos, Meme; en fin la lista es muy grande. No puedo dejar de mencionar a personas que fueron y son muy importantes en mi vida como Dánae y Ari son maravillosas; no se me puede olvidar a mi primi esa que siempre me ayudó que nunca me dijo no que me quiere mucho Lisset eres lo máximo y al Migue que no solo me ayudó en la tesis sino que se convirtió en un gran amigo para mi. Quiero agradecerles a mis tutores(Dannier y Ramos muchas gracias son mis hermanos) y todos mis profes desde la primaria, que a pesar de todo me formaron y crearon en mi un poco de lo que soy hoy. Si alguien se queda fuera de mi lista por favor me disculpe créanme que la amistad se lleva dentro y se demuestra sin decir tan siquiera la palabra amigo, los quiero a todos.

**DEDICATORIA.**

Dedico mi tesis a quien siempre se sintió muy orgullosa de mí y hoy en día no se encuentra con nosotros: mi abuelita, te amo.



*No conozco a nadie que haya llegado a la cima sin trabajo duro. Esa es la receta. No siempre te llevará a la cima, pero te dejará bastante cerca.*

*Margaret Thatcher.*

## RESUMEN.

La Sociedad Cubana de Informática Médica (SOCIM) tiene un portal web que actualmente no permite a sus asociados las inscripciones en línea, la discusión de los temas publicados, el envío y revisión de sus artículos científicos automatizados. Además no cuenta con un mecanismo de control del acceso de los usuarios y del pago de sus cuotas anuales de afiliación. Por tales cuestiones, el objetivo de esta investigación es desarrollar un portal web para la SOCIM donde se gestione toda la información de interés para sus asociados.

Para su desarrollo se adoptó el Proceso Unificado de Desarrollo como metodología, el Lenguaje Unificado de Modelado 2.1 y el Enterprise Architect 7.1 como herramienta de modelado; se empleó el Postgres SQL 8.3 como gestor de base de datos y pgAdmin III como herramienta de administración de la misma. En el diseño de las interfaces se utilizaron el Artisteer 2 y el Fireworks CS4. Como tecnología base fue usado el Sistema de Gestión de Contenidos Drupal 6.15 y el Netbeans 6.9 como herramienta de desarrollo.

Este portal posibilitará el ingreso de un mayor número de investigadores a la SOCIM debido a que el proceso de registro será más fácil, contará con una interfaz más amigable y los asociados tendrán espacios para debatir sobre los temas científicos de su interés. Se podrán gestionar los artículos de forma rápida y eficiente, lo que traerá consigo un crecimiento del conocimiento de la sociedad y un mayor desarrollo investigativo.

**Palabras claves:** Informática Médica, Portal Web, Sistema de Gestión de Contenidos (CMS), Sociedad Cubana de Informática Médica (SOCIM).

**ÍNDICE.**

**INTRODUCCIÓN..... 1**

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ..... 6**

1.1. Sistemas automatizados vinculados al campo de acción..... 6

1.1.1. Soluciones informáticas existentes en el ámbito internacional..... 6

1.1.2. Soluciones informáticas existentes en el ámbito nacional..... 7

1.2. Técnicas, tecnologías, metodologías y software usados para la solución del problema..... 8

1.3 Arquitectura del Sistema..... 17

**CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA ..... 21**

2.1 Modelo de Dominio..... 21

2.1.1. Conceptos fundamentales..... 21

2.2 Requerimientos de software..... 23

2.2.1 Requerimientos funcionales..... 23

2.2.2 Requerimientos no funcionales..... 25

2.3 Descripción del sistema propuesto..... 26

2.3.1 Actores del sistema..... 27

2.3.2 Vista Global de los Actores del Sistema..... 28

2.3.3 Definición de los Casos de Uso..... 28

**CAPÍTULO 3: ANÁLISIS Y DISEÑO..... 34**

3.1 Análisis..... 34

3.1.1 Modelo de Análisis..... 34

3.2 Diseño..... 36

3.2.1 Modelo de Diseño..... 37

3.3	Modelo de Datos .....	48
3.1.1	Descripción de las Tablas.....	50
3.4	Modelo de Despliegue. ....	52
<b>CAPÍTULO 4: IMPLEMENTACIÓN.....</b>		<b>54</b>
4.1	Modelo de Implementación.....	54
4.1.1	Diagrama de Componentes.....	54
<b>CONCLUSIONES.....</b>		<b>59</b>
<b>RECOMENDACIONES.....</b>		<b>60</b>
<b>TRABAJOS CITADOS.....</b>		<b>61</b>
<b>BIBLIOGRAFÍA.....</b>		<b>65</b>
<b>ANEXOS .....</b>		<b>69</b>
	Anexo 1. Fases y flujos de trabajos de la Metodología RUP. ....	69
	Anexo 2. Arquitectura: Cliente-Servidor. ....	70
	Anexo 3. Descripción de la arquitectura. ....	71

## INTRODUCCIÓN.

La Medicina como ciencia, la prestación de servicios de salud, la docencia, la formación de recursos humanos y la investigación son áreas que se encuentran en continuos cambios, que ocurren cada vez con mayor rapidez. Un factor clave de estos cambios, es que se sustentan en las nuevas Tecnologías de la Informática y las Comunicaciones (TIC) a través del uso de las herramientas de comunicación y de la informática médica. Esta última es una disciplina emergente e integradora que surge producto de la intersección sinérgica de la Informática con la Medicina, ella estudia la estructura, el comportamiento y la interacción de los sistemas médicos y las tecnologías de la información.

La informática médica tuvo sus inicios en la década de los años 50, con el crecimiento de los dispositivos computacionales. Sus primeras denominaciones fueron desde computación médica, procesamiento electrónico de datos médicos, procesamiento de la información médica hasta ingeniería de software médico y tecnología computacional médica. Ello necesariamente refleja diferentes momentos de la apreciación del alcance y el enfoque de la actual informática médica. Su rápido avance, en especial, durante los últimos 25 años, se debe al creciente reconocimiento de la imposibilidad de manejar las bases del conocimiento de la medicina con los métodos tradicionales basados en el papel y la convicción de que el proceso de toma de decisiones informadas es importante en la medicina moderna. (1)

La informática médica abarca la creación de sistemas automáticos de diagnóstico, terapia y comunicación de datos médicos, la arquitectura de registros médicos electrónicos u otros sistemas para la facturación, programación e investigación. Abarca además el desarrollo de sistemas de apoyo a la toma de decisiones y el intercambio de información médica, la gestión de los sistemas de clasificación y los vocabularios médicos controlados, y sistemas de clasificación. Por otra parte desarrolla dispositivos portátiles para asistir a los proveedores de salud con sistemas de entrada y recuperación de datos o de apoyo a la toma de decisiones. Dentro de sus aplicaciones más novedosas están el procesamiento de señales (electroencefalografía, electromiografía) e imágenes (radiografía, resonancia magnética, angiografía, tomografía), la telemedicina, así como la comunicación médica basada en la Web e Internet. (2)

El gran número de investigaciones que se realizan en esta nueva área del conocimiento humano es incalculable, por lo que se hace necesario la creación de sociedades que gestionen los avances tecnológicos e investigativos producto del quehacer científico internacional. La Asociación Internacional de

Informática Médica (IMIA por sus siglas en inglés) provee un ente mundial para la salud y la bioinformática que actúa como asociación de asociaciones, cruzando y juntando las asociaciones y sus miembros.

En Cuba, la Sociedad Cubana de Informática Médica (SOCIM), es la organización no gubernamental voluntaria de carácter científico técnico que agrupa a profesionales y entidades afines a la informática médica. Esta es miembro pleno de la Asociación Internacional de Informática Médica (IMIA) y de la Organización Regional para América Latina y el Caribe (IMIA-LAC). La SOCIM, adscripta al Consejo Nacional de Sociedades Científicas del Ministerio de Salud Pública de la República de Cuba, tiene como finalidad integrar, coordinar, apoyar, difundir y desarrollar las diferentes áreas teóricas y de aplicación de la informática en las Ciencias de la Salud dentro del país. (3)

Actualmente, dicha sociedad presenta diversas dificultades para gestionar la información referente a sus asociados y a los resultados de las investigaciones producidas en esta área. Aunque cuenta con un portal web para dar a conocer dichos resultados, este no es altamente funcional y presenta limitaciones que le restan eficacia en la transmisión del conocimiento.

Las inscripciones para ser un asociado a la Sociedad Cubana de Informática Médica se realizan de forma manual y para ello la persona debe presentarse en la sede nacional. Esto trae como consecuencias que científicos e investigadores de otras provincias tengan que viajar largas distancias para la inscripción o que prefieran no hacerlo, lo que impide la socialización de los nuevos conocimientos. Adicionalmente el proceso de inscripción de una persona puede ser complejo y podrían perderse los datos debido a que las planillas se deterioran con el tiempo o pudiesen extraviarse.

El Portal Web SOCIM no permite que los asociados se autentiquen para acceder a los contenidos, por lo que cualquier persona puede interactuar con los mismos; lo que impide que se tenga un registro de qué usuarios están en línea, qué acciones han realizado, qué temas le resultan de más interés a la comunidad, entre otros aspectos. Además los asociados no cuentan con espacios donde puedan aclarar sus dudas y expresar sus criterios acerca de los artículos, cursos o conferencias publicados. Hoy en día esta sociedad científica continúa en la búsqueda por crear otros espacios de intercambio de conocimientos, pues el portal web no cuenta con la posibilidad de realizar foros, comentarios o tener un chat para sus afiliados.

Como condición obligatoria para que una persona sea asociada a la SOCIM y se mantenga como tal, debe pagar una cuota anual. El registro de los pagos se realiza de forma semi-automatizada, ya que se

utiliza un documento Excel. Este documento es incapaz de alertar ante un retraso o tomar alguna acción correctiva. Esto impide que se puedan emprender acciones con los afiliados que tienen pagos atrasados, y permite que estos mantengan sus derechos de publicación.

La principal deficiencia del portal se encuentra en los procesos de envío y revisión de los artículos científicos desarrollados por los asociados; verdadera razón de ser de la sociedad. Actualmente el portal web es completamente estático y no brinda la posibilidad de gestionar los artículos. Por lo que, los asociados los envían por correo, son revisados por los especialistas y publicados como una página estática que no permite crear comentarios ni emitir votos.

En el proceso puede que los artículos no lleguen los correos a los revisores por exceso de tamaño u otras cuestiones, que sean borrados sin ser leídos u olvidados durante algún tiempo en el buzón de correo. Todo lo anterior puede ocurrir con los artículos de la Revista Cubana de Informática Médica, de la cual la SOCIM debe emitir semestralmente una nueva edición.

Como una traba adicional para la promoción, el actual diseño gráfico de la interfaz del portal no fue creado tomando en cuenta la opinión de los usuarios ni directivos de la SOCIM. Lo que trae consigo que estos no se sientan identificados y que por ello no se cumplan los objetivos del portal. Además el hecho de que está conformado por páginas estáticas, hace que la actualización de nuevas noticias, eventos, conferencias y cursos sea muy lenta, debido a que se conciben a través de un editor de páginas web.

Por las cuestiones expresadas anteriormente se vislumbran los aspectos principales de la situación problemática que se presenta el Portal Web de la Sociedad Cubana de Informática Médica. Las deficiencias existentes, conllevan a que los esfuerzos de esta investigación estén encaminados a solucionar el siguiente problema a resolver: ¿Cómo facilitar la gestión de la información del Portal Web de la Sociedad Cubana de Informática Médica?

El objeto de estudio se enmarca en el proceso de gestión de la información en los portales web para sociedades científicas; teniendo como campo de acción el proceso de gestión de la información en el Portal Web de la Sociedad Cubana de Informática Médica. Por lo queda definido como objetivo general: Desarrollar un Portal Web para la Sociedad Cubana de Informática Médica, que facilite la gestión de toda la información de interés para sus asociados.

Para dar cumplimiento al objetivo general se proponen las siguientes tareas de la investigación:

1. Realizar el estudio del arte de los sistemas existentes a nivel internacional y nacional, relacionados con los procesos que se realizan en la Sociedad Cubana de Informática Médica (SOCIM), que sirvan como punto de partida en la propuesta del sistema.
2. Asimilar las tecnologías y arquitectura definidas por el Centro de Informática Médica (CESIM) de la Universidad de las Ciencias Informáticas, para el desarrollo de la solución.
3. Documentar los artefactos que propone el Proceso Unificado de Desarrollo en los flujos de trabajo: Modelado del Negocio, Requerimientos, Análisis y Diseño e Implementación, que sirvan de base para el desarrollo del sistema.
4. Diseñar el sistema sobre la base de la metodología de desarrollo seleccionada.
5. Implementar las funcionalidades definidas para los módulos del Portal Web de la Sociedad Cubana de Informática Médica permitiendo el cumplimiento de las peticiones de los asociados.

Con el desarrollo del Portal Web de la Sociedad Cubana de Informática Médica se espera que:

- Permita al asociado inscribirse sin tener que ir a una oficina de la SOCIM y que la misma disminuya la pérdida de información personal de sus afiliados.
- Lograr que la información de los artículos de la Revista Cubana de Informática Médica esté centralizada y que los procesos de envío, revisión y publicación sean más viable.
- Poder conocer con exactitud, los usuarios que han realizado el pago o no, e impedir que los atrasados puedan autenticarse y tener privilegios en el Portal.
- Facilite la comunicación de todos los asociados a través de un chat y un foro donde pueden aclarar sus dudas o proponer temas de su interés.
- Ayude a comprender a personas de otros países la información del sitio mediante la traducción de la misma a otros idiomas.

Este documento está organizado en cuatro capítulos. En el primero de ellos, FUNDAMENTACIÓN TEÓRICA, se describen los conceptos fundamentales asociados al dominio del problema, se expone el estado del arte del tema tratado, tanto a nivel nacional como internacional, se hace un análisis crítico entre las soluciones ya existentes y la propuesta, y se explican y justifican las tendencias, tecnologías y herramientas en las que se apoya la solución al problema.

En el segundo capítulo, CARACTERÍSTICAS DEL SISTEMA, se ilustra el flujo actual de los procesos a través de la descripción de los Procesos del Negocio, identificando las causas que dan origen a la situación problemática, propiciando el análisis necesario para definir las funcionalidades del sistema a desarrollar. En el tercero, DISEÑO DEL SISTEMA, se modela el sistema y se le da forma para que soporte todos los requisitos y restricciones, consolidando una arquitectura que sirva de base para la implementación.

En el cuarto y último capítulo, IMPLEMENTACIÓN, se describen en términos de componentes, las clases y subsistemas encontrados durante el diseño. Se implementan todas las funcionalidades identificadas, logrando un sistema que satisface las principales necesidades del cliente.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

El desarrollo de una investigación generalmente trae consigo la obtención o reafirmación de nuevos conocimientos y diferentes puntos de vistas al observar los fenómenos o procesos. Una de las tareas primordiales que ayudan a formar estos otros puntos de vista y a comprobar el estado actual del área de conocimiento explorado lo constituye el estudio de estado de arte. Como resultado del mismo se pueden identificar fallas o vulnerabilidades, nuevas y mejores formas de hacer las cosas, y ayudar a conformar una idea más sólida de los que se quiere lograr.

Por tales motivos en este capítulo se exponen los resultados de estudio de los portales web a nivel nacional e internacional que gestionan información enmarcada dentro del campo de acción de la investigación. Además se analizan las tecnologías, metodologías y herramientas a utilizar en el desarrollo de la investigación.

### **1.1. Sistemas automatizados vinculados al campo de acción.**

A nivel nacional e internacional existen portales web que representan las sociedades científicas de informática médica de diferentes países y que constituyen una fuente de información innata para el desarrollo de esta investigación. Dentro de ellas se puede destacar el Portal de la Sociedad Española de Informática de la Salud, la Asociación Internacional de Informática Médica y el propio Portal SOCIM. Por constituir portales web que manejan información semejante, sólo se analiza el Portal SOCIM. A continuación se hace referencia a otras aplicaciones encontradas.

#### **1.1.1. Soluciones informáticas existentes en el ámbito internacional.**

##### **Portal de la comunidad médica IntraMed.**

IntraMed es el primer portal exclusivo para la comunidad médica, desarrollado para ofrecer la más variada información del ámbito medicinal y de la salud. Su filosofía se basa en la renovación constante, un concepto de evolución para brindar a la comunidad médica latinoamericana la plataforma tecnológica más adecuada para el desarrollo de las más eficaces herramientas de información, actualización, docencia, investigación, atención clínica y gestión de salud. (4)

Actividades del sitio:

- Publicación de comentarios.
- Revisión de artículos.
- Publicación de cursos, conferencias, noticias y artículos.

### **Portal congresos-médicos.com.**

Es un portal que publican las reseñas de miles de congresos, simposios, cursos y otros eventos de interés sanitario.

Servicios del sitio:

- Publicación de información de los usuarios.
- Revisión de la información proporcionada por los usuarios.
- Gestión de cursos.
- Selección de los cursos más activos.
- Gestión de próximos congresos.
- Selección de próximos congresos.

#### **1.1.2. Soluciones informáticas existentes en el ámbito nacional.**

### **Portal Web SOCIM.**

Es el Portal Web de la Sociedad Cubana de Informática Médica tiene como finalidad integrar, coordinar, apoyar, difundir y desarrollar las diferentes áreas teóricas y de aplicación de la informática en las Ciencias de la Salud en la República de Cuba, así como integrar a los profesionales del país en los distintos foros nacionales e internacionales en el campo de la informática en la salud y promover la participación de los grupos representados. (5)

Los servicios brindados por el portal de la SOCIM son:

- Muestra vínculos con otras sociedades de informática médica.

- Le permite al visitante del sitio web que observe varios eventos de informática médica y noticias de la actualidad mediante vínculos a importantes fuentes de la información en Cuba.
- Se publican conferencias para los asociados a la SOCIM.
- Existe un espacio para que los asociados conozcan sus derechos y sus deberes.
- Presenta un vínculo con los principales contactos de la SOCIM.
- Presenta un enlace con la Revista Cubana de Informática Médica y otros sitios de interés.

### **Revista Cubana de Informática Médica.**

La Revista Cubana de informática Médica (RCIM) difunde los resultados de trabajos relacionados con la informática médica y otras especialidades afines, preparadas por profesionales, docentes, técnicos así como estudiantes de maestrías, diplomados y pregrado.

Se aceptan colaboraciones nacionales y extranjeras en los idiomas español e inglés a condición de que observen las instrucciones estipuladas. Tales colaboraciones se publicarán en español. Son objeto de tratamiento priorizado los trabajos inéditos, de carácter teórico, metodológico o práctico cuya difusión contribuya a su aplicación práctica o resulte de utilidad sustancial para los lectores.

### **Políticas:**

Las recomendaciones a los autores emitidas para la redacción de sus manuscritos y posterior envío al Comité Editorial de la RCIM, se basan en la observancia de la normativa Uniform Requirements for Manuscripts Submitted to Biomedical Journals: Writing and Editing for Biomedical Publication (Requisitos uniformes) con el fin de mejorar la preparación de los materiales propuestos, en cuanto a calidad y claridad de la exposición se refiere. (6)

### **1.2. Técnicas, tecnologías, metodologías y software usados para la solución del problema.**

Las tecnologías a utilizar para la nueva construcción del Portal Web SOCIM son aquellas que han sido definidas por el Área Temática Sistema de Apoyo a la Salud (SAS), las cuales en su mayoría son libres. A continuación se muestran aspectos respecto a las mismas que justifican su elección.

### **Proceso Unificado Racional (RUP).**

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un software. Sin embargo, es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema o software en construcción está formado por componentes interconectados a través de interfaces.

Las características fundamentales que marcan el desarrollo y adaptación de la metodología en un proyecto según RUP:

- Centrado en la arquitectura, lo que posibilita que se defina la estructura fundamental del sistema y que la misma nunca se pierda.
- Guiado por los casos de uso, brindando la característica de que el desarrollo se vea marcado por una serie de construcciones organizadas por los casos de uso, posibilitando una mejor organización y avance en el desarrollo.
- Iterativo e incremental, posibilitando establecer las construcciones del sistema por partes y que a medida que sean terminadas dichas partes se vayan obteniendo incrementos en el desarrollo del software.
- RUP divide el proceso en cuatro fases, y a su vez divide estas en iteraciones a través de las cuales se van obteniendo productos que posibilitan incrementos en el software. Las fases definidas son: Fase de Inicio, Fase de Elaboración, Fase de Construcción y Fase de Transición. **(Ver Anexo 1)**
- Brinda una serie de flujos de trabajos que son realizados durante las fases anteriormente mencionadas y que garantizan el orden en que deben ser ejecutadas todas las tareas que establece esta metodología, además de lograr que la realización de la aplicación sea de alta calidad, que resuelva las necesidades del usuario dentro de un cronograma y que el producto sea obtenido con el menor costo posible. RUP utiliza el UML (Lenguaje Unificado de Modelado) para representar todos los diagramas en la concepción y diseño del Sistema de Planificación a desarrollar.

- Es muy útil para eliminar carga de trabajo al servidor, siendo idóneo para las validaciones de datos y acciones que son posibles hacer del lado del cliente, con el objetivo de agilizar el proceso que puede desencadenar el servidor como respuesta a una petición del cliente. (7)

### **Lenguaje de Modelado Unificado UML 2.1.**

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) permite visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como los procesos de negocio y funciones de sistema. Además este lenguaje gráfico permite realizar actividades concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

#### **Ventajas de UML 2.1:**

- Produce un aumento en la calidad del desarrollo.
- Reduce los costos del proyecto.
- Mejora en un 50% o más los tiempos totales de desarrollo.
- Permite especificar la estructura y el comportamiento del sistema y comunicarlo a todos los integrantes del proyecto.
- Brinda la posibilidad de obtener un "plano" del sistema.
- Permite dimensionar mejor los riesgos de un proyecto, tener un mejor rendimiento antes de construir el sistema.
- Facilita la documentación de las decisiones de la arquitectura del proyecto.
- Ofrece un mejor soporte a la planificación y control del proyecto.
- Ofrece mayor rigurosidad en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y viceversa. (8)

### **Enterprise Architect 7.1.**

Es considerada una de las herramientas más potentes y flexibles para la plataforma de Windows. Una herramienta de análisis de negocio y UML orientada a objetos para el desarrollo completo del ciclo de vida del software. Enterprise Architect provee el límite competitivo para el desarrollo de software,

administración de proyecto, administración de requerimientos y análisis de negocio. Entre sus características más importantes figuran:

- Última especificación UML 2.1.
- Importación/Exportación XMI 2.1.
- Nuevo motor de Reporte HTML.
- Transformaciones MDA.
- Perfiles y soporte de tecnologías.
- Pruebas, rastreo de recursos, mantenimiento.

Otras de sus características son:

- Modelado Basado en el Equipo.
  - Archivos compartibles o modelos basados del repositorio.
  - Control de versiones con cualquier herramienta.
  - Seguridad incorporada y administración de permisos.
- Modelado visual por excelencia.
  - Ingeniería reversa de código fuente en 10 lenguajes.
  - Importar esquema de base de datos.
  - Importar fuente XSD y WSDL.
  - Importar binarios .NET y Java.
  - Integración con Visual Studio y Eclipse. (9)

### **Aplicaciones Web.**

Las aplicaciones Web son sistemas informáticos, usados por los usuarios para acceder a un servidor Web, a través de Internet o Intranet, y obtener así la información que se encuentra almacenada en él. Las aplicaciones Web son muy utilizadas hoy en día debido a las ventajas que las mismas ofrecen. Algunas de estas se ven reflejadas a continuación:

- **Compatibilidad multiplataforma.** Tienen un camino mucho más sencillo para la compatibilidad con múltiples plataformas que las aplicaciones de escritorio. Varias tecnologías incluyendo Java, Flash, ASP y AJAX permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.

- **Actualización.** Están siempre actualizadas con la última versión, sin necesidad de llamar la atención del usuario o interferir con sus hábitos de trabajo, con la posibilidad de que va a iniciar nuevas descargas y procedimientos de instalación.
- **Inmediatez de acceso.** No necesitan ser descargadas, instaladas y configuradas.
- **Menos Bugs (errores).** Con aplicaciones basadas en Web, todos los usuarios utilizan la misma versión y los errores pueden ser corregidos tan pronto como son descubiertos.
- **Múltiples usuarios concurrentes.** Las aplicaciones web pueden ser utilizadas por múltiples usuarios al mismo tiempo. (10)

Para que la información de las aplicaciones web viaje de forma segura por la red, existen medios que permiten la encriptación u ocultamiento de la información de forma que no exista ningún tercero que sea capaz de obtenerla y utilizarla con fines lesivos que perjudiquen a los involucrados. Este medio se llama protocolos de comunicación.

### **Servidor Web Apache 2.2.**

Es considerado el servidor web por excelencia pues desde su surgimiento ha demostrado que es estable y que tiene mejor rendimiento que sus competidores. Además brinda algunas ventajas que son fundamentales para el desarrollo web como son:

- **Fiabilidad:** Más del 90% de los servidores con más alta disponibilidad funcionan bajo un servidor Apache.
- **Software libre:** El servidor Apache es totalmente gratuito y es distribuido bajo la licencia de Apache en la cual se permite realizarle cambios al código fuente.
- **Extensibilidad:** Se pueden añadir módulos para ampliar aún más las capacidades del servidor. (11)

### **PostgreSQL 8.3.**

El Gestor de Base de Datos PostgreSQL 8.3 es un sistema de base de datos muy potente de código abierto. Soporta gran parte del estándar SQL y, en algunos aspectos, está diseñado para que sea extensible por los usuarios. Se caracteriza por posibilitar transacciones ACID, claves foráneas, vistas, secuencias, subpeticiones, lanzadores, tipos y funciones definidas por el usuario, control de concurrencia

multiversión. También posee interfaces gráficas de usuario y enlazadores para algunos lenguajes de programación. (12)

#### **Principales características:**

- Bloqueos consultivos: permiten el control de objetos de bases de datos a nivel de aplicación usando el motor rápido de bloqueos de PostgreSQL.
- Sentencias preparadas: tiene nuevas interfaces administrativas y mejoras de rendimiento en sentencias preparadas.
- Criptografía: el módulo pgcrypto, soporta criptografía dentro de la base de datos, el cual fue actualizado con los mejores algoritmos de criptografía. (13)

#### **PgAdmin III.**

Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP o Unix Domain Sockets (en plataformas \*nix), y puede encriptarse mediante SSL para mayor seguridad. (14)

#### **Adobe Fireworks CS4.**

Adobe Fireworks CS4 brinda una de las formas más simples de crear, optimizar y exportar gráficos interactivos en un entorno único y centrado en el Web. Automatiza el proceso de producción de gráficos Web, crea múltiples gráficos a partir de una sola fuente de datos y gana tiempo para lograr diseños integrales y racionalizados. (15)

Con Fireworks CS4 se puede generar un diseño completo y exportarlo como página Web entero o en partes. Escribe código HTML y JavaScript automáticamente y facilita el proceso de actualización si el diseño cambia o es preciso hacerle modificaciones. Permite crear automáticamente gráficos y formatos JavaScript para botones de navegación, interfaces interactivas y menús emergentes sin tener que escribir el código fuente. También crea archivos emergentes en formato CSS. (16)

### Artisteer 2.

Artisteer permite diseñar temas para insertarlos en gestores de contenidos como Drupal, Joomla, Wordpress o Blogger, consiguiendo tener una Web completamente personalizada, en la que se puede configurar hasta el más mínimo aspecto. (17)

Las principales características de **Artisteer** son:

- ✓ Creación de plantillas para páginas Web sencillas.
- ✓ Creación de plantillas para Wordpress, Drupal o Joomla.
- ✓ Ayuda en hacer nuevas ideas para el diseño Web.
- ✓ Código HTML y CSS en conformidad con estándares Web.
- ✓ Soporta los formatos: ARTX, HTML, JPG, PNG, GIF.
- ✓ Interfaz simple e intuitiva.
- ✓ Fácil de usar.
- ✓ Soporte en diferentes idiomas. (18)

### HTML.

El **Lenguaje de Marcas de Hipertexto (HTML)** por sus siglas en inglés) define las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web. (19)

### Javascript 1.1.

Es un lenguaje de programación que proporciona dinamismo a las páginas web, facilitando la aparición y desaparición del texto, animación, y acciones que se activan al oprimir en botones y ventanas que aparecen con mensajes. Es un lenguaje interpretado, por lo que no es necesario compilar el código para ejecutarlo, es decir, es posible probarlos en cualquier navegador sin necesidad de procesos intermedios.

Está diseñado para controlar la apariencia y eventos de los objetos de un formulario, siendo soportado en múltiples navegadores, compartiendo elementos con otros lenguajes de alto nivel como son el C, Java o PHP, no solo en su sintaxis sino en su formato. (20)

## **CSS 2.**

Las Hojas de Estilo en Cascada (*Cascading Style Sheets*), es un mecanismo que describe cómo se va a mostrar un documento en la Web, o incluso cómo se va a imprimir. Este mecanismo ofrece a los desarrolladores el control sobre el estilo y el formato de sus documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Estos estilos permiten a los desarrolladores Web controlar el formato de múltiples páginas Web al mismo tiempo, posibilitando ante cualquier cambio en el estilo marcado para un elemento en la CSS, que sean afectadas todas las páginas vinculadas a dicha CSS en las que aparezca ese elemento. (21)

## **PHP 5.3.**

**PHP** (Hypertext Pre-Processor) es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Es un lenguaje muy potente y usado en el desarrollo Web.

Ventajas de PHP:

- ✓ La principal ventaja se basa en ser un lenguaje multiplataforma.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- ✓ Posee una muy buena documentación en su página oficial.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Desventajas de PHP:

- ✓ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- ✓ La orientación a objetos es aún muy deficiente para aplicaciones grandes. (22)

## **NetBeans 6.9.**

Es una aplicación de código abierto diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. NetBeans IDE dispone de soporte para

crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de packs. En NetBeans IDE se puede encontrar la solución más completa para programar en Java.

Entre las mejoras y novedades de Netbeans 6.9 se destaca:

- JavaFX Composer: una herramienta visual para construir aplicaciones JavaFX.
- Soporte de la plataforma OSGI. (23)
- Mejoras en el soporte de los frameworks JavaFX SDK 1.3, PHP Zend, Ruby on Rails 3.0 y Spring Framework 3.0. (24)

### **CMS Drupal 6.15.**

Drupal es el equivalente fonético en inglés a la palabra neerlandesa “druppel” que significa gota, fue originalmente escrito por Dries Buytaert, empezó como un pequeño BBS (Sistema de Tablón de Anuncios, por sus siglas en inglés, Bulletin Board System). Drupal ha llegado a ser mucho más que sólo un portal de noticias gracias a su arquitectura flexible.

Constituye una plataforma dinámica para la construcción de sitios web que permite a un individuo o una comunidad de usuarios publicar, manejar y organizar una variedad de contenido. Integra muchas características populares de los Sistemas de Gestión de Contenido, Weblogs, herramientas de colaboración y comunidad de discusión, todo en un solo paquete fácil de utilizar.

Muy conveniente para este trabajo que se desea implementar, dado que posee varios módulos que serían de gran ayuda. Por poner un ejemplo, el módulo Workflow (Flujo de trabajo) es implementado para el traspaso de información antes de ser publicada a través de un flujo de trabajo manejado por diferentes roles asignados a varias personas. Con este módulo se asegura de que la información sólo pueda ser vista por un rol específico en un momento determinado.

Drupal es de código abierto con una comunidad a su alrededor muy extensa, de variados idiomas y conocimientos, esto lo ha hecho crecer y posicionarse encima de los demás CMS; es libre para descargarlo de Internet y usarlo. (25)

### 1.3 Arquitectura del Sistema.

La arquitectura, según Roger S. Pressman, es la estructura de las estructuras del sistema, comprende los componentes de software, las propiedades de esos componentes visiblemente externos y las relaciones entre ellos. Esta no es más que la organización de los componentes del sistema de forma que quede registrado la manera en la que colaboran y se relacionan entre ellos. Es una vía en la cual el sistema queda modelado desde distintas perspectivas con el objetivo de lograr y establecer como deberá ser construido el futuro sistema. (26)

#### Cliente-Servidor.

Este tipo de arquitectura es un modelo que las aplicaciones clientes utilizan para comunicarse con el servidor. Donde un servidor es una aplicación que ofrece servicios a usuarios en Internet, el servidor recibe una solicitud de un servicio por parte de un cliente y devuelve los resultados en forma de una respuesta a la solicitud realizada. (**Ver Anexo 2**)

Los usuarios son los que inician la invocación del servicio en la parte cliente de la aplicación en la cual se construye la solicitud del servicio y es enviada al servidor utilizando el protocolo TCP como medio de transporte. (27)

Con la implantación de este modelo cada usuario, llamado cliente, tiene la posibilidad de trabajar desde cualquier lugar solamente haciendo una solicitud al servidor y manipulando la información que este le devuelva en forma de respuesta, en dependencia de las necesidades del cliente.

Bajo esta forma de comunicación, la información se puede mantener de forma centralizada, evitando así la pérdida de la misma por problemas de hardware u otros motivos relacionados con situaciones técnicas que puedan presentarse con los clientes. Este tipo de modelo es aplicable bajo cualquier circunstancia, el mismo puede implantarse tanto en Internet como en una red local (Intranet); además el procedimiento no es muy costoso ya que solo se requiere una PC y un navegador web como cliente de la aplicación. Igualmente cualquier tipo de mantenimiento en la aplicación sería transparente para el usuario, lo que posibilitaría reducir aún más, los costos en su implantación.

#### **Características que presenta la arquitectura Cliente-Servidor:**

- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.

- Un servidor brinda servicios a múltiples clientes en forma concurrente.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- Las funciones cliente-servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.
- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo. (28)

**Ventajas de la arquitectura Cliente-Servidor:**

- Permite un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión. Por ejemplo, la compartición de servidores y dispositivos periféricos (como impresoras) entre máquinas clientes.
- Se pueden utilizar componentes, tanto de hardware como de software de distintos fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- Proporcionan un mejor acceso a los datos. La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información.
- La arquitectura Cliente Servidor elimina la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Finalmente la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios.
- En una arquitectura como esta, los clientes y los servidores son independientes los unos de los otros, con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema. (29)

### **Modelo Vista Controlador (MVC).**

El sistema está implementado usando el CMS Drupal, por lo que la arquitectura y los patrones usados se heredan del mismo. La arquitectura del CMS Drupal utiliza el patrón MVC. Este patrón de arquitectura de software separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, (**Ver Anexo 3**) la vista es la página HTML y el código que provee de datos dinámicos a la página, las funcionalidades de la capa de presentación consisten en intercambiar información con los usuarios; el modelo es la capa de acceso a datos, esta está implementada con la utilización de una clase del negocio que se encarga de establecer la conexión con el servidor, garantizando el acceso a ficheros del mismo; y el controlador es el responsable de recibir los eventos de entrada desde la vista, encargándose de esta forma de recibir y responder cada petición de los usuarios.

La principal ventaja de esta separación reside en la facilidad para realizar cambios en la aplicación puesto que:

- ✓ Cuando se realiza un cambio de base de datos, programación o interfaz de usuario, solo se toca uno de los componentes.
- ✓ Se puede modificar uno de los componentes sin conocer cómo funcionan los otros.
- ✓ Incrementa la reutilización y flexibilidad.

Se decide la utilización del MVC por la facilidad que brinda para la creación de sistemas web, además de la organización que provee en el código fuente, por lo que se consigue un mantenimiento más sencillo de las aplicaciones. (30)

En este capítulo se analizaron varios sistemas que se han desarrollado tanto en Cuba como en el resto del mundo, y que fueron concebidos para la gestión de información de informática médica. Sin embargo, ninguno de ellos se ajusta a las necesidades de la SOCIM, debido a las peculiaridades que esta presenta en cuanto a la forma de inscripción y al pago de las cuotas de afiliación, principalmente. Mediante el análisis de los portales de las sociedades de informática médica de otros países se pudo comprobar que todas gestionen el conocimiento de forma semejante y que constan de publicaciones especializadas, que en el caso particular cubano es la Revista de Informática Médica.

Además se realizó un estudio acerca de las tecnologías y herramientas que han sido definidas por el Departamento Sistemas de Apoyo a la Salud (SAS), arribando a la conclusión de que las mismas son idóneas a utilizar, debido a que constituyen tecnologías de punta que recogen las mejores prácticas para el desarrollo de software. La mayoría de estas tecnologías son libres, lo cual constituye una salida para los países en busca de desarrollo y pequeñas empresas que no tendrían que someterse a las limitaciones impuestas por el software propietario.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

En el presente capítulo se describen las características del sistema a construir; desarrollando un modelo de dominio donde se exponen los principales conceptos manejados en el marco de la organización. Se realiza una especificación de las condiciones, capacidades y cualidades que debe tener el sistema, enumerándolas en los requisitos funcionales y no funcionales, así como los casos de uso del sistema, permitiendo un mejor entendimiento del mismo.

### 2.1 Modelo de Dominio.

El modelo de dominio es la representación de los conceptos más importantes y significativos en el desarrollo de un sistema. Representa clases conceptuales, conceptos del mundo real y no de los componentes del software. Su principal objetivo es ayudar a comprender los conceptos que utilizan los usuarios, las definiciones con las que trabajan y con las que deberá trabajar la aplicación. (31)

Para la realización del Portal Web para la Sociedad Cubana de Informática Médica (SOCIM) se decide realizar un Modelo de Dominio; esto se debe a que los límites del negocio no están definidos con exactitud y este constituye una buena opción para que los miembros del equipo conozcan el contexto organizacional donde se desarrollan los procesos. Los conceptos y definiciones que son tratados se obtienen a partir de las necesidades propias de la SOCIM, pues representan objetos del mundo real que serán abordados por la aplicación en elaboración.

#### 2.1.1. Conceptos fundamentales.

**Información de informática médica:** Representa toda la información de informática médica contenida en el portal, como por ejemplo: noticias de última hora, eventos, conferencias, cursos y artículos de la Revista Cubana de Informática Médica que presenta el Portal Web.

**Asociado:** Este rol es el que se asigna automáticamente a todos los usuarios que se hayan autenticado en el portal, sin más permisos que crear un artículo y modificarlo.

**Noticia:** Es el contenido de una comunicación antes desconocida, un hecho divulgado o la divulgación de una doctrina. En otras palabras, la noticia es un conocimiento o una noción.

**Evento:** Es un suceso de informática médica de importancia que se encuentra programado.

**Curso:** Es el estudio sobre una materia, estructurada como una unidad.

**Conferencia:** Es una disertación en público sobre algún tema concreto.

**Artículo:** Escrito de cierta extensión de un periódico, revista o publicación periódica.

**Planilla de Inscripción:** Dicha planilla presenta todos los datos del proceso de inscripción para los asociados en la Sociedad Cubana de Informática Médica.

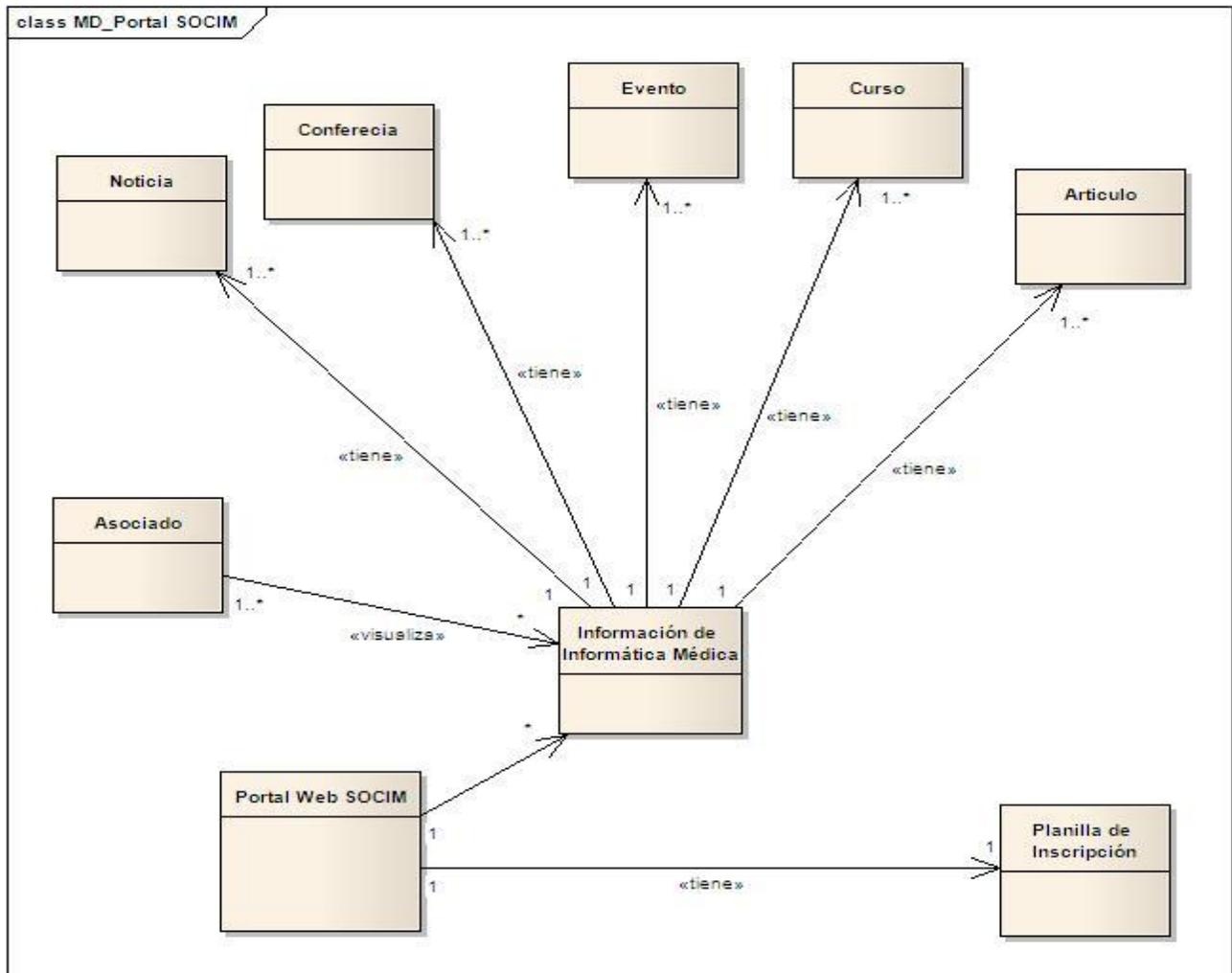


Fig. 1. Diagrama del Modelo del Dominio del Portal Web SOCIM.

**2.2 Requerimientos de software.**

Los requerimientos representan, propiedades, condiciones o capacidades que los usuarios necesitan, para resolver un problema o alcanzar una meta, siendo características que el sistema debe tener para cumplir las necesidades que lo impulsan. Definen que es lo que debe y no debe hacer el sistema, permitiendo destacar las funcionalidades y restricciones que se requieren. (32)

**2.2.1 Requerimientos funcionales.**

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, no alteran la funcionalidad del producto, lo cual quiere decir que se mantienen invariables sin importar con que propiedades o cualidades se relacionen. (33) A continuación se muestra el listado de requerimientos obtenido luego de ser analizados los principales conceptos del Modelo de Dominio.

Listado de Requerimientos funcionales	
RF1. Autenticar.	RF14. Mostrar rol.
RF2. Crear cuenta.	RF15. Crear evento.
RF3. Modificar cuenta.	RF16. Eliminar evento.
RF4. Eliminar cuenta.	RF17. Modificar evento.
RF5. Mostrar cuenta.	RF18. Buscar evento.
RF6. Mostrar curso.	RF19. Mostrar evento.
RF7. Eliminar curso.	RF20. Crear noticia.
RF8. Modificar curso.	RF21. Eliminar noticia.
RF9. Crear curso.	RF22. Modificar noticia.
RF10. Buscar curso.	RF23. Mostrar noticia.
RF11. Crear rol.	RF24. Buscar noticia.
RF12. Eliminar rol.	RF25. Crear artículo.
RF13. Modificar rol.	

<p>RF26. Eliminar artículo.</p> <p>RF27. Modificar artículo.</p> <p>RF28. Mostrar artículo.</p> <p>RF29. Buscar artículo.</p> <p>RF30. Crear usuario.</p> <p>RF31. Eliminar usuario.</p> <p>RF32. Modificar usuario.</p> <p>RF33. Mostrar usuario.</p> <p>RF34. Recibir Notificación de Contenidos Creados.</p> <p>RF35. Recibir Notificación de Contenidos Aprobados.</p> <p>RF36. Recibir Notificación de Contenidos Rechazados.</p> <p>RF37. Recibir Notificación de Contenidos Eliminados.</p>	<p>RF38. Crear conferencia.</p> <p>RF39. Eliminar Conferencia.</p> <p>RF40. Modificar Conferencia.</p> <p>RF41. Buscar Conferencia.</p> <p>RF42. Mostrar conferencia.</p> <p>RF43. Mostrar Pagos Realizados.</p> <p>RF44. Crear mensaje.</p> <p>RF45. Modificar mensaje.</p> <p>RF46. Eliminar mensaje.</p> <p>RF47. Mostrar mensaje.</p> <p>RF48. Buscar asociados a pagar.</p> <p>RF49. Realizar pago atrasado de asociado.</p> <p>RF50. Activar cuenta de asociado.</p> <p>RF51. Bloquear cuenta de asociado.</p> <p>RF52. Realizar pago de asociado.</p>
--	--

Tabla 1. Requisitos funcionales.

### 2.2.2 Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. (34)

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. (35)

#### **Usabilidad.**

**RNF 1.** El sistema emitirá mensajes informativos, de advertencia o de error, que servirán para informar al usuario acerca de la realización satisfactoria o no de determinadas acciones, y en otros casos para indicar que se no se ha introducido algún dato obligatorio o que se introdujo de forma incorrecta.

**RNF 2.** Todos los mensajes de error o de advertencia del sistema deben mostrarse de forma similar permitiendo una mayor organización en la visualización de la información.

#### **Seguridad.**

**RNF 3.** La información en el portal web estará protegida con un mecanismo de autenticación (usuario, contraseña) y autorización (asignación de roles), permitiendo este último la asignación de permisos a los módulos, bloques, páginas, interfaces y vistas (views).

#### **Eficiencia.**

**RNF 4.** Para agilizar la navegación, el portal debe guardar en cache páginas de contenidos.

#### **Hardware.**

**RNF 5.** Los servidores de base de datos y de aplicación deberán contar con:

- Tipo de procesador: Intel Pentium IV o superior.
- Velocidad del procesador: 3.00 GHz
- Memoria RAM: 4 GB o superior.
- Disco duro: 60 GB o más.
- Se requiere tarjeta de red.

**RNF 6.** Las estaciones de trabajo clientes deberán contar con:

- Tipo de procesador: Intel Pentium IV o superior.
- Velocidad del procesador: 512 MHz o superior.
- Memoria RAM: 512 MB o superior.
- Disco duro: 4 GB o más.
- Se requiere tarjeta de red.

### **Software.**

**RNF 7.** El servidor de aplicación debe contar con el servidor de aplicaciones web Apache 2.2.

**RNF 8.** El servidor de base de datos debe contar con el gestor de base de datos PostgreSQL 8.3.

**RNF 9.** En las computadoras de los clientes se requiere de un navegador web Mozilla Firefox versión 4.0 o superior.

### **2.3 Descripción del sistema propuesto.**

El sistema propuesto será un portal web que satisfaga las necesidades de información acerca de los servicios que brinda la SOCIM, así como las funcionalidades que se le agregarán, entre las que se encuentran el registro de la información del asociado, la revisión de la información que ellos publican y la gestión de eventos, conferencias, cursos y artículos de la Revista Cubana de Informática Médica. Su creación debe basarse en los requisitos planteados anteriormente para cumplir exitosamente los objetivos del mismo.

Para la realización del portal se usará el Sistema de Gestión de Contenidos (CMS) Drupal, por las características planteadas en el capítulo anterior. La aplicación contará con cinco roles: invitado, asociado, responsable de la Revista Cubana de Informática Médica, revisor y administrador. El rol de invitado podrá navegar por el contenido referente a las publicaciones, ya sean noticias o artículos de interés, foros,

encuestas, eventos y cursos. El asociado, tendrá las mismas funcionalidades que un invitado, pero podrá acceder al Portal Web por su cuenta personal, y además podrá crear artículos, comentarlos, dar su opinión en los foros, entre otras.

Por su parte el revisor, es el encargado de realizar la revisión o actualización de los artículos, noticias, foros, encuestas, entre otros contenidos que serán publicados. El responsable de la Revista Cubana de Informática Médica será el rol asignado directamente a una persona que es parte del colectivo de la Revista Cubana de Informática Médica (RCIM); este responsable deberá determinar si un artículo aprobado por el revisor se podrá publicar en la RCIM. Mientras que el rol de administrador se encarga de llevar a cabo las funcionalidades relacionadas con la actualización y mantenimiento del portal de forma general.

**2.3.1 Actores del sistema.**

Actor	Descripción
Revisor	Es el actor que se encarga de realizar el proceso de revisión de cada una de las noticias y artículos publicados en el Portal Web.
Administrador	Es el actor encargado de realizar las funciones administrativas en el portal como corregir errores, crear un usuario nuevo, darle un rol al usuario y activar los permisos que llevará el mismo, etc.
Invitado	Este actor se emplea para los usuarios que no tienen cuenta o que no se han autenticado, solo puede visualizar la información, no tiene permisos para modificarla y puede crear su propia cuenta.

Asociado	Este actor es el que se asigna automáticamente a todos los usuarios que ingresan, además pueden ver su cuenta una vez creada y crear artículos que pueden o no ser publicados.
Responsable de la RCIM	Este actor es el que se encarga seleccionar de los artículos aprobados por el revisor del Portal Web de la SOCIM y luego publicarlos en la Revista Cubana de Informática Médica.

Tabla 2. Descripción de actores del sistema.

2.3.2 Vista Global de los Actores del Sistema.

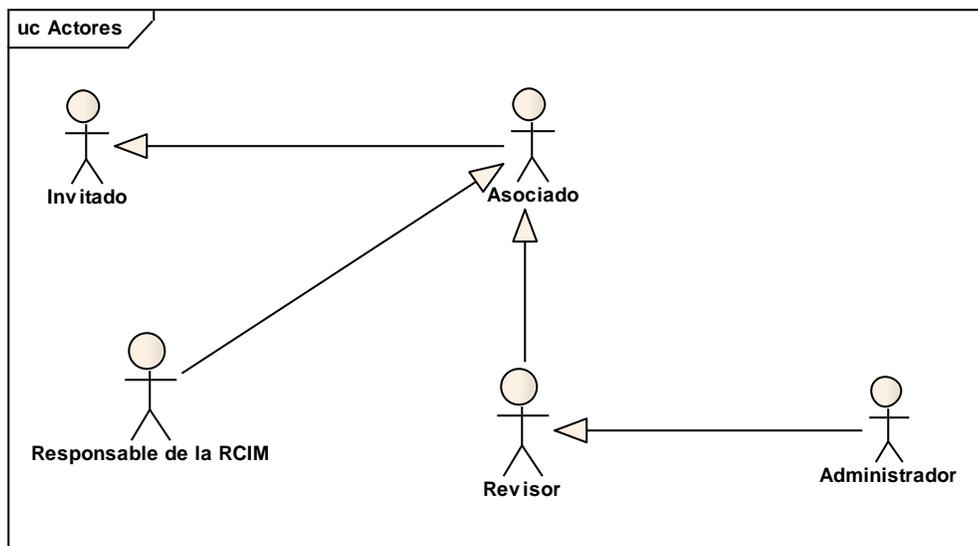


Fig. 2. Diagrama de Actores del Sistema.

2.3.3 Definición de los Casos de Uso.

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos identificados. Este proporciona la entrada fundamental para el análisis y el diseño en

el desarrollo de un sistema de software. Es un modelo del sistema que contiene actores, casos de uso y las relaciones que existen entre ellos. (36)

### **Diagrama de Casos de Uso del Sistema.**

A continuación se muestra el diagrama de casos de uso del sistema, donde se muestran las relaciones entre actores y casos de uso.

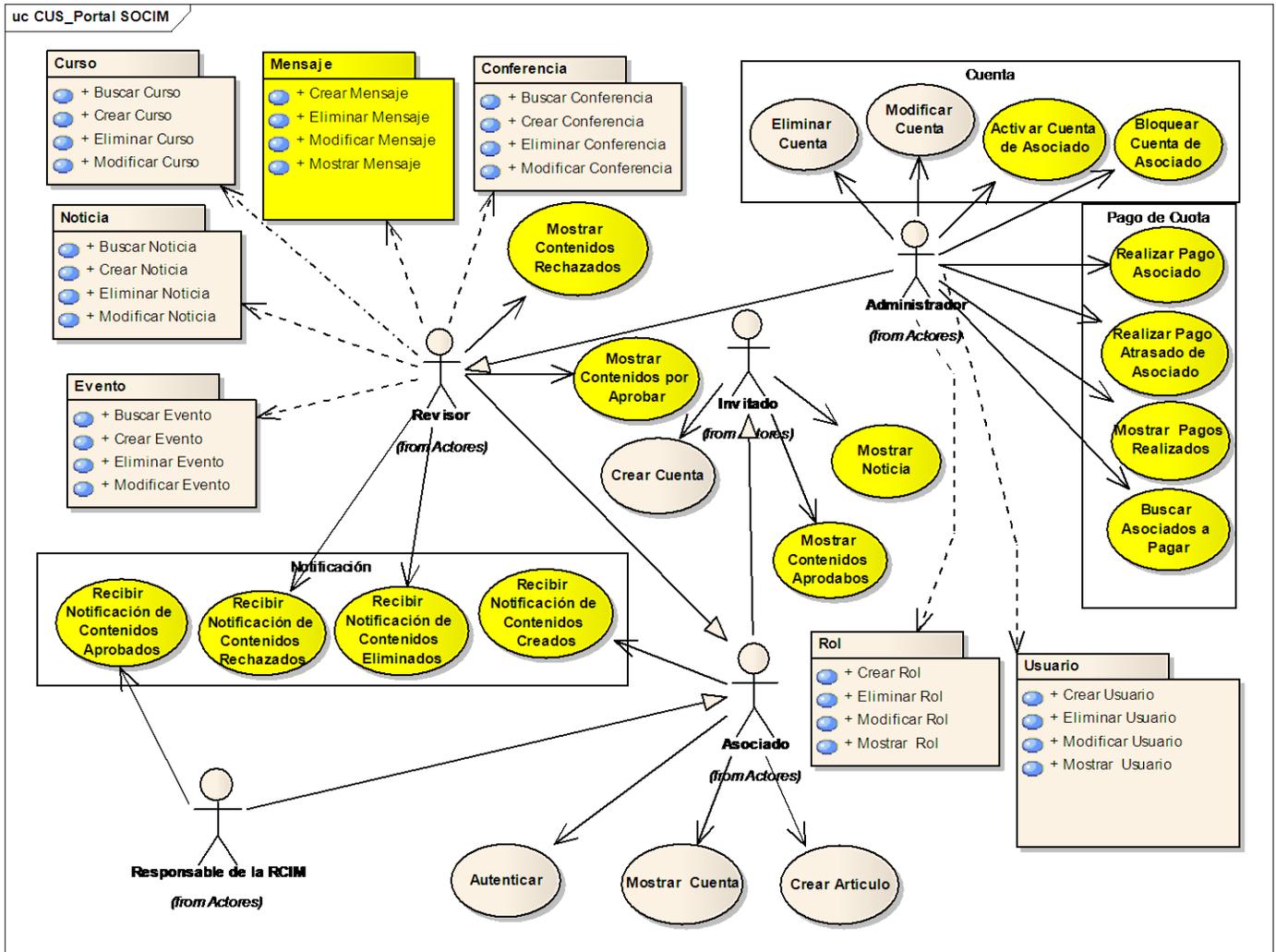


Fig. 3. Diagrama de Casos de Uso del Sistema.

**Descripción textual de Casos de Uso del Sistema.**

A continuación se muestra una breve descripción de algunos de los casos de usos:

Caso de Uso:	CU_Eliminar Mensaje.
Actores:	Revisor.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Eliminar Mensaje, el

	sistema brinda la posibilidad de eliminar los mensajes existentes, el caso de uso termina.
Precondiciones:	Debe existir el mensaje a eliminar.
Referencias:	RF46.
Prioridad:	Crítico.
Pos condiciones:	El mensaje ha sido eliminado.

Tabla 3. Descripción del Caso de Uso CU\_Eliminar Mensaje.

Caso de Uso:	CU_Mostrar Mensaje.
Actores:	Revisor
Resumen:	El caso de uso inicia cuando el revisor selecciona un mensaje y accede a la opción de Ver datos del mensaje, el sistema muestra los datos del mensaje, el caso de uso termina.
Precondiciones:	Debe de existir al menos un mensaje.
Referencias:	RF47.
Prioridad:	Crítico.
Pos condiciones:	Muestra los datos de la notificación de un artículo de informática médica.

Tabla 4. Descripción del Caso de Uso CU\_Mostrar Mensaje.

Caso de Uso:	CU_Crear Mensaje.
Actores:	Revisor.

<b>Resumen:</b>	El caso de uso inicia cuando el revisor accede a la opción Crear Mensaje, el sistema brinda la posibilidad de introducir y seleccionar los datos para crear el mensaje, el revisor introduce los datos del mensaje, el sistema crea el mensaje, el caso de uso termina.
<b>Precondiciones:</b>	Debe de existir al menos un mensaje.
<b>Referencias:</b>	RF44.
<b>Prioridad:</b>	Crítico.
<b>Pos condiciones:</b>	Muestra los datos de la notificación de un artículo de informática médica.

Tabla 5. Descripción del Caso de Uso CU\_Crear Mensaje.

<b>Caso de Uso:</b>	CU_Modificar Mensaje.
<b>Actores:</b>	Revisor.
<b>Resumen:</b>	El caso de uso inicia cuando el revisor selecciona un mensaje y accede a la opción Editar, el sistema muestra los datos del mensaje y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el revisor modifica los datos que necesita, el sistema actualiza los datos del mensaje, el caso de uso termina.
<b>Precondiciones:</b>	No puede existir el mensaje a crear.
<b>Referencias:</b>	RF45.
<b>Prioridad:</b>	Crítico.
<b>Pos condiciones:</b>	El mensaje ha sido modificado.

Tabla 6. Descripción del Caso de Uso CU\_Modificar Mensaje.

En este capítulo se obtuvo una representación del sistema que se propone desarrollar. Haciendo un levantamiento de requisitos funcionales y no funcionales, a partir del análisis detallado del modelo del dominio. Además se identificaron y describieron los actores que interactuarán con las funcionalidades previstas para cada uno de los casos de usos; quedando sentadas las bases que darán lugar al diseño.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO.

Con el desarrollo de este capítulo se profundiza en el análisis y diseño del sistema, se tienen como objetivos fundamentales: transformar los requerimientos definidos a una propuesta de diseño que será la guía a seguir para la implementación del sistema, evolucionar hacia una arquitectura del software robusta y flexible ante la aplicación de nuevos cambios, y realizar la propuesta de interfaz de usuario del software adaptando al diseño para que coincida con el ambiente de implementación.

Durante el diseño, se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y no funcionales del sistema en cuestión. Es la parte del proceso de desarrollo de software cuyo propósito primario es decidir cómo la implementación del sistema se llevará a cabo. Para lograr lo antes expuesto está dirigido este capítulo, realizando para ello artefactos como los Diagramas de Interacción y de Clases del Diseño, y la Descripción de la Arquitectura del Diseño.

### 3.1 Análisis.

Durante el análisis, se analizan los requisitos que fueron descritos en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar todo el sistema, incluyendo su arquitectura.

#### 3.1.1 Modelo de Análisis.

El modelo de análisis, que en realidad es una serie de modelos, es la primera representación técnica de un sistema. Utiliza una combinación de formatos en textos y diagramas para representar los requisitos de los datos, las funciones y el comportamiento de una manera que es relativamente fácil de entender y, aún más importante, conduce a una revisión para lograr la corrección, la integridad y la consistencia.

Aunque en el modelo del análisis hay un refinamiento de los requisitos, no se tiene en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, los componentes prefabricados o reusables de otras aplicaciones, entre otras características que afectan al sistema, ya que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución. (37)

### Clases de Análisis.

A continuación se hace una descripción de las clases que se utilizarán en la realización de dichos diagramas y que permitirán obtener una mejor visión del sistema. Los prototipos para identificar las diferentes clases que participan en el análisis son los siguientes:

- **CI\_<Nombre de la clase>**: estas son las Clases Interfaz, las cuales modelan la interacción entre los actores y el sistema, ventanas, formularios, comunicación con otros sistemas o dispositivos.
- **CC\_<Nombre de la clase>**: estas son las Clases Controladoras, las cuales coordinan el trabajo de las clases encapsulan el comportamiento de cada caso de uso y las funciones más complejas.
- **CE\_<Nombre de la clase>**: estas son las Clases Entidad, las cuales modelan toda la información del sistema que posee una vida larga y que puede ser persistente además de modelar el comportamiento asociado a una información.

### Diagrama de clases del análisis.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo. (38)

A continuación se realiza muestran los diagramas de clases del análisis de algunos casos de uso del sistema. Los diagramas restantes se encuentran en el Expediente de Proyecto.

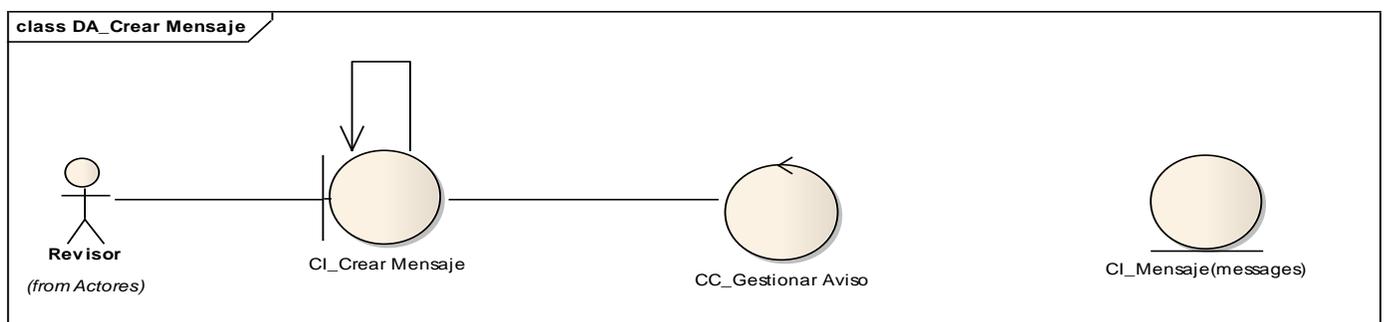


Fig. 4. Diagrama de Clases del Análisis CU\_Crear Mensaje.

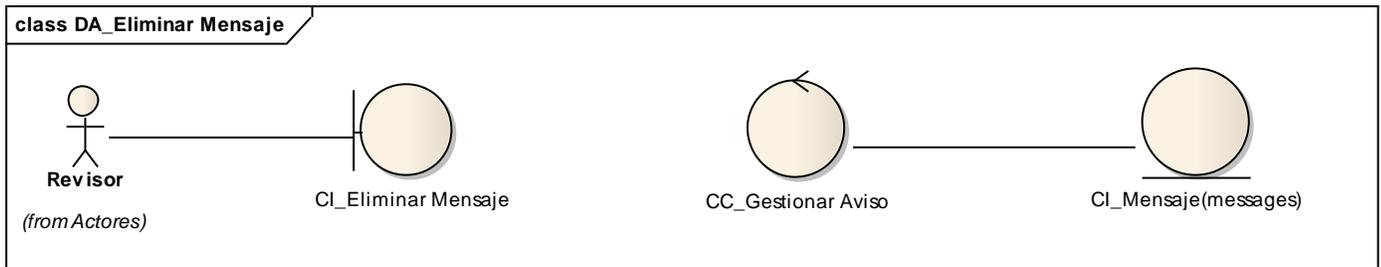


Fig. 5. Diagrama de Clases del Análisis CU\_Eliminar Mensaje.

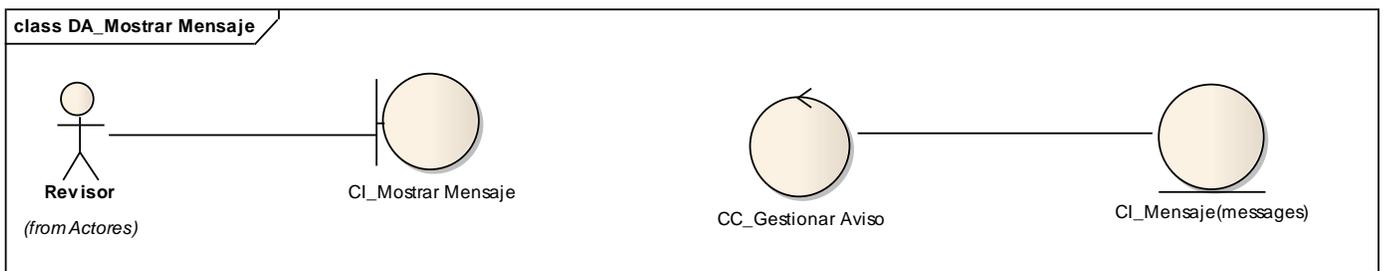


Fig. 6. Diagrama de Clases del Análisis CU\_Mostrar Mensaje.

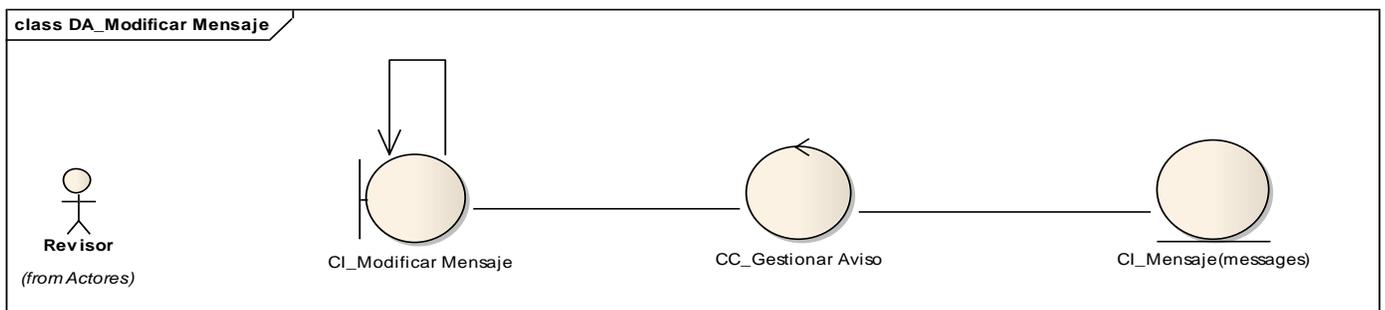


Fig. 7. Diagrama de Clases del Análisis CU\_Modificar Mensaje.

### 3.2 Diseño.

A continuación se desarrolla el flujo de trabajo de diseño, a través del cual se modela el sistema que se pretende construir para que este sea capaz de soportar todos los requisitos. El diseño es el centro de atención final de la fase de elaboración y el comienzo de las iteraciones de construcción, no es más que el refinamiento del análisis, el cual contribuye a una arquitectura estable y sólida, además de crear un plano del modelo de implementación. Más tarde durante la fase de construcción, cuando la arquitectura es estable y los requisitos están bien entendidos, el centro de atención se desplaza a la implementación.

### 3.2.1 Modelo de Diseño.

Es el principal artefacto utilizado como entrada para el correcto desarrollo de la fase de implementación. Como parte del mismo se construyen los diagramas de interacción y diagramas de clases del diseño.

#### **Diagrama de clases de diseño web.**

El CMS Drupal depende de un grupo de clases o módulos que permiten a los usuarios finales interactuar con la aplicación, como plataforma flexible permite configurar sus funcionalidades, crear nuevos tipos de contenido y la forma de mostrar estos. Drupal posee un tipo de contenido genérico llamado node, que puede ser extendido por los desarrolladores, tiene como propiedades básicas de publicación el título, autor, fecha de creación, estatus del contenido, y el desarrollo del contenido, además Drupal proporciona los mecanismos para la creación, edición y publicación de este.

Posee un sistema de seguridad basado en la creación de usuarios y roles, en dependencia de los permisos que se necesiten para administrar el sitio y acceder a los módulos, el mismo CMS se encarga de todo, así como del control de accesos a los diferentes módulos según los permisos definidos por el administrador. Dada su modularidad e independencia estructural se dividen en módulos por separados de acuerdo a sus funcionalidades. En lo adelante se refieren a paquetes cuando se habla de un modulo del Sistema de Gestión de Contenido. (39)

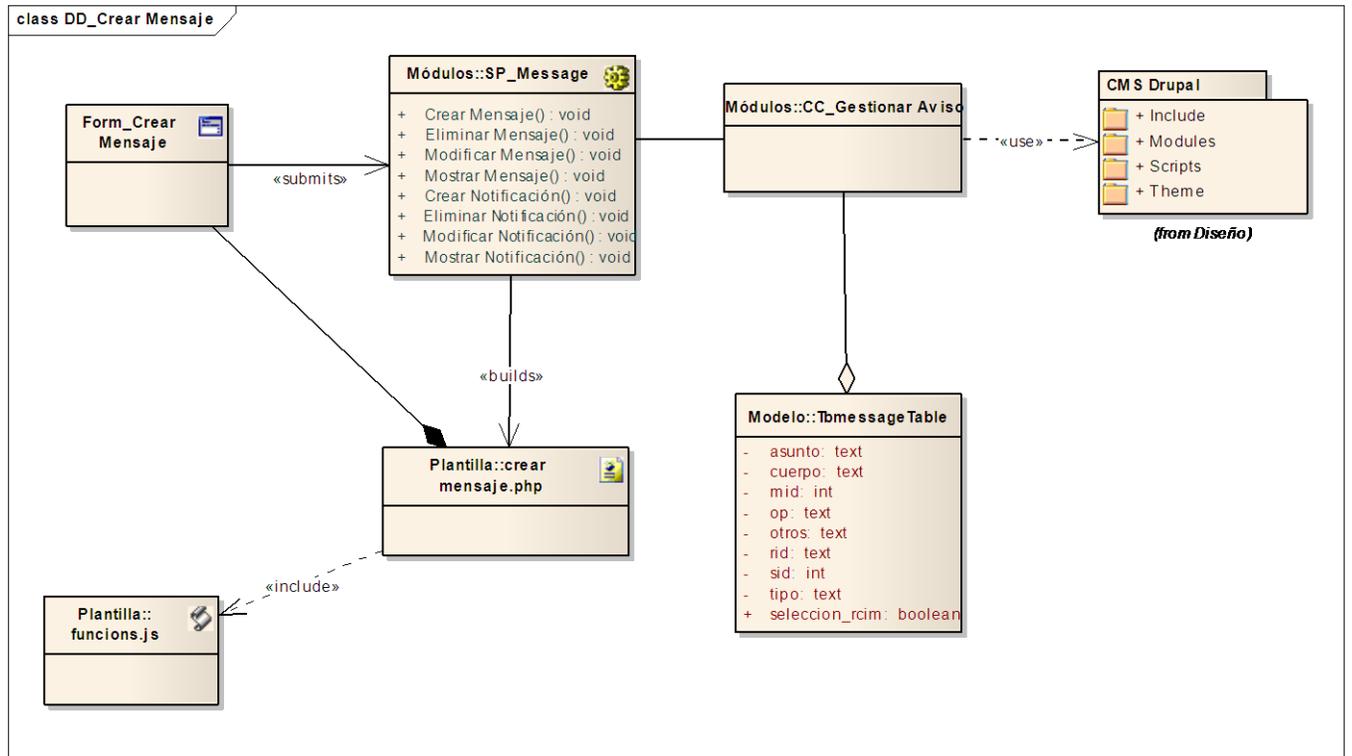


Fig. 8. Diagrama de Clases del Diseño CU\_Crear Mensaje.

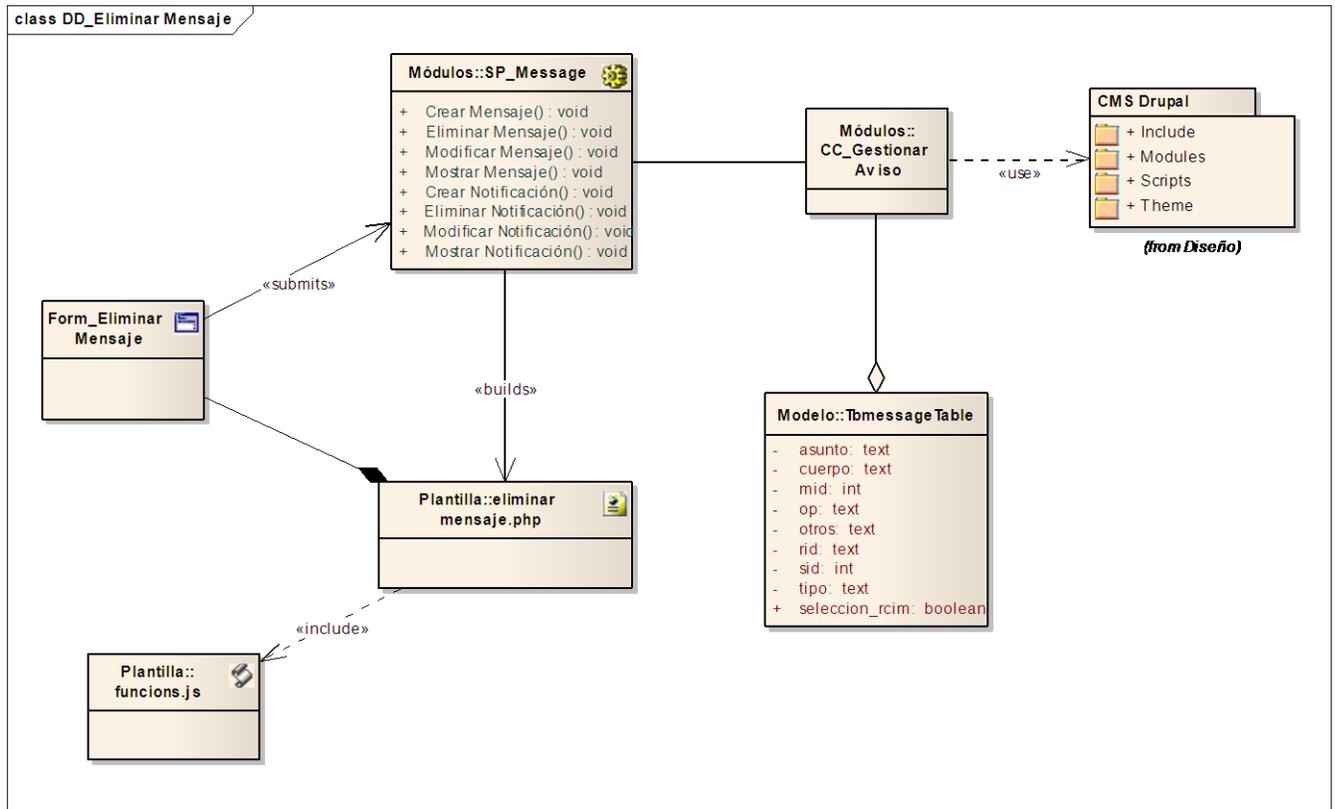


Fig.9. Diagrama de Clases del Diseño CU\_Eliminar Mensaje

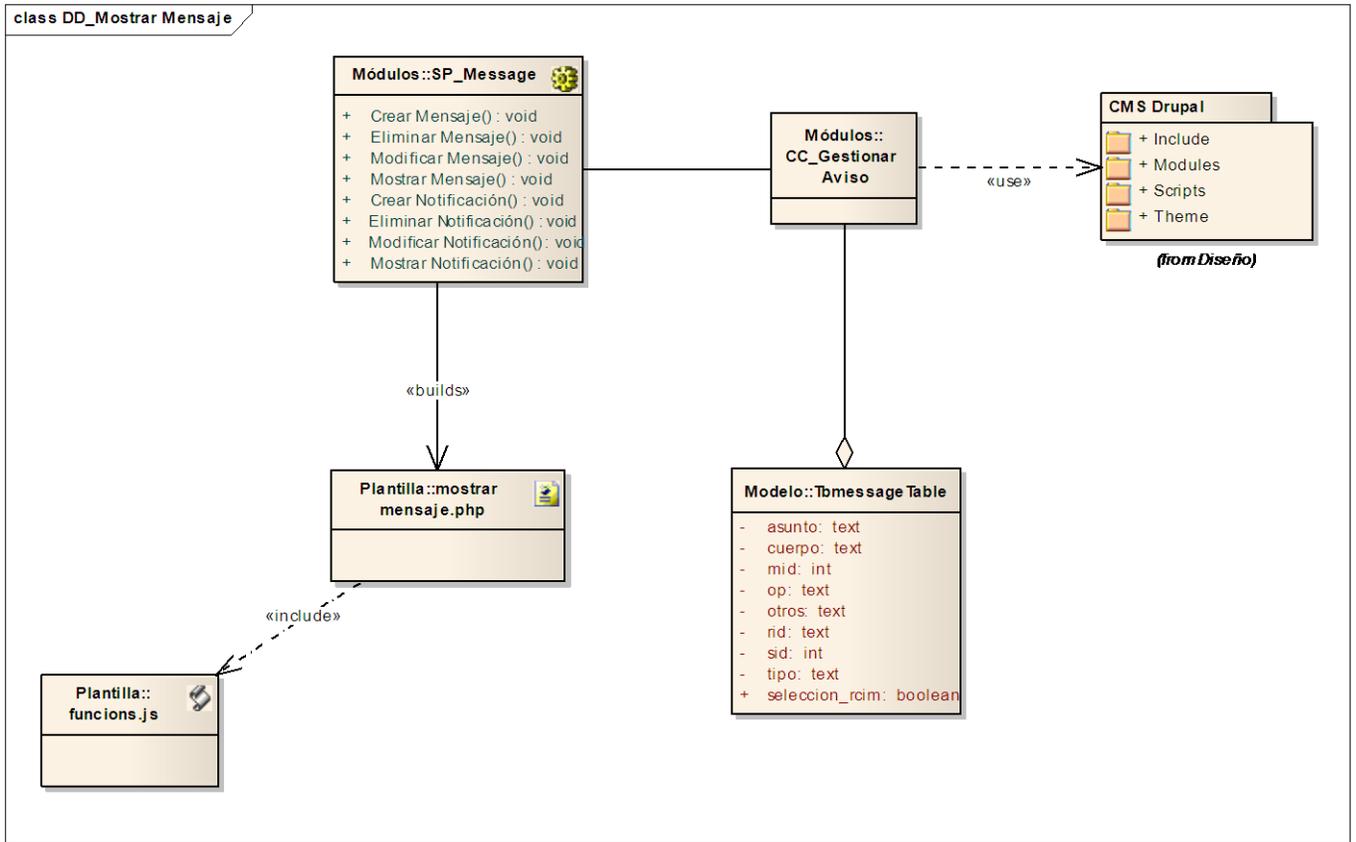


Fig.10. Diagrama de Clases del Diseño CU\_Mostrar Mensaje.

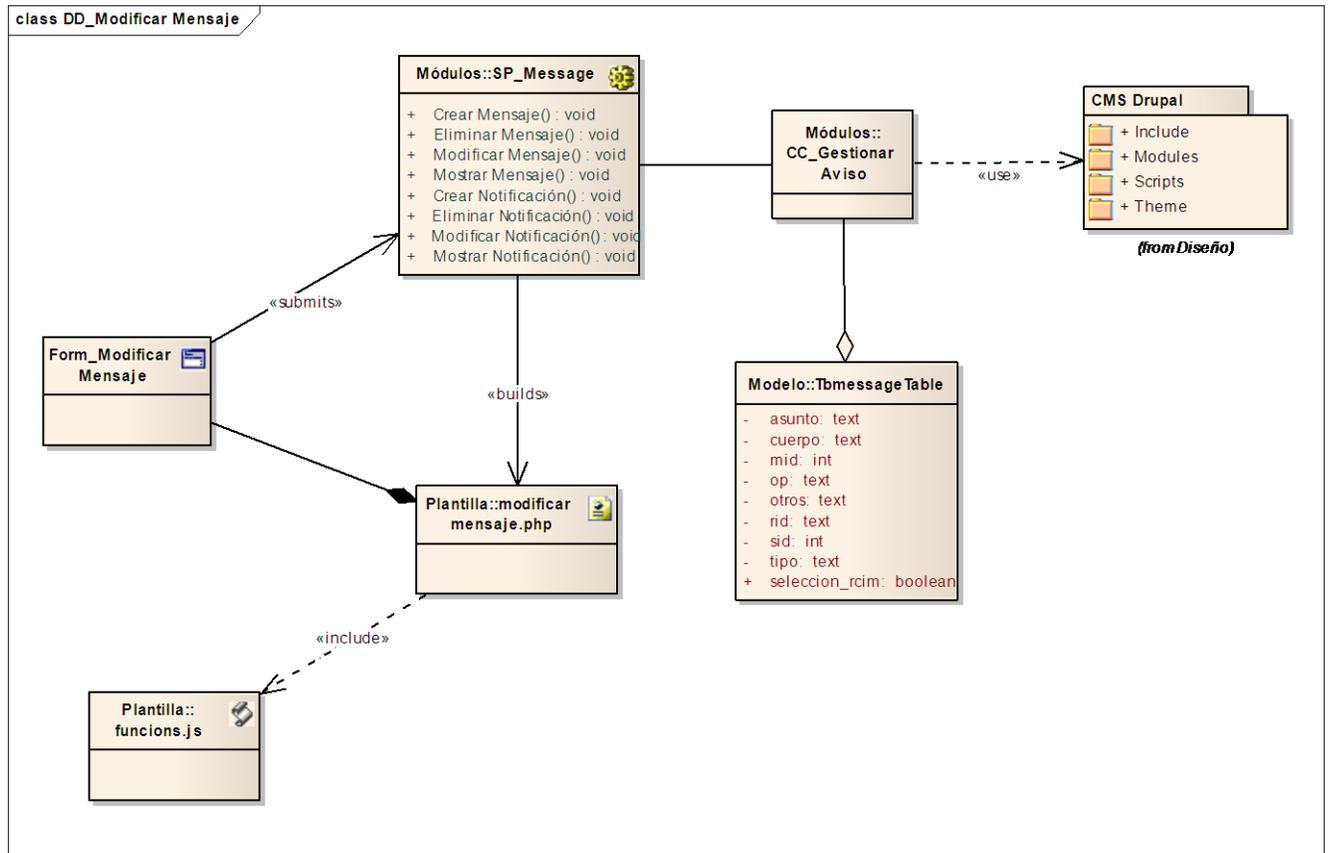


Fig.11. Diagrama de Clases del Diseño CU\_Modificar Mensaje.

**Descripción de clases de diseño web.**

**Clase CP\_client page.**

El propósito de estas clases es representar el acceso del usuario al sistema, se encargan de atender todas las selecciones y peticiones que pueda realizar un determinado usuario al sistema. A continuación se muestra un ejemplo de una de ellas ya que todas se comportan de igual manera.

*Ejemplo:*

Clase CP\_Crear Mensaje.php. El propósito de esta clase es representar como se crea un nuevo mensaje.

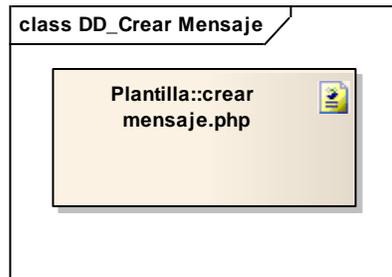


Fig.12. CU\_ Crear Mensaje. CP\_Crear Mensaje.php.

### Formularios.

Las clases formularios representan la entrada y salida de datos en el sistema. Seguidamente se muestra un ejemplo de la descripción y el propósito de un formulario, debido a que todos los formularios se comportan de manera similar.

*Ejemplo:*

Clase Form\_ Crear Mensaje. El propósito de esta clase es representar cómo se lleva a cabo la inserción de los datos del mensaje.

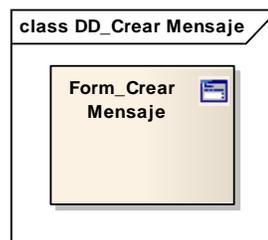


Fig.13. CU\_ Crear Mensaje. Form\_ Crear Mensaje.

### Clase SP\_server page.

Las clases server page son las encargadas de manejar los eventos de la aplicación, seguidamente se muestra un ejemplo de una de ella, debido a que todas se comportan de manera similar.

*Ejemplo:*

Clase SP\_Message. El propósito de esta clase es que maneja los eventos de la aplicación.

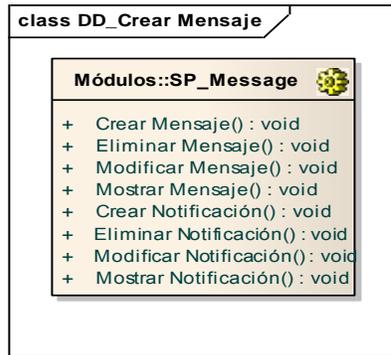


Fig.14. CU\_ Crear\_Mensaje. SP\_Message.

**Diagrama de Interacción.**

Un diagrama que muestra un interacción, consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos, los diagramas de interacción trata la vista dinámica de un sistema, un término genérico que se aplica a varios tipos de diagramas que enfatizan las interacciones de objetos, incluyendo diagramas de secuencia, diagramas de colaboración y diagramas de actividad. (40)

**Diagrama de Comunicación.**

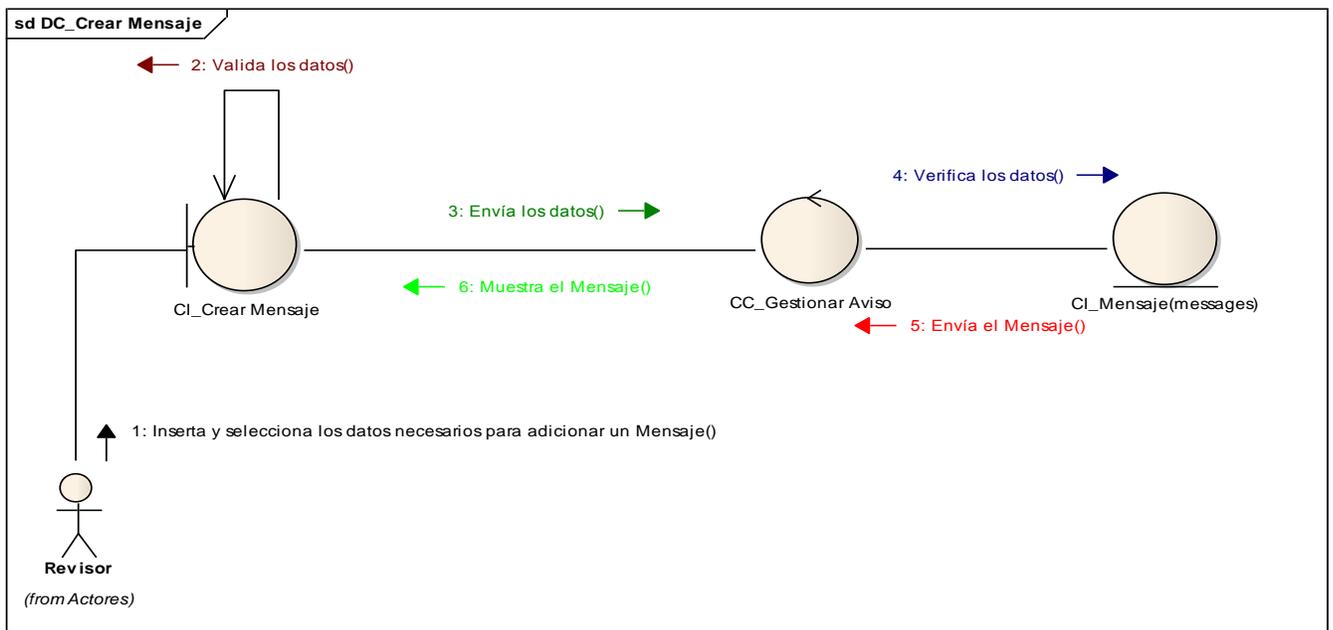


Fig.15. Diagrama de Comunicación CU\_ Crear Mensaje.

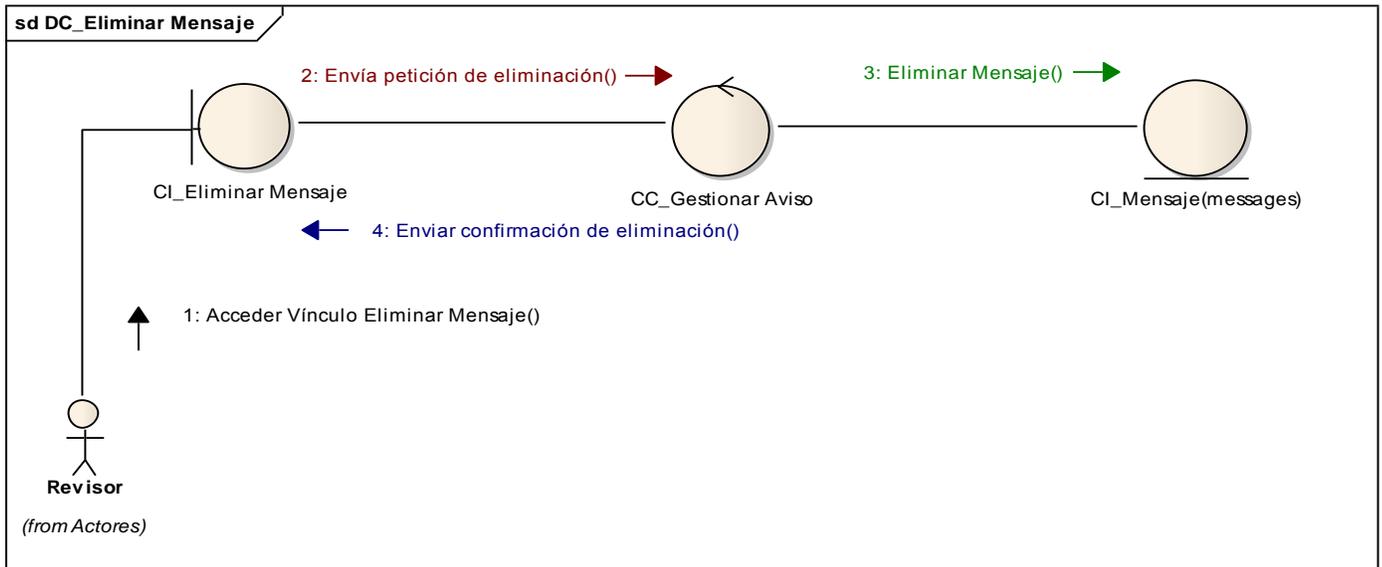


Fig.16. Diagrama de Comunicación CU\_ Eliminar Mensaje.

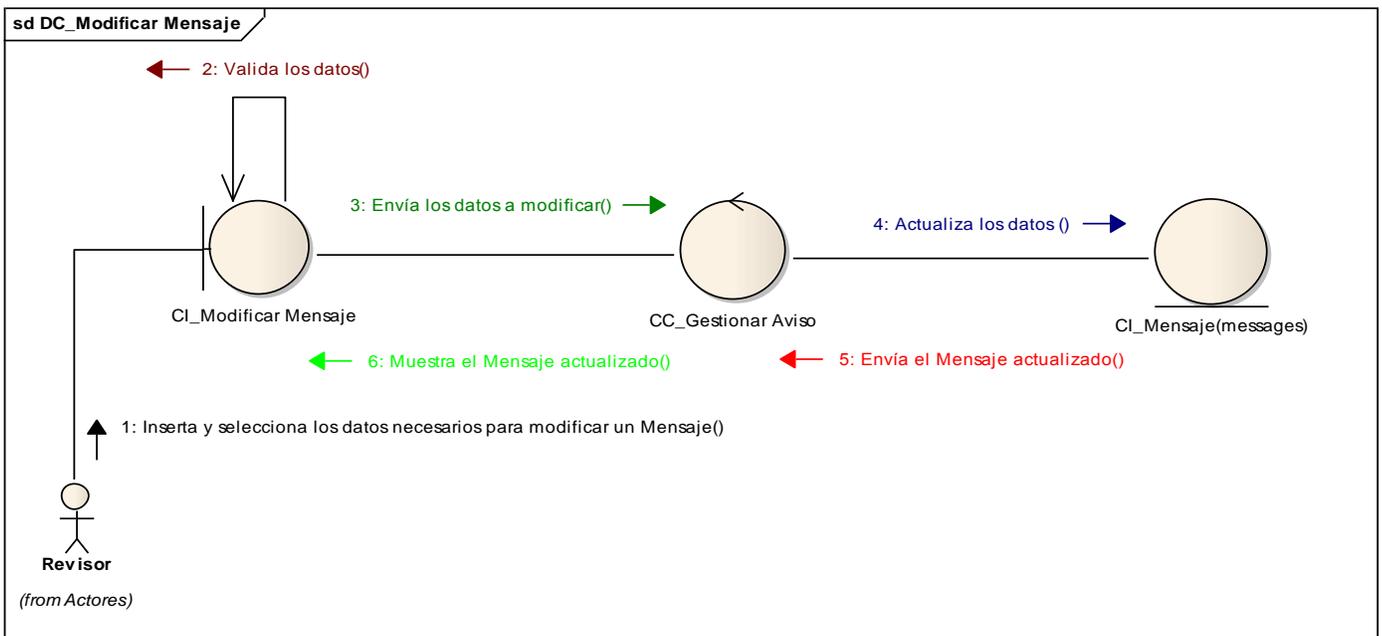


Fig.17. Diagrama de Comunicación CU\_ Modificar Mensaje.

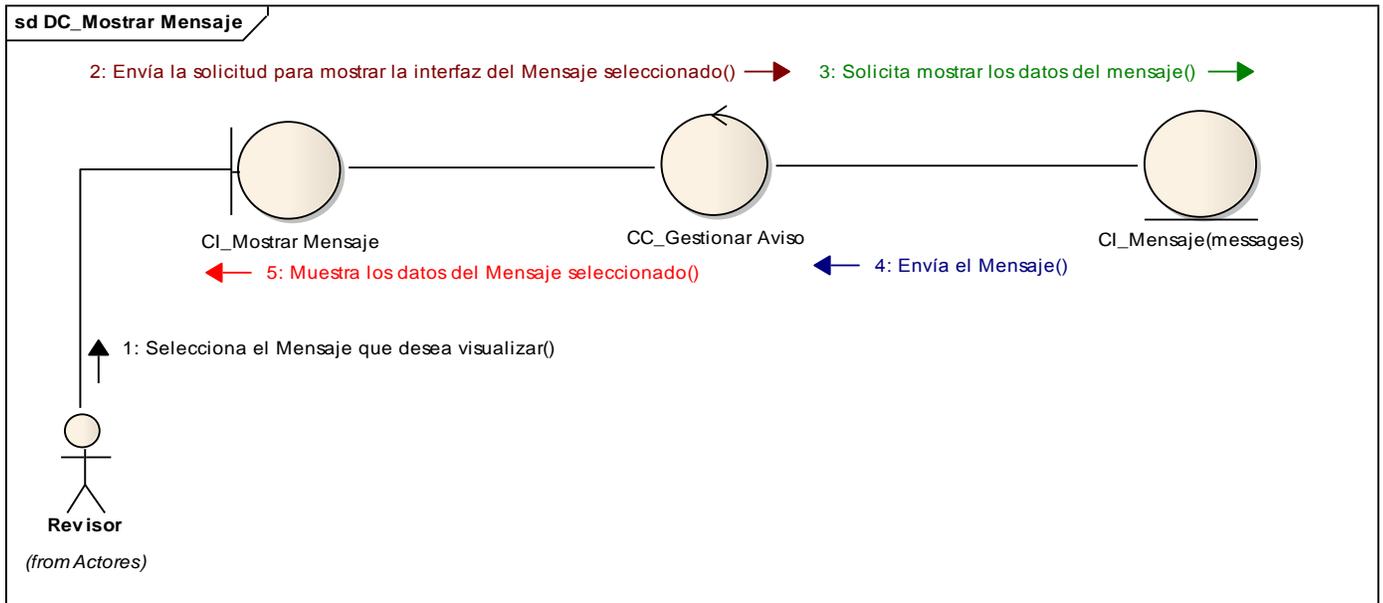


Fig.18. Diagrama de Comunicación CU\_Mostrar Mensaje.

**Diagrama de Secuencia.**

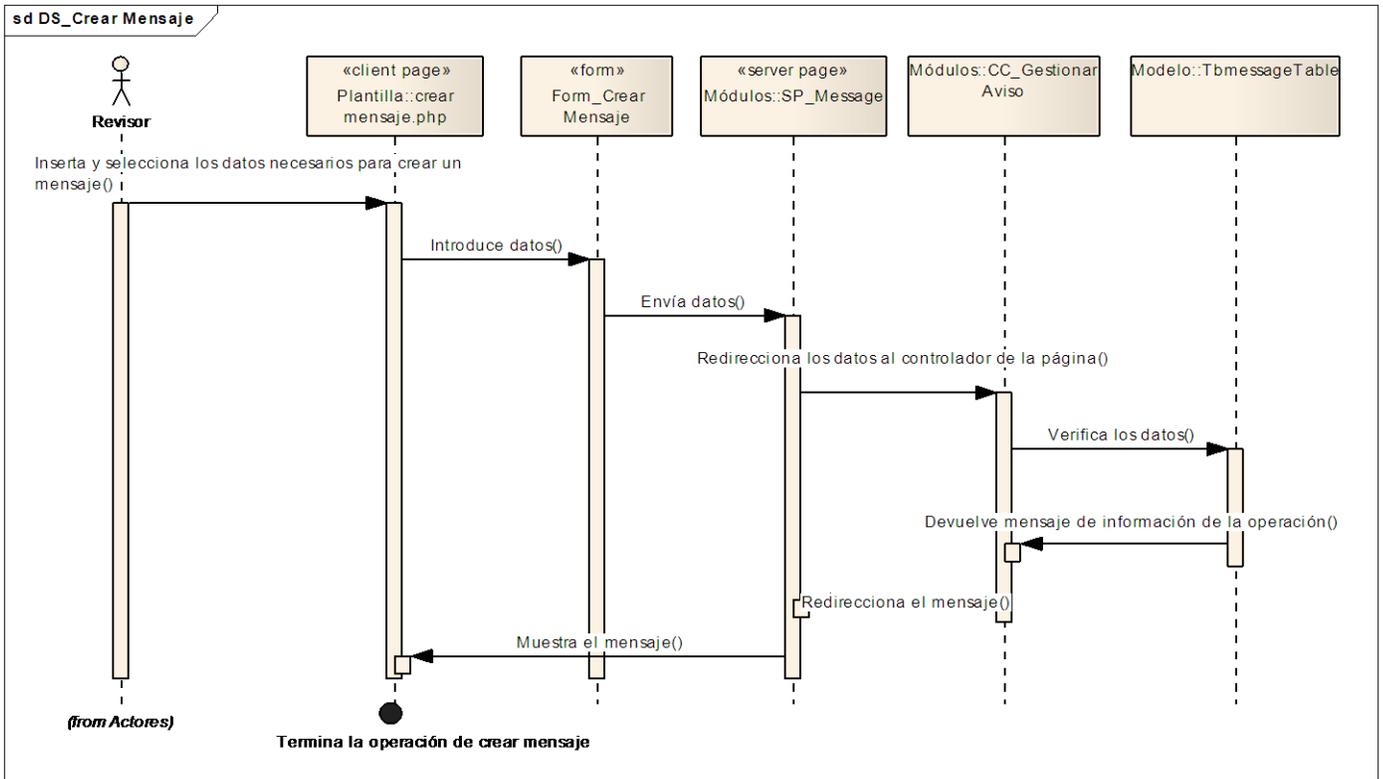


Fig.19. Diagrama de Secuencia CU\_ Crear Mensaje.

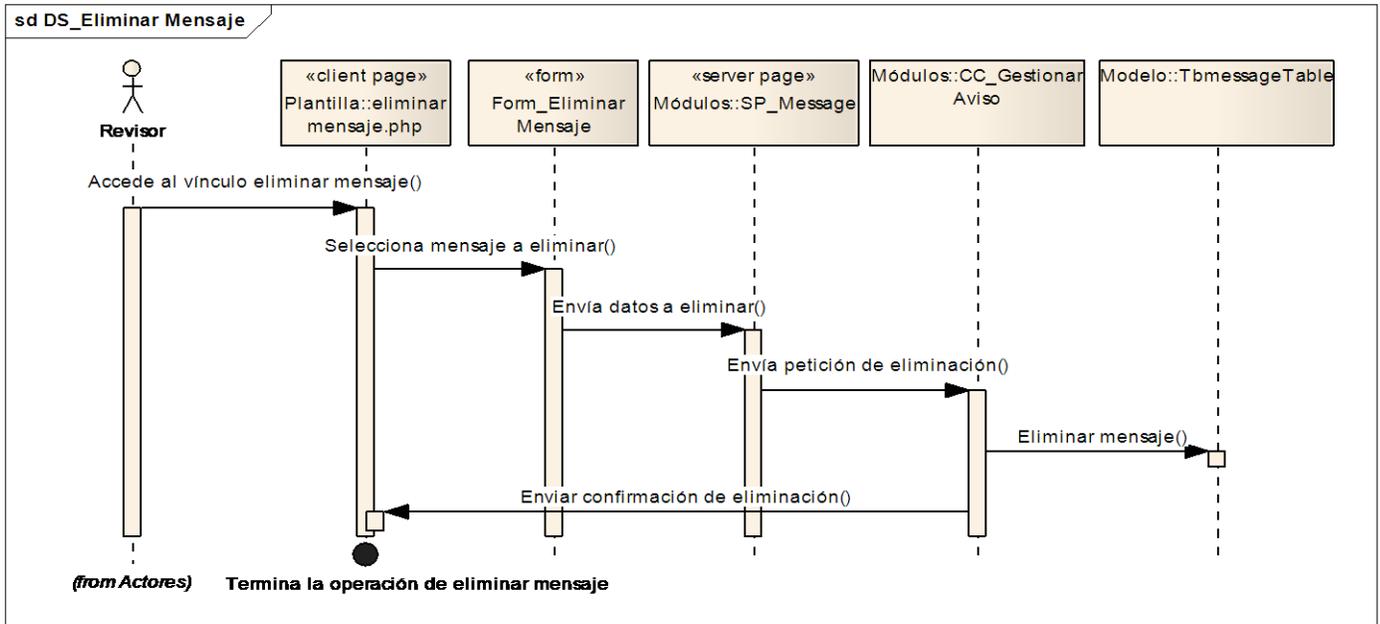


Fig.20. Diagrama de Secuencia CU\_Eliminar Mensaje.

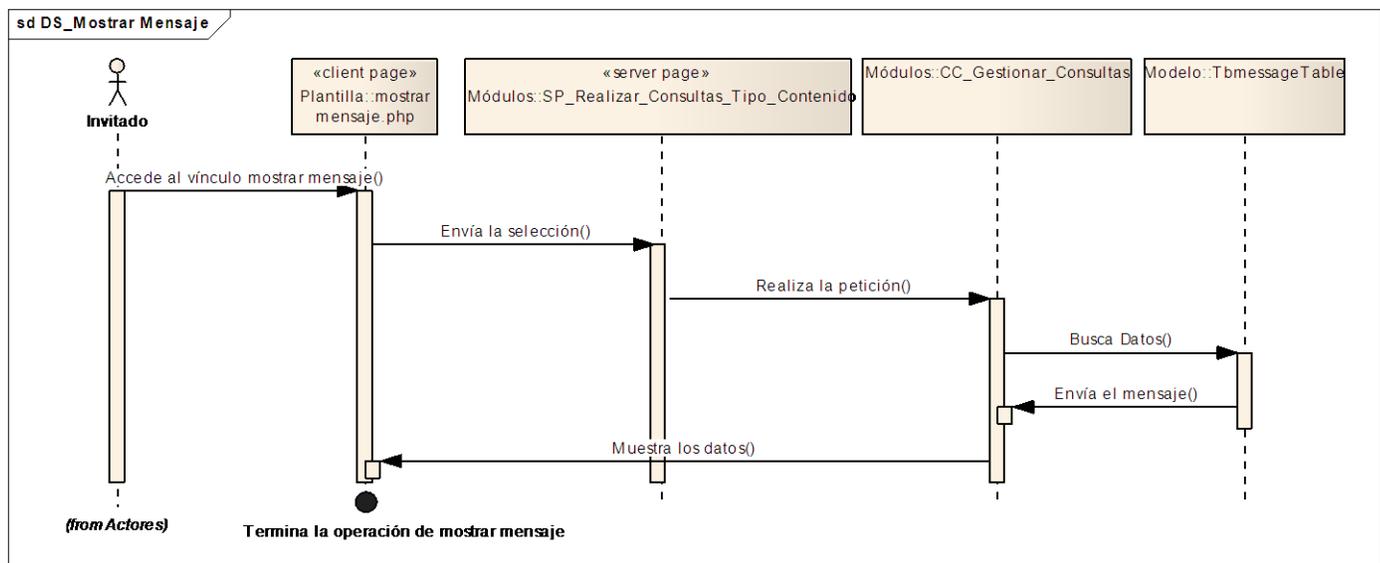


Fig.21. Diagrama de Secuencia CU\_Mostrar Mensaje.

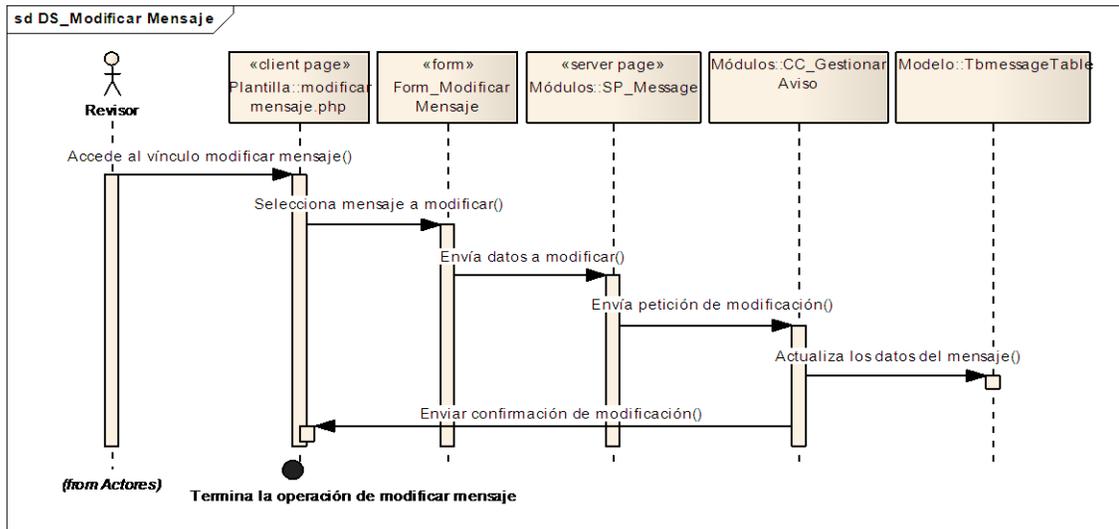


Fig.22. Diagrama de Secuencia CU\_Modificar Mensaje.

### 3.3 Modelo de Datos.

Un modelo de datos es básicamente, una descripción de un contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son elementos físicos, son abstracciones que permiten la implementación de un sistema eficiente de base de datos. Por lo general, se refieren a algoritmos y conceptos matemáticos; y aportan la base conceptual para diseñar aplicaciones que hacen uso intensivo de datos.

Para el funcionamiento del sistema se diseñó un modelo de datos de clases persistentes que describe los distintos datos, de la información que se manejará en el software. A continuación se muestra el Modelo de Datos definido para describir la representación lógica y física de la información persistente que se maneja en el Portal Web de la SOCIM. (41)

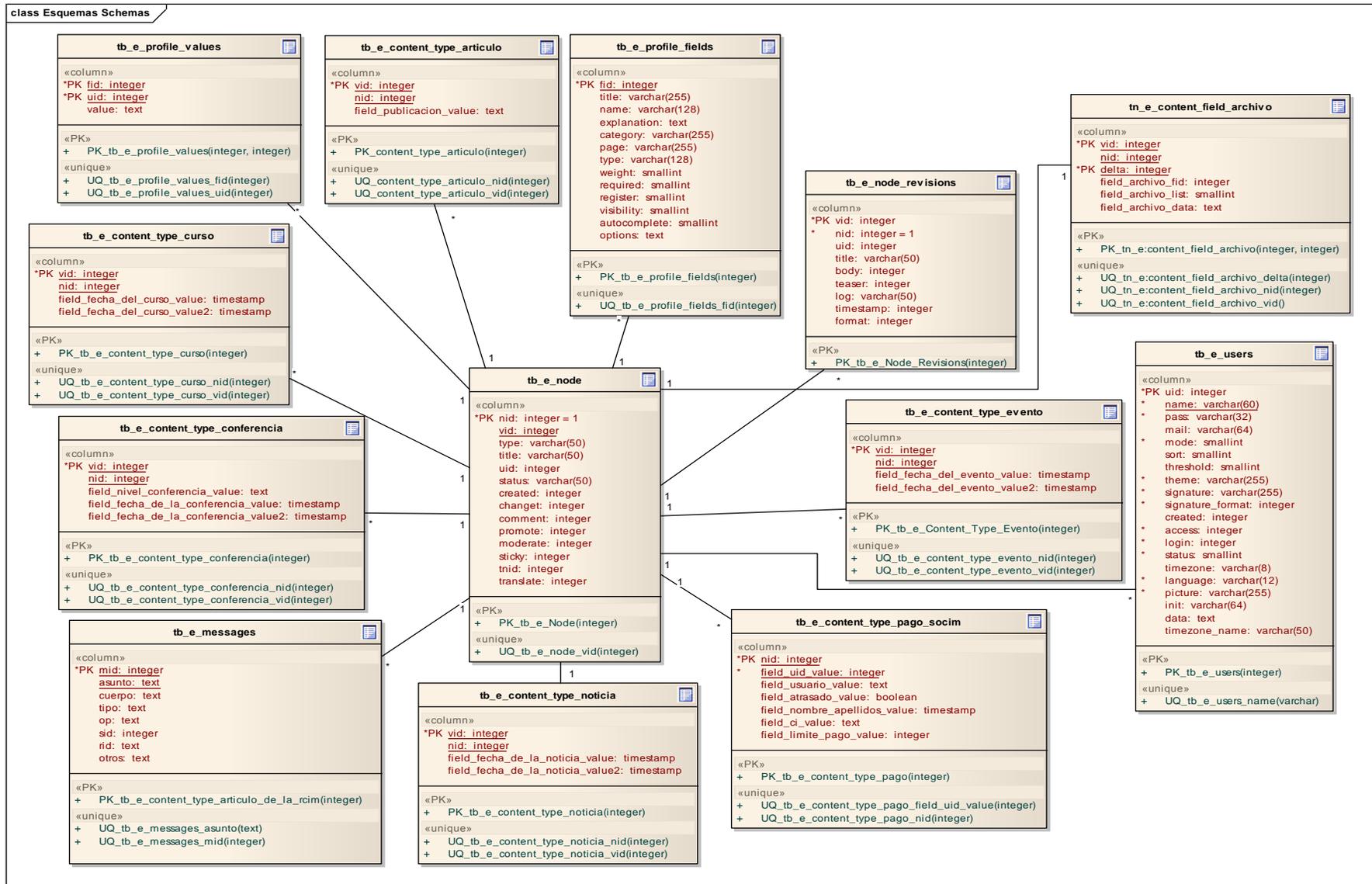


Fig.23. Modelo de Datos.

## 3.1.1 Descripción de las Tablas.

Nombre: content_type_articulo		
<b>Descripción:</b> Contiene todos los datos del tipo de contenido artículo.		
Atributo	Tipo	Descripción
vid	int	Identificador de la tabla vocabulary.
nid	int	Identificador de la tabla node.
field_publicacion_value	text	Campo del sitio para publicar el artículo.

Tabla 7. Descripción de la tabla tb\_content\_type\_articulo.

Nombre: content_type_noticia		
<b>Descripción:</b> Contiene todos los datos de el campo del archivo a subir para el portal.		
Atributo	Tipo	Descripción
vid	int	Identificador de la tabla vocabulary.
nid	int	Identificador de la tabla node.
field_fecha_de_la_noticia_value	timestamp	Fecha de la noticia
field_fecha_de_la_noticia_value2	timestamp	Fecha de la publicación de la noticia.

Tabla 8. Descripción de la tabla tb\_content\_type\_noticia.

Nombre: content_type_evento		
<b>Descripción:</b> Contiene todos los datos de el campo del archivo a subir para el portal.		
Atributo	Tipo	Descripción
vid	int	Identificador de la tabla vocabulary.
nid	int	Identificador de la tabla node.
field_fecha_del_evento_value	timestamp	Fecha del evento.

field_fecha_del_evento_value2	timestamp	Fecha de la publicación del evento.
-------------------------------	-----------	-------------------------------------

Tabla 9. Descripción de la tabla tb\_content\_type\_evento.

Nombre: content_type_curso		
<b>Descripción:</b> Contiene todos los datos de el campo del archivo a subir para el portal.		
Atributo	Tipo	Descripción
vid	int	Identificador de la tabla vocabulary.
nid	int	Identificador de la tabla node.
field_fecha_del_curso_value	timestamp	Fecha del curso.
field_fecha_del_curso_value2	timestamp	Fecha de la publicación del curso.

Tabla 10. Descripción de la tabla tb\_content\_type\_curso.

Nombre: content_field_archivo		
<b>Descripción:</b> Contiene todos los datos de el campo del archivo a subir para el portal.		
Atributo	Tipo	Descripción
vid	int	Identificador de la tabla vocabulary.
nid	int	Identificador de la tabla node.
field_archivo_fid	int	Identificador de la tabla content_field_archivo.

Tabla 11. Descripción de la tabla tb\_content\_field\_archivo.

Nombre: content_type_pago		
<b>Descripción:</b> Contiene todos los datos de el campo del archivo a subir para el portal.		
Atributo	Tipo	Descripción
uid	int	Identificador del usuario del pago.
nid	int	Identificador de la tabla node.

field_nombre_apellidos_value	text	Campo de el nombre y los apellidos del asociado que realiza el pago.
field_ci_value	text	Campo de el carne de identidad del asociado que realiza el pago.
field_uid_value	text	Identificador de la tabla user.
field_limite_pago_value	integer	Fecha límite del último pago del asociado.
field_usuario_value	text	Nombre del usuario que realiza el pago.
field_pago_value	text	Campo que indica si el asociado pagó o no atrasado ese año.

Tabla 12. Descripción de la tabla tb\_content\_type\_pago.

### 3.4 Modelo de Despliegue.

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución.

En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes.

En este caso las computadoras clientes tienen la posibilidad de interactuar a través del protocolo HTTP con el Servidor de Aplicaciones y este último mediante el protocolo TCP/IP interactúa con el Servidor de Base de Datos que posee instalado PostgreSQL como SGBD. (42)

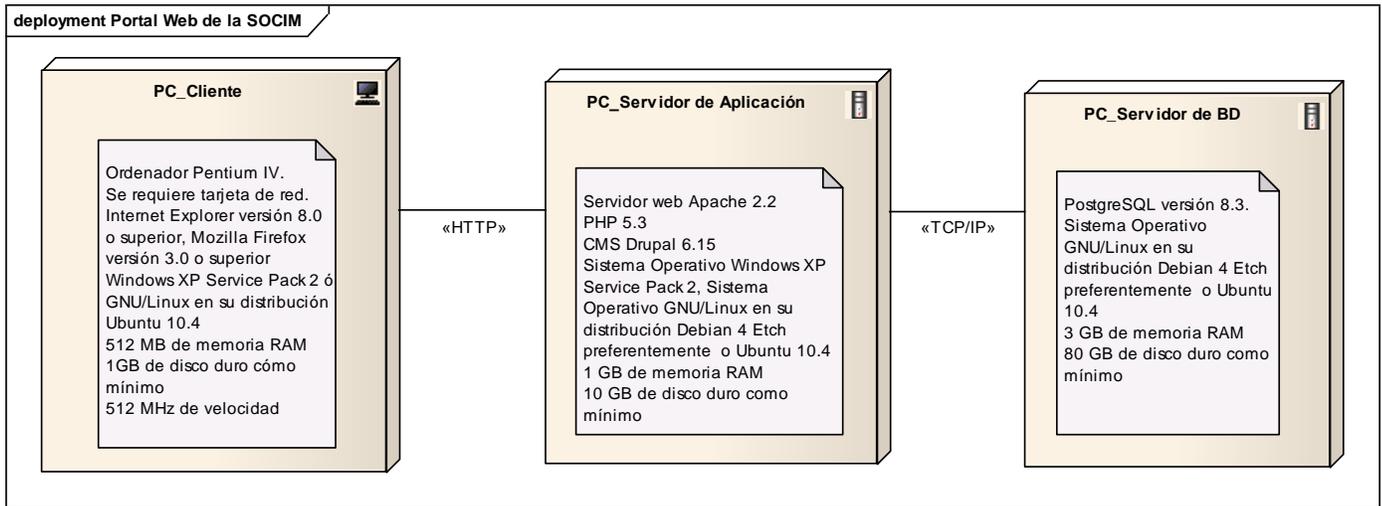


Fig. 24. Modelo de Despliegue

Durante el desarrollo del este capítulo se logró modelar el sistema de manera que soporte los requisitos funcionales y no funcionales. A partir del modelado de los diagramas de análisis, diseño e interacción, se muestra la estructura del funcionamiento del sistema internamente. Se realizó la descripción de la arquitectura del software de acuerdo a las características y estructura, propuestas por el CMS Drupal.

## CAPÍTULO 4: IMPLEMENTACIÓN.

El trabajo en este capítulo parte del resultado del análisis y diseño obtenidos en el capítulo anterior y se procederá a implementar las clases y subsistemas en términos de componentes (ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares). Además, se explicarán las estrategias de codificación, tratamiento de errores y seguridad. Se construye el Modelo de Despliegue, donde se representa la estructura física del sistema, y se describe el Modelo de Datos sobre el cual se va a realizar la implementación.

### 4.1 Modelo de Implementación.

El modelo de implementación es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros de código fuente, y otros tipos de ficheros necesarios para la implantación y despliegue del sistema. Describe también, como se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen de los componentes unos de otros. (43)

#### 4.1.1 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente.

Algunos estereotipos estándar de componentes son los siguientes:

- Ejecutable: Es un programa que se puede ejecutar en un nodo.
- Biblioteca: Es una biblioteca de objetos estática o dinámica.
- Tabla: Es una tabla de una BD.
- Archivo: Es un fichero que contiene código fuente o datos.
- Documento: Es un documento.
- Página Web: Es una página que se obtiene de la ejecución del sistema.

Los componentes tienen las siguientes características:

- Tienen relaciones de traza con los elementos del modelo que implementan.

➤ Pueden implementar varios elementos. Por ejemplo, varias clases. Sin embargo la forma exacta en que se crea esta traza depende de cómo van a ser estructurados y modularizados los ficheros de código fuente, dado el lenguaje de programación que se esté usando. (44)

A continuación se describe por detallados los diagramas de componentes del sistema desarrollado. El primer paquete a analizar se constituye del CMS Drupal, se explican todos sus subpaquetes funcionales y se brinda el diagrama general:

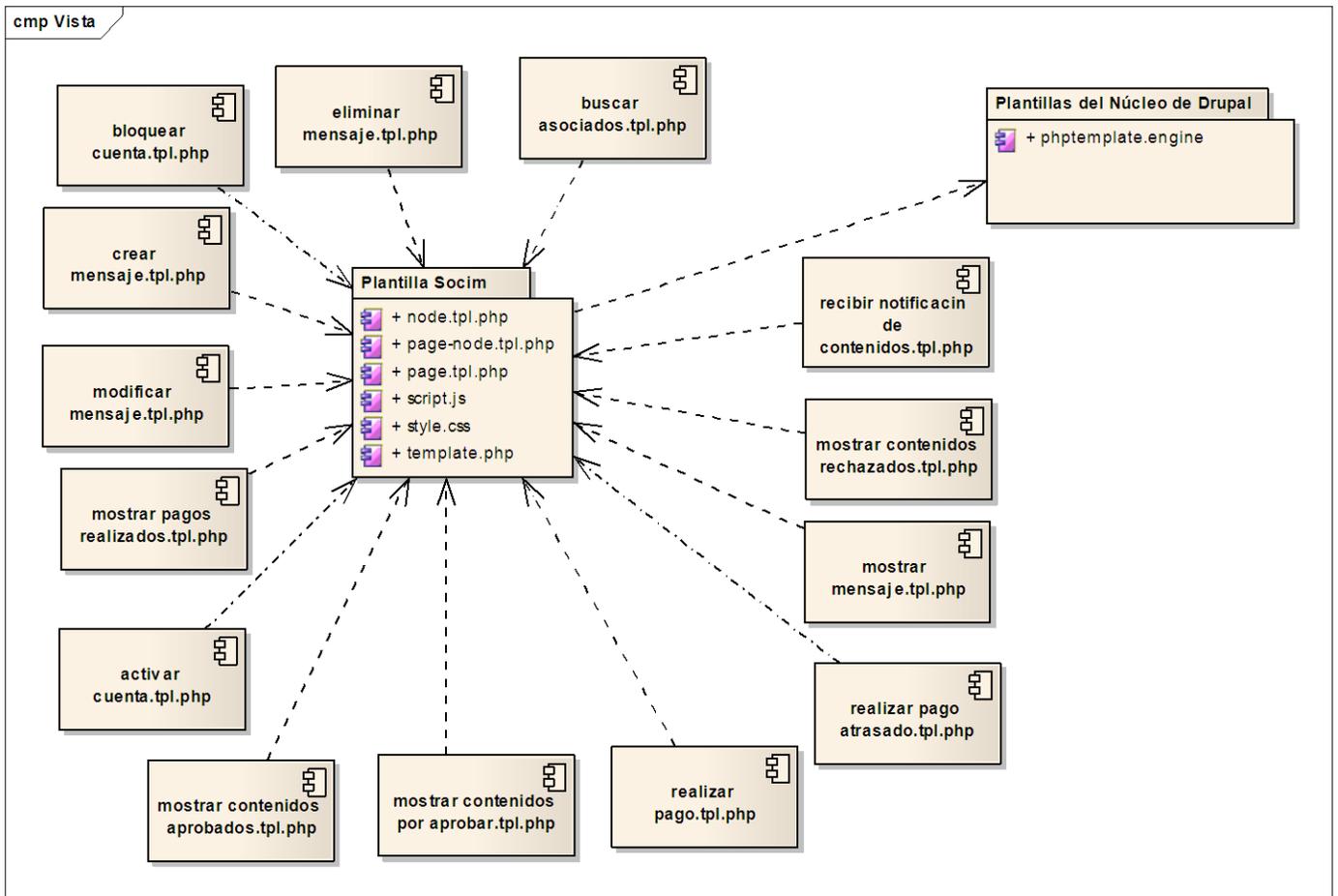


Fig. 25. Modelo de Componentes. Componente Vista del CMS Drupal.

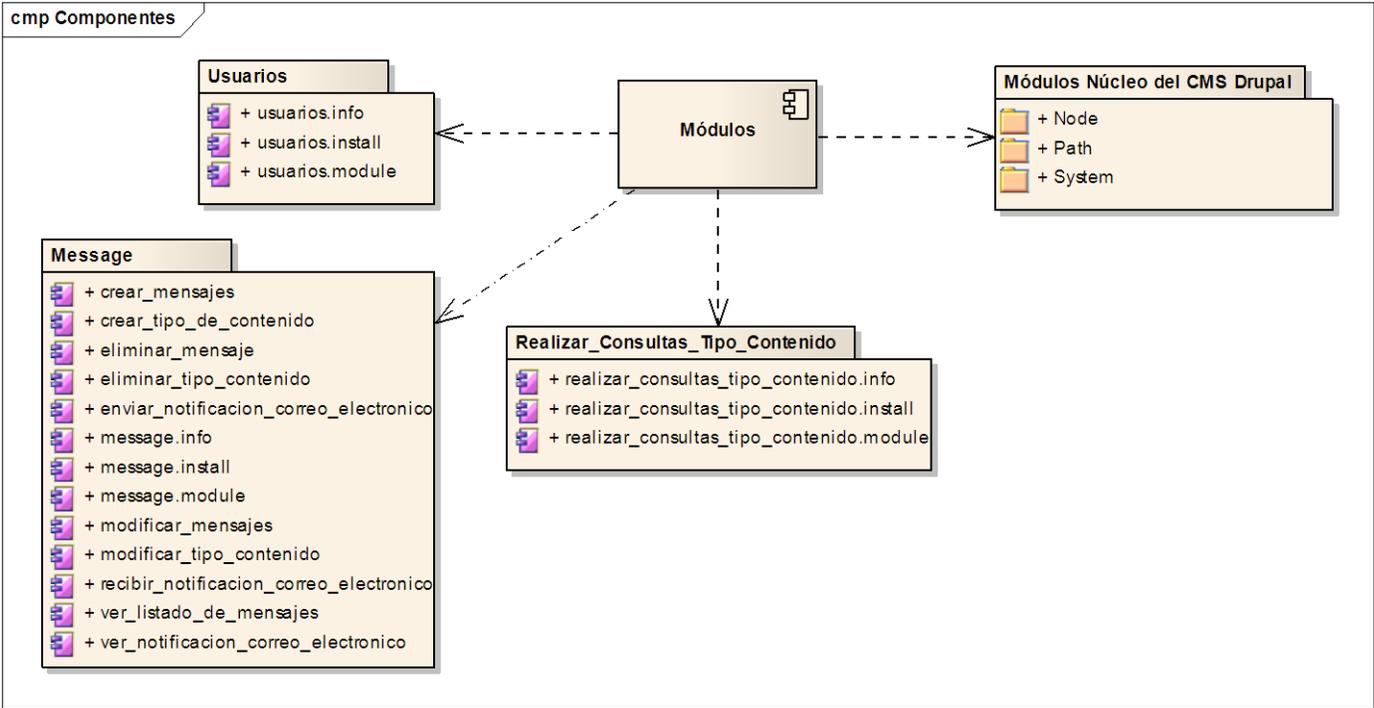


Fig. 26. Modelo de Componentes. Componente Controlador del CMS Drupal.

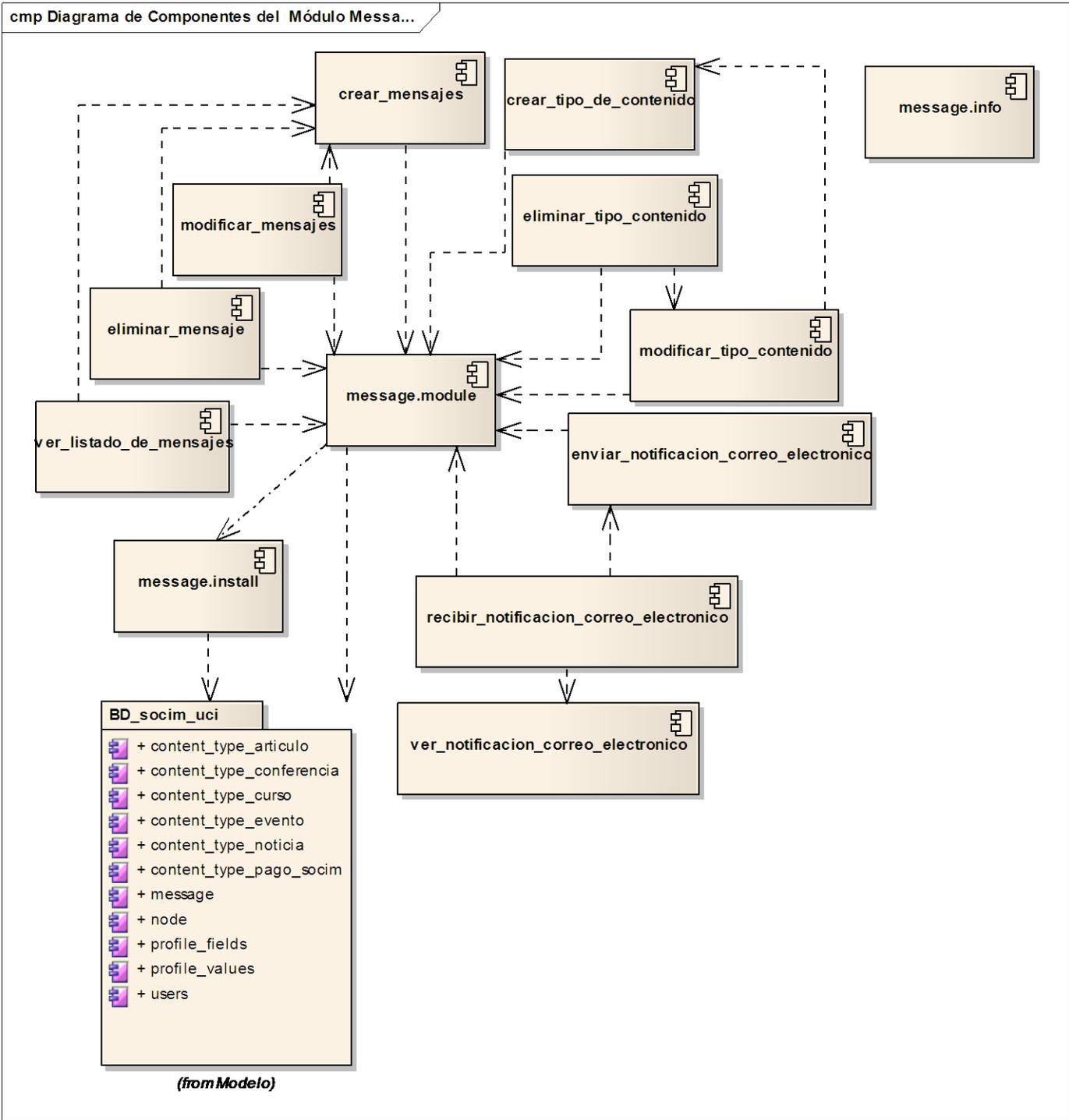


Fig. 27. Modelo de Componentes del Módulo Message.

Durante la confección del capítulo que recién concluye se abrdaron cuestiones pertenecientes a la etapa de implementación. Obteniéndose el modelo de implementación, con los diagramas de componentes. El sistema permite realizar la inscripción y autenticación de los asociados, así como la publicación de artículos, realizar notificaciones automaticas y registrar pagos de los asociados.

## CONCLUSIONES

- El estudio del estado del arte permitió analizar varios portales concebidos para la gestión de información de informática médica a nivel internacional y nacional, de los cuales se tomaron las mejores prácticas para incorporarlas al Portal SOCIM. No se pudo adoptar ninguno, pues no se ajustan a las necesidades de la sociedad, por sus peculiaridades en cuanto a la forma de inscripción y al pago de las cuotas de afiliación, principalmente.
- Se asimilaron las tecnologías y herramientas que han sido definidas por el CESIM, por ser idóneas para darle solución al problema planteado. Constituyen tecnologías de punta que recogen las mejores prácticas para el desarrollo de software y la mayoría son libres.
- La metodología RUP permitió documentar todo el proceso de desarrollo del software mediante los artefactos propuestos en los flujos de trabajo Modelado del Negocio, Requerimientos, Análisis y Diseño e Implementación, facilitando la obtención, análisis, diseño e implementación de las funcionalidades del Portal.
- Se implementó el Portal Web de la Sociedad Cubana de Informática Médica que satisface las necesidades de la SOCIM y sus asociados. Este permite la inscripción y autenticación en línea, la publicación automatizada de artículos. Además realiza notificaciones automáticas para la revisión de los artículos de la Revista Cubana de Informática Médica, el registro del pago de los asociados, foros, eventos, conferencias, cursos y chat.

## RECOMENDACIONES

Los objetivos de este trabajo han sido logrados, teniendo en cuenta que se cumplieron todos los requerimientos planteados. No obstante, para futuras investigaciones y proyectos que guarden relación con este trabajo se hacen las siguientes recomendaciones:

- Implementar un nuevo procedimiento para las notificaciones del proceso de revisión de los artículos de la Revista Cubana de Informática Médica, donde se puedan enviar mensajes a usuarios específicos; permitiendo de esta forma asignar las responsabilidades en el proceso de revisión.
- Realizar un módulo de pago online donde el asociado pueda insertar un número de cuenta, el identificador de su tarjeta magnética y un saldo para efectuar el pago anual, sin tener que ir a una oficina regional o central de la SOCIM.

## TRABAJOS CITADOS

1. **Cañedo Andalia, Rubén Lic. y Ramos Ocho, Raúl E. Dr. .** Biblioteca Virtual en Salud. [En línea] [Citado el: 25 de 10 de 2010.] [http://bvs.sld.cu/revistas/aci/vol13\\_5\\_05/aci07505.htm](http://bvs.sld.cu/revistas/aci/vol13_5_05/aci07505.htm).
2. **Perezleo Solórzano, Ligeya , y otros.** Biblioteca Virtual en Salud. [En línea] [Citado el: 25 de 10 de 2010.] [http://bvs.sld.cu/revistas/aci/vol11\\_4\\_03/aci07403.htm](http://bvs.sld.cu/revistas/aci/vol11_4_03/aci07403.htm).
3. **MiSabueso.** [En línea] [Citado el: 25 de 10 de 2010.] <http://www.misabueso.com/directorio/Regional/Am%C3%A9rica/Cuba/Salud/Asociaciones/>.
4. **SIEGFRIED RHEIN.** [En línea] [Citado el: 25 de 10 de 2010.] [http://www.siegfried.com.mx/index.php?option=com\\_content&view=article&id=23&Itemid=6&lang=es](http://www.siegfried.com.mx/index.php?option=com_content&view=article&id=23&Itemid=6&lang=es).
5. **SOCIM.** [En línea] [Citado el: 20 de 10 de 2010.] <http://www.socim.sld.cu/index.htm>.
6. **RCIM.** [En línea] [Citado el: 25 de 10 de 2010.] [http://www.rcim.sld.cu/nota\\_a\\_los\\_autores.htm](http://www.rcim.sld.cu/nota_a_los_autores.htm).
7. **Lenguajes de Ciencia de la Computación.** Universidad de Málaga. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.lcc.uma.es/~jignacio/index\\_archivos/TEMA4.pdf](http://www.lcc.uma.es/~jignacio/index_archivos/TEMA4.pdf).
8. **Zapata, Carlos Mario y Garcés, Gilma Liliana.** Revista EIA. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.scielo.unal.edu.co/scielo.php?pid=S1794-12372008000200008&script=sci\\_arttext](http://www.scielo.unal.edu.co/scielo.php?pid=S1794-12372008000200008&script=sci_arttext).
9. **Buenastareas.** [En línea] [Citado el: 25 de 10 de 2010.] <http://www.buenastareas.com/temas/modelo-del-dominio-de-un-hotel/140>.
10. **Masadelante.** [En línea] [Citado el: 25 de 10 de 2010.] <http://www.masadelante.com/faqs/tipos-de-servidores>.
11. **GestioPolis.** [En línea] [Citado el: 25 de 10 de 2010.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/laodelsxxi-2.htm>.
12. **Sajú.** [En línea] [Citado el: 25 de 10 de 2010.] [http://www.google.com/url?sa=t&source=web&cd=8&ved=0CEEQFjAH&url=http%3A%2F%2Fwww.miportal.edu.sv%2Fportal%2Fcomunidad%2Fdfs-filesystemfile.ashx%2F\\_\\_key%2FCommunityServer.Components.PostAttachments%2F00.00.04.41.83%2Fbases-de-datos.docx&rct=j&q=El%20Gesto](http://www.google.com/url?sa=t&source=web&cd=8&ved=0CEEQFjAH&url=http%3A%2F%2Fwww.miportal.edu.sv%2Fportal%2Fcomunidad%2Fdfs-filesystemfile.ashx%2F__key%2FCommunityServer.Components.PostAttachments%2F00.00.04.41.83%2Fbases-de-datos.docx&rct=j&q=El%20Gesto).

13. Slideboom. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.slideboom.com/presentations/21683/Introduccion.PostgreSQL>.
14. postgresql. [En línea] [Citado el: 20 de 10 de 2010.] [http://www.guia-ubuntu.org/index.php?title=PgAdmin\\_III](http://www.guia-ubuntu.org/index.php?title=PgAdmin_III).
15. Abcdatos. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.abcdatos.com/webmasters/programa/l9126.html>.
16. Killersites. [En línea] [Citado el: 25 de 10 de 2010.] [www.killersites.com.ar/cursos-fireworks.htm](http://www.killersites.com.ar/cursos-fireworks.htm).
17. Artister. [En línea] [Citado el: 25 de 10 de 2010.] <http://artisteer.malavida.com/>.
18. Descargar. [En línea] [Citado el: 25 de 10 de 2010.] <http://descargar.k77.eu/programas/descargas/artisteer.php>.
19. Slideshare. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.slideshare.net/alinacarrion/html-exposicion>.
20. **Tortosa del Toro, Enrique.** [En línea] [Citado el: 25 de 10 de 2010.] [http://enriquetortosa.com/descargas/guia\\_diseno\\_web\\_basico.pdf](http://enriquetortosa.com/descargas/guia_diseno_web_basico.pdf).
21. Consorcio World Wide Web (W3C). [En línea] [Citado el: 25 de 10 de 2010.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
22. Vas libre. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.vaslibre.org.ve/publicaciones/phpflisol2006.pdf>.
23. [En línea] [Citado el: 25 de 10 de 2010.] <http://es.scribd.com/doc/54037649/24/Beneficios-de-la-Plataforma-OSGi>.
24. Softonic. [En línea] [Citado el: 25 de 10 de 2010.] <http://netbeans-ide.softonic.com/>.
25. Paratupagina. [En línea] [Citado el: 25 de 10 de 2010.] <http://paratupagina.com/topic/415-que-es-drupal/>.
26. Visión General de la Ingeniería de Software. [En línea] [Citado el: 25 de 10 de 2010.] <http://barranquillo.ucaldas.edu.co/rgarcia/ingsoft/visiongral.htm>.

27. Componentes De Aplicaciones Cliente Servidor. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.mitecnologico.com/Main/ComponentesDeAplicacionesClienteServidor>.
28. mitecnologico. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.mitecnologico.com/Main/ComponentesDeAplicacionesClienteServidor>.
29. Kioskea. [En línea] [Citado el: 25 de 10 de 2010.] <http://es.kioskea.net/contents/cs/csintro.php3>.
30. Crossposting from IdeSeg.com. [En línea] [Citado el: 25 de 10 de 2010.] <http://geeks.ms/blogs/csegura/>.
31. Bdigital. [En línea] [Citado el: 25 de 10 de 2010.] <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDT629/capitulo6.pdf>.
32. mitecnologico. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.mitecnologico.com/Main/ProcesosDeLaIngenieriaDeRequerimientos>.
33. Ídem 27. [En línea]
34. Ídem 29. [En línea]
35. Ídem 27. [En línea]
36. scribd. [En línea] [Citado el: 25 de 10 de 2010.] <http://es.scribd.com/doc/44942492/Presentacion-Guia-RUP>.
37. Xuletas. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.xuletas.es/ficha/adfadsf/>.
38. La Queb de Joaquín. [En línea] [Citado el: 25 de 10 de 2010.] <http://jms32.eresmas.net/tacticos/UML/UML04/UML0401.html>.
39. Upcommons. [En línea] [Citado el: 25 de 10 de 2010.] <http://upcommons.upc.edu/pfc/bitstream/2099.1/9749/1/65350.pdf>.
40. Unican. [En línea] [Citado el: 25 de 10 de 2010.] <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-i/materiales-de-clase-1/is1-t02-trans.pdf>.
41. Mitecnologico. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.mitecnologico.com/Main/DefinicionModeloDeDatos>.

42. Kybele. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.kybele.etsii.urjc.es/docencia/IS\\_ITSIT/2010-2011/Material/TEMAV\\_Dise%C3%B1o.pdf](http://www.kybele.etsii.urjc.es/docencia/IS_ITSIT/2010-2011/Material/TEMAV_Dise%C3%B1o.pdf).
43. Risktechnology. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.risktechnology.net/metodologia-rup.aspx?skin=printerfriendly>.
44. Dsi. [En línea] [Citado el: 25 de 10 de 2010.] [www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf](http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf).

## BIBLIOGRAFÍA

- [En línea] [Citado el: 25 de 10 de 2010.] <http://es.scribd.com/doc/54037649/24/Beneficios-de-la-Plataforma-OSGi>.
- Abcdatos. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.abcdatos.com/webmasters/programa/l9126.html>.
- Artister. [En línea] [Citado el: 25 de 10 de 2010.] <http://artisteer.malavida.com/>.
- Bdigital. [En línea] [Citado el: 25 de 10 de 2010.] <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDT629/capitulo6.pdf>.
- Buenastareas. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.buenastareas.com/temas/modelo-del-dominio-de-un-hotel/140>.
- **Cañedo Andalia, Rubén Lic. y Ramos Ocho, Raúl E. Dr. .** Biblioteca Virtual en Salud. [En línea] [Citado el: 25 de 10 de 2010.] [http://bvs.sld.cu/revistas/aci/vol13\\_5\\_05/aci07505.htm](http://bvs.sld.cu/revistas/aci/vol13_5_05/aci07505.htm).
- Componentes De Aplicaciones Cliente Servidor. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.mitecnologico.com/Main/ComponentesDeAplicacionesClienteServidor>.
- Consorcio World Wide Web (W3C). [En línea] [Citado el: 25 de 10 de 2010.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
- Crossposting from IdeSeg.com. [En línea] [Citado el: 25 de 10 de 2010.] <http://geeks.ms/blogs/csegura/>.
- Descargar. [En línea] [Citado el: 25 de 10 de 2010.] <http://descargar.k77.eu/programas/descargas/artisteer.php>.
- Dsi. [En línea] [Citado el: 25 de 10 de 2010.] [www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf](http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf).
- GestioPolis. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/laodelsxxi-2.htm>.
- **Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2004. ISBN. *El proceso unificado de desarrollo de software*.** España : Addison Wesley, 2004. ISBN.

- Killersites. [En línea] [Citado el: 25 de 10 de 2010.] [www.killersites.com.ar/cursos-fireworks.htm](http://www.killersites.com.ar/cursos-fireworks.htm).
- Kioskea. [En línea] [Citado el: 25 de 10 de 2010.] <http://es.kioskea.net/contents/cs/csintro.php3>.
- Kybele. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.kybele.etsii.urjc.es/docencia/IS\\_TSIT/2010-2011/Material/TEMAV\\_Dise%C3%B1o.pdf](http://www.kybele.etsii.urjc.es/docencia/IS_TSIT/2010-2011/Material/TEMAV_Dise%C3%B1o.pdf).
- La Queb de Joaquín. [En línea] [Citado el: 25 de 10 de 2010.] <http://jms32.eresmas.net/tacticos/UML/UML04/UML0401.html>.
- Lenguajes de Ciencia de la Computación. Universidad de Málaga. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.lcc.uma.es/~jignacio/index\\_archivos/TEMA4.pdf](http://www.lcc.uma.es/~jignacio/index_archivos/TEMA4.pdf).
- **Zapata, Carlos Mario y Garcés, Gilma Liliana.** Revista EIA. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.scielo.unal.edu.co/scielo.php?pid=S1794-12372008000200008&script=sci\\_arttext](http://www.scielo.unal.edu.co/scielo.php?pid=S1794-12372008000200008&script=sci_arttext).
- Masadelante. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.masadelante.com/faqs/tipos-de-servidores>.
- MiSabueso. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.misabueso.com/directorio/Regional/Am%C3%A9rica/Cuba/Salud/Asociaciones/>.
- mitecnologico. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.mitecnologico.com/Main/ComponentesDeAplicacionesClienteServidor>.
- mitecnologico. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.mitecnologico.com/Main/ProcesosDeLaIngenieriaDeRequerimientos>.
- Mitecnologico. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.mitecnologico.com/Main/DefinicionModeloDeDatos>.
- Paratupagina. [En línea] [Citado el: 25 de 10 de 2010.] <http://paratupagina.com/topic/415-que-es-drupal/>.
- **Perezleo Solórzano, Ligeya , y otros.** Biblioteca Virtual en Salud. [En línea] [Citado el: 25 de 10 de 2010.] [http://bvs.sld.cu/revistas/aci/vol11\\_4\\_03/aci07403.htm](http://bvs.sld.cu/revistas/aci/vol11_4_03/aci07403.htm).

- postgresql. [En línea] [Citado el: 20 de 10 de 2010.] [http://www.guia-ubuntu.org/index.php?title=PgAdmin\\_III](http://www.guia-ubuntu.org/index.php?title=PgAdmin_III).
- RCIM. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.rcim.sld.cu/nota\\_a\\_los\\_autores.htm](http://www.rcim.sld.cu/nota_a_los_autores.htm).
- Risktechnology. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.risktechnology.net/metodologia-rup.aspx?skin=printerfriendly>.
- Sajú. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.google.com/url?sa=t&source=web&cd=8&ved=0CEEQFjAH&url=http%3A%2F%2Fwww.miportal.edu.sv%2Fportal%2Fcomunidad%2Fcfs-filesystemfile.ashx%2F\\_\\_key%2FCommunityServer.Components.PostAttachments%2F00.00.04.41.83%2Fbases-de-datos.docx&rct=j&q=El%20Gesto](http://www.google.com/url?sa=t&source=web&cd=8&ved=0CEEQFjAH&url=http%3A%2F%2Fwww.miportal.edu.sv%2Fportal%2Fcomunidad%2Fcfs-filesystemfile.ashx%2F__key%2FCommunityServer.Components.PostAttachments%2F00.00.04.41.83%2Fbases-de-datos.docx&rct=j&q=El%20Gesto).
- scribd. [En línea] [Citado el: 25 de 10 de 2010.] <http://es.scribd.com/doc/44942492/Presentacion-Guia-RUP>.
- SIEGFRIED RHEIN. [En línea] [Citado el: 25 de 10 de 2010.] [http://www.siegfried.com.mx/index.php?option=com\\_content&view=article&id=23&Itemid=6&lang=es](http://www.siegfried.com.mx/index.php?option=com_content&view=article&id=23&Itemid=6&lang=es).
- Slideboom. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.slideboom.com/presentations/21683/Introduccion.PostgreSQL>.
- Slideshare. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.slideshare.net/alina carrion/html-exposicion>.
- SOCIM. [En línea] [Citado el: 20 de 10 de 2010.] <http://www.socim.sld.cu/index.htm>.
- Softonic. [En línea] [Citado el: 25 de 10 de 2010.] <http://netbeans-ide.softonic.com/>.
- **Tortosa del Toro, Enrique.** [En línea] [Citado el: 25 de 10 de 2010.] [http://enriquetortosa.com/descargas/guia\\_diseno\\_web\\_basico.pdf](http://enriquetortosa.com/descargas/guia_diseno_web_basico.pdf).
- Unican. [En línea] [Citado el: 25 de 10 de 2010.] <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-i/materiales-de-clase-1/is1-t02-trans.pdf>.

- Upcommons. [En línea] [Citado el: 25 de 10 de 2010.] <http://upcommons.upc.edu/pfc/bitstream/2099.1/9749/1/65350.pdf>.
- Vas libre. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.vaslibre.org.ve/publicaciones/phpflisol2006.pdf>.
- Visión General de la Ingeniería de Software. [En línea] [Citado el: 25 de 10 de 2010.] <http://barranquillo.ucaldas.edu.co/rgarcia/ingsoft/visiongral.htm>.
- Xuletas. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.xuletas.es/ficha/adfadsf/>.

## ANEXOS

## Anexo 1. Fases y flujos de trabajos de la Metodología RUP.

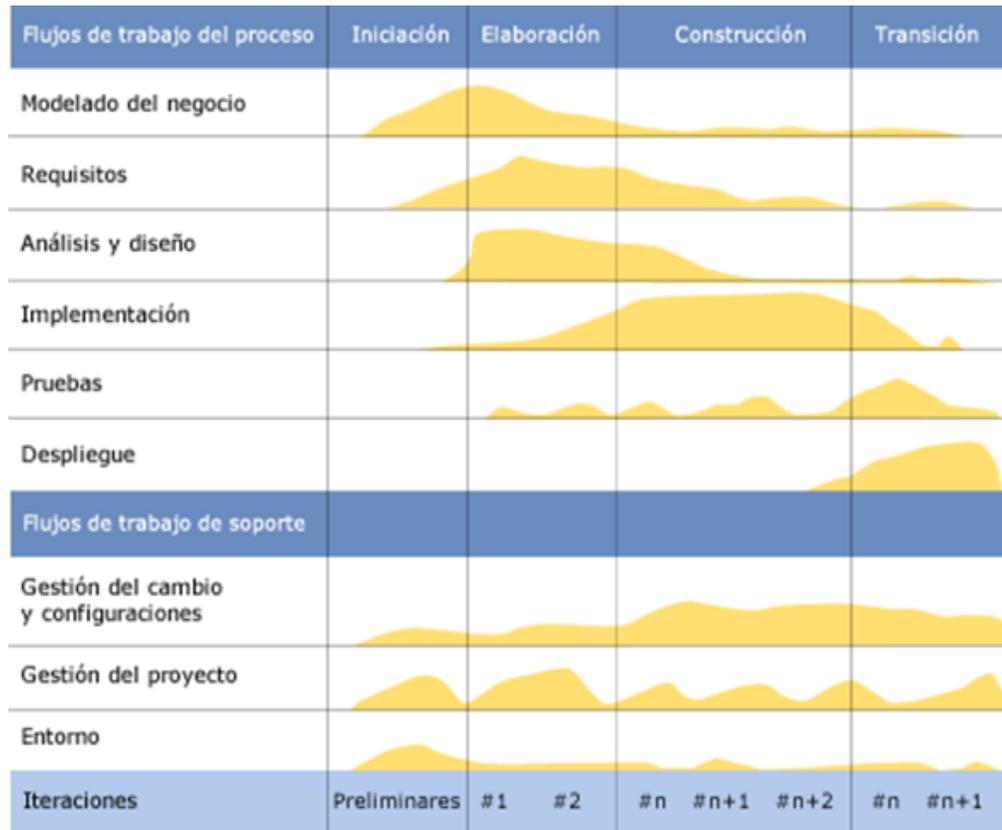


Fig. 28. Fases y flujos de trabajos de la Metodología RUP.

## Anexo 2. Arquitectura: Cliente-Servidor.



Fig. 29. Arquitectura: Cliente-Servidor.

### Anexo 3. Descripción de la arquitectura.

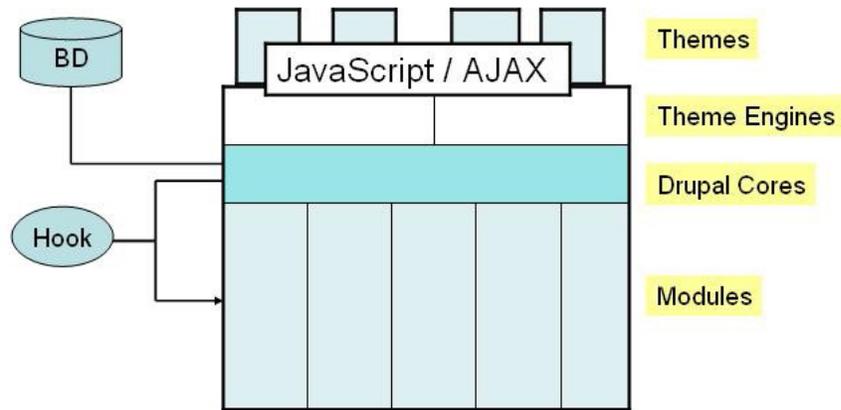


Fig. 30. Implementación del MVC según el CMS Drupal.