

Universidad de las Ciencias Informáticas
Facultad 6



**Título: “Diseño e implementación del Sistema de
Ayuda Médica para la
Atención de las Dislipoproteinemias versión 2.0.”**

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

Autor(es): Dayana Canova Ramírez
Saidel Pérez Dopaso

Tutor (es): Ing. Ileana Martí Pérez.
Ing. Liusmila Nieto Cervantes.

Co-tutor: Dr. Alfredo Nasiff Hadad

Ciudad de La Habana, 2008-2009.

"Los análisis de laboratorio, extienden, pero nunca reemplazarán las habilidades clínicas; lo que hace la tecnología cuando más moderna, es sustituir la tecnología vieja pero nunca al médico"

(M.A. Moreno).

DATOS DE CONTACTOS

Ing. Liusmila Nieto Cervantes (email: lnieto@uci.cu).

Profesora Ingeniera en Ciencias Informáticas. En el curso 2007-2008, fue Tribunal de Tesis y Tutora. Ha impartido la asignatura de Ingeniería de Software I y II. Posee categoría docente de Instructor Recién Graduado. Actualmente se desempeña como profesora de Herramienta CASE del Segundo Perfil de la Facultad, en la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI).

Ing. Ileana Marti Pérez (email: imarti@uci.cu).

Profesora Ingeniera en Ciencias Informáticas. En el curso 2007-2008, fue Oponente y Tutora. Ha impartido la asignatura de Teleinformática. Posee categoría docente de Instructor Recién Graduado. Actualmente se desempeña como profesora de Teleinformática Jefa del Departamento del Polo de Gestión de Información Biomédica, en la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI).

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo de Diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del Autor

Firma del Autor

Firma del Tutor

Firma del Tutor

Han sido muchas las personas que a lo largo de mi vida me han brindado amistad, cariño y apoyo incondicional. Por eso no quisiera pasar por alto este momento para hacerles llegar mi más profundo agradecimiento.

A mi mamá, a mi novio y a mi hermanita.

A mi padrastro por soportar tanta malcriadez y haberme ayudado a superar los obstáculos que la vida pone por delante.

A mis amigos del alma Daykenis, Alcalá, Marisleydis, Javier, Dennis, Amaury. Por ser tan especiales y haber estado para ahí cada vez que los he necesitado, por darme su apoyo y su amistad.

A mis amigos Chaviano y Adrián. Por haberme ayudado tanto durante estos 5 años, espero que nuestra amistad no termine aunque el destino nos separe por cuestiones de geografía.

A mis tutoras Llu y Lili gracias por haber soportado todas mis peleas, mis protestas y mis malcriadeces y haberme ayudado tanto como lo hicieron. Les agradezco de corazón todo el apoyo y todo el empeño que de su parte pusieron ya que sin ustedes esta tesis no hubiera sido nunca lo que hoy es.

A Yasel y Alejandro por todo el apoyo y la ayuda que nos han brindado.

A mis compañeros de aula desde que empezamos en primer año a los que se nos unieron más tarde y a aquellos que por un motivo u otro no están entre nosotros para cumplir con tal añorado sueño (Leniuska y Rubiney). A mis amigos David por imprimirme tantas conferencias, a los chicos del 41105 a la Maye, a mis vecinos Lety, Mileidy, Yuniesky, a todos aquellos que de una forma u otra ayudaron a que este día llegara y les pudiera agradecer desde lo más profundo de mi corazón.

A todos los profesores que han contribuido con nuestra formación profesional.

Dayana Canova Ramírez.

A mi madre por su comprensión y confianza, por su ejemplo de superación constante, por su amor y amistad, por su fuerza ante la vida.

A mis hermanos por siempre estar ahí.

A mi abuela por el cariño y los conocimientos que me transmitió durante la niñez y por ser ejemplo de incondicionalidad.

A mi padre que aunque no este hoy con nosotros siempre lo he tenido presente en cada paso de mi vida y sin querer tal vez me ha hecho ser mejor.

A mis tíos y primos por brindarme todo su y tenerme siempre presente.

A todos mis sobrinos por ser las personitas mas lindas de este mundo.

A Yamil por ser mucho mas que mi cuñado y siempre dar todo de si.

A todos mis amigos de la infancia por pasar toda su vida junto a mí y nunca defraudarme.

A mis amigos Rafa, José, el chino, Janier, Dismey, la gente del 71106 y 71105, por todo su apoyo y por hacer de estos años en la universidad los más inolvidables de mí vida.

A mi novia Yaima por ser de esa clase de personas que todo lo comprenden y dan lo mejor de sí sin esperar nada a cambio.

A mis tutoras que sin ellas tal vez mi tesis no fuera lo que hoy es.

Saidel Pérez Dopaso

De: Dayana Canova Ramírez

Quiero dedicar este trabajo a las tres personas que más amo:

A mi “MAMA” por ser el mejor padre y la mejor madre del mundo al mismo tiempo.

Duro ha sido el camino que juntas hemos transitado, pero juntas hemos sabido llegar al final. Gracias por enseñarme a no ceder nunca ante nada ni nadie, gracias por tus consejos, tu interminable comprensión, paciencia y amor. Espero no defraudarte nunca y poder demostrarte siempre que eres lo más grande que tengo en este mundo y que te estaré eternamente agradecida por haberme dado la vida y haberme convertido en la mujer que soy.

A mi “Pity” por ser la personita más chiquitica que tengo en el mundo.

Se que ha sido muy poco el tiempo que hemos pasado juntas desde que abristes tus ojitos por primera vez y te vi envuelta en aquel trapito verde con tus pelitos negros y esa naricita llena de pinticas blancas, pero espero poder sembrar en ti la semilla que nuestra madre supo sembrar en mi, ser tu ejemplo y poder darte el mismo amor, la misma comprensión y los mismos consejos que una vez me fueron dados. Quiero ayudarte a encontrar el final del camino que hace algún tiempo has comenzado a transitar para que algún día puedas escribir tu propia dedicatoria.

A mi “Machupichu” Alexander por ser lo mejor que me ha pasado en la vida.

Gracias por ser mi apoyo durante todo este tiempo y ayudarme en todo lo que has podido si hoy he llegado hasta aquí en gran parte es gracias a ti, a la paciencia, a la comprensión, a la confianza y a todo ese amor que de manera incondicional me has dado durante estos casi 4 años. Gracias por estar siempre a mi lado y haberme dado fuerzas para salir adelante en los buenos y en los malos momentos. Gracias por ser la persona que ha hecho posible que logre escribir mi dedicatoria.

De: Saidel Pérez Dopaso

A mi madre por su comprensión y confianza, por su ejemplo de superación constante, por su amor y amistad, por su fuerza ante la vida.

A mis hermanos por siempre estar ahí.

A mi padre que aunque no este hoy con nosotros siempre lo he tenido presente en cada paso de mi vida y sin querer tal vez me ha hecho ser mejor.

Resumen

Las dislipoproteinemias son enfermedades de difícil diagnóstico. Los médicos se enfrentan a una gran variedad de situaciones que conllevan a soluciones distintas para cada paciente, lo que hace necesario una evaluación rigurosa teniendo en cuenta sus factores de riesgo, grupo de riesgo al que pertenece, causas secundarias, antecedentes patológicos personales y familiares.

Hoy en día los profesionales de la salud no cuentan con un sistema que los apoye en la toma de decisiones a la hora de emitir diagnósticos y tratamientos adecuados a las características individuales de los pacientes afectados por estas enfermedades y que integre además funcionalidades que ayuden a prevenirlas. Por tales motivos se hace necesario el desarrollo del sistema informático de ayuda médica para la atención a las dislipoproteinemias v2.0.

El sistema propuesto estará disponible vía web y dispone de los mecanismos de seguridad a través del SAAA, servicio que brinda el sistema de salud basado en el modelo de Autenticación, Autorización y Auditoría (Authentication, Authorization and Accounting, AAA). Se utilizaron en su gran mayoría herramientas de software libre teniendo en cuenta las políticas del país para la soberanía tecnológica.

Palabras claves: atención, dislipoproteinemias, diagnósticos, integre, sistema informático, tratamientos.



Índice

INTRODUCCIÓN	1
Capítulo 1: Fundamentación Teórica.	6
1.1. GESTIÓN DE LA INFORMACIÓN.....	6
1.2. SISTEMAS DE GESTIÓN DE INFORMACIÓN.....	7
1.3. LA WEB COMO SISTEMA DE GESTIÓN DE INFORMACIÓN.....	8
1.4. APLICACIONES DE SOFTWARE MÉDICOS EN EL MUNDO.....	10
1.5. APLICACIONES Y SOFTWARE MÉDICOS EN CUBA.....	12
1.6. ¿POR QUÉ DESARROLLAR EL DISEÑO E IMPLEMENTACIÓN DE ALASLIPO v2.0?	15
1.7. ARQUITECTURA DE SOFTWARE EN EL SISTEMA NACIONAL DE SALUD (SNS).....	16
1.8. ANÁLISIS DE LAS HERRAMIENTAS SELECCIONADAS.....	18
1.9. PATRONES DE DISEÑO.....	30
1.10. PATRÓN DE ARQUITECTURA MODELO VISTA CONTROLADOR (MVC).....	34
Capítulo 2: Diseño del Sistema.....	36
2.1. CARACTERÍSTICAS DEL SISTEMA.....	36
2.2. ARQUITECTURA Y DISEÑO.....	51
2.3. PAUTAS DEL DISEÑO.....	60
2.4. VISTAS ARQUITECTÓNICAS.....	65
2.5. VISTA LÓGICA.....	66
2.6. VISTA DE DESPLIEGUE.....	87
Capítulo 3: Implementación del sistema.....	90
3.1. VISTA DE IMPLEMENTACIÓN.....	90
3.2. DIAGRAMA DE COMPONENTES.....	92
3.3. CÓDIGO FUENTE.....	93
3.4. INTEGRACIÓN CON AJAX.....	102
3.5. MAPA DE NAVEGACIÓN.....	104
3.6. INTERFACES PRINCIPALES.....	105
3.7. VALIDACIÓN DE LA APLICACIÓN.....	114
Conclusiones.....	118
Recomendaciones.....	119
Referencias bibliográficas.....	120
Bibliografía.....	124
Anexos	127
Glosario de términos.....	¡Error! Marcador no definido.



INTRODUCCIÓN

La medicina es la ciencia dedicada al estudio de la vida, la salud, las enfermedades y la muerte del ser humano, e implica el arte de ejercer tal conocimiento técnico para el mantenimiento y recuperación de la salud, aplicándolo al diagnóstico, tratamiento y prevención de las enfermedades. Junto al desarrollo de la medicina y otras ciencias, el hombre ha trabajado en cómo manejar el cúmulo de conocimientos obtenidos, es decir “la información”.

Con la necesidad del procesamiento de las informaciones obtenidas surge como una nueva rama de la ciencia, la informática, disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales. Una de las aplicaciones más importantes de la informática es facilitar información en forma oportuna y veraz, lo cual puede tanto facilitar la toma de decisiones como permitir el control de procesos críticos. Actualmente es difícil concebir un área que no use, de alguna forma, el apoyo de la informática. La misma es un enorme abanico que cubre desde las más simples cuestiones domésticas hasta los cálculos científicos más complejos.

La unión entre estas dos ramas de la ciencia dieron lugar al surgimiento de la Bioinformática, esta disciplina proporciona herramientas y recursos para favorecer la investigación biomédica y trata de desarrollar sistemas que sirvan para entender el flujo de información desde los genes a las estructuras moleculares, su función bioquímica, conducta biológica y finalmente, su influencia en las enfermedades y la salud.

Los avances obtenidos en la ciencia y la tecnología han permitido conseguir un notable progreso de la medicina, esto ha repercutido en el bienestar de la humanidad permitiendo estilos de vidas más saludables y a prolongar la expectativa de vida. En la actualidad se asiste a una menor morbimortalidad de tal modo que las personas pueden llegar a una mayor edad y pueden sufrir enfermedades ya no tanto infecciosas sino más bien de carácter crónico como la diabetes mellitus, obesidad, hipertensión arterial y las dislipoproteinemias.



Las dislipoproteinemias son enfermedades que se caracterizan por el “aumento, disminución (en este último caso, especialmente (HDLc) y/o cambios en la estructura y composición de una o más lipoproteínas en sangre” [1].

En relación a las dislipoproteinemias, se han efectuado aportes científicos muy importantes; se han establecido los niveles de normalidad y cuales son las concentraciones de lípidos sanguíneos que ofrecen riesgo a largo plazo sobre la salud y la vida de las personas, se ha demostrado que las intervenciones terapéuticas en las personas con riesgo de sufrir enfermedades cardiovasculares, como consecuencia de un incremento de las lipoproteínas sanguíneas, han conseguido disminuir la aparición de los eventos cardiovasculares y la reducción de la mortalidad por tales eventos, lo que posibilitará no solo la prevención si no que también la regresión de la arterosclerosis.

Las dislipoproteinemias son enfermedades de difícil diagnóstico cuando se trata de diferenciar entre formas primarias y secundarias. Los profesionales de la salud se enfrentan a una gran variedad de situaciones que conllevan a soluciones distintas para cada paciente, lo que hace necesario una evaluación rigurosa teniendo en cuenta sus factores de riesgo, grupo de riesgo al que pertenece, causas secundarias, antecedentes patológicos personales y familiares para la indicación del diagnóstico y tratamiento adecuado a las características individuales de cada paciente.

A nivel mundial existen muchos softwares que ayudan a las personas a prevenir el padecimiento de enfermedades cardíacas y cerebrovasculares, aunque la mayoría no brindan la información necesaria y de forma organizada acerca de la enfermedad.

Cuba a pesar de ser un país bloqueado, en su afán de mantener los logros obtenidos en la esfera de la salud, de mejorar la calidad de vida de las personas, así como de brindar herramientas de apoyo a los médicos para la toma de decisiones, destina incalculables recursos.

Como antecedentes de este trabajo se tiene conocimiento de que en el país se desarrollaron anteriormente sistemas como LIPICID, creado en 1996, que permitía diagnosticar y tratar a todo paciente dislipoproteinémico, así como promover un estilo de vida más saludable y sugerir medidas para actuar sobre los factores modificables, pero que en la práctica resultó ser poco amigable, de difícil



manejo y rechazado por médicos y pacientes; el otro sistema que le sucedió fue el Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias versión 1.0 (SAMAD), el cual fue concebido con objetivos similares pero tratando de resolver los problemas que presentó el anterior sistema. Esta aplicación Web no quedó totalmente concluida pues no se terminó el ciclo de consultas que se le debe realizar a un paciente que padezca esta enfermedad y además no poseía una arquitectura que permitiera que la aplicación fuera desplegada en la red de Infomed para que los médicos y pacientes tuvieran acceso.

A raíz de este problema y con la necesidad de desplegar el sistema en los servidores de Infomed durante los últimos meses del año 2007 y mitad del año 2008 se trabajó sobre una nueva versión de SAMAD la cual estaría sustentada en una arquitectura que fuera capaz de acoplarse a las exigencias del Sistema Nacional de Salud (SNS) para el despliegue de aplicaciones en la red de Infomed. La misma se basó en el refinamiento de los requerimientos existentes y se desarrollaron los nuevos modelos de análisis y diseño para la nueva versión.

Esta nueva versión no completó el ciclo del desarrollo del software. De ahí que se defina entonces como **problema científico** del presente trabajo ¿cómo diseñar e implementar el Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias versión 2.0?

En función del problema científico identificado se determina como **objeto de estudio** en esta investigación el Proceso de desarrollo de software para los sistemas de gestión de información y como **campo de acción** el Proceso de desarrollo de software para la implementación del Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias versión 2.0.

Para dar solución a lo anteriormente expuesto se precisa como **objetivo general** del presente trabajo:

- Realizar el diseño e implementación del Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias versión 2.0.

Para cumplir con el objetivo propuesto se definen los siguientes **objetivos específicos** de la investigación:

- Diseñar el sistema.



- Implementar las clases del diseño.

Para lograr el cumplimiento de los objetivos específicos señalados se definen las siguientes **tareas de investigación**:

- Estudio del diseño propuesto.
- Obtención de un nuevo diagrama de clases del diseño.
- Obtención de un nuevo modelo de datos.
- Obtención del diagrama de componentes.
- Implementación de las clases definidas en el diseño.
- Validación del sistema a nivel de desarrollador.

Al finalizar el presente trabajo los involucrados en este proceso esperamos obtener como **posible resultado**:

- El Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias versión 2.0.

El presente trabajo consta de Introducción, Conclusiones, Bibliografía, Anexos y 3 Capítulos que abordan los temas fundamentales distribuidos de la siguiente manera:

Capítulo 1: Fundamentación Teórica.

En el **Capítulo 1** se brinda una breve panorámica sobre los sistemas de gestión de información, los proyectos y aplicaciones web que presentan relación con el presente trabajo a nivel mundial y nacional. Se fundamenta la utilización de la metodología, las herramientas y los tipos de patrones que se utilizarán para el desarrollo de la aplicación. Se hace un análisis crítico en la propuesta de por qué realizar el diseño e implementación de alasLIPO v2.0. Se aborda el tema de la arquitectura establecida por Softel. Se exponen los roles que se desempeñarán y los artefactos que se generarán durante los flujos de trabajo de Análisis y diseño e implementación.

Capítulo 2: Diseño del Sistema.



En el **Capítulo 2** se exponen elementos importantes para una solución satisfactoria tales como los requisitos funcionales y no funcionales, los actores del sistema así como una descripción resumida de los casos de uso del sistema. Además se realiza el diseño del sistema obteniéndose los diagramas de clases del diseño web, interacción (secuencia), despliegue, clases persistentes y el modelo de datos.

Capítulo 3: Implementación del Sistema.

En el **Capítulo 3** se presentan los diagramas de componentes elaborados a partir de los diagramas de clases de diseño. Se muestra el código fuente de uno de los principales métodos con la descripción del mismo. Se argumenta sobre la validación y seguridad aplicada a la aplicación. Además se presentan las interfaces de la aplicación más significativas de alasLIPO v2.0.

Capítulo 1: Fundamentación Teórica.

Introducción.

En el presente capítulo se brinda una breve panorámica sobre los sistemas de gestión de información, los proyectos y aplicaciones web que presentan relación con el presente trabajo a nivel mundial y nacional. Se fundamenta la utilización de la metodología, las herramientas y los tipos de patrones que se utilizarán para el desarrollo de la aplicación. Se hace un análisis crítico en la propuesta de por qué realizar el diseño e implementación de alasLIPO v2.0. Se aborda el tema de la arquitectura establecida por Softel. Se exponen los roles que se desempeñarán y los artefactos que se generarán durante los flujos de trabajo de análisis y diseño e implementación.

1.1. Gestión de la información.

En la era de la información, de la explosión de las tecnologías, se vive la etapa en la que la humanidad ha alcanzado un desarrollo imprevisible. Se habla constantemente sobre la sociedad de la información, es visible el paso de las sociedades industriales a las postindustriales y del conocimiento, donde el factor esencial de progreso es el conocimiento. Esta nueva sociedad, se sustenta en un desarrollo tecnológico sin precedentes, es el punto en el cual se planifican los productos en función de la gestión de la información, el conocimiento y de la viabilidad para su obtención.

La información es un factor primordial para el desarrollo, con el decursar de los años, la gestión de la información ocupa, cada vez más, un espacio mayor en la economía de los países a escala mundial.

La “**Gestión de información**”: Conjunto de instancias responsables por la identificación de políticas y acciones en la organización con relación a lo siguiente:

- La forma en cómo la información se adquiere, registra y guarda.
- El cómo la información se usa y acumula.
- La manera como las personas manejan la información, aplican sus habilidades y cooperan entre ellas.
- El cómo la información se usa y acumula.
- La manera como las personas manejan la información, aplican sus habilidades y cooperan entre ellas.

- La efectividad con que las actividades relacionadas con la información contribuyen al logro de los objetivos de las organizaciones y los individuos.
- La forma como se usan las tecnologías de la información en todas estas actividades.
- Los costos y beneficios que conllevan las actividades de información.”[2].

La gestión de la información se vincula con la concepción, la puesta en práctica de estrategias, y el establecimiento de políticas para el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas trazadas en materia de desempeño y calidad para mejorar la toma de decisiones en todos los niveles.

1.2. Sistemas de Gestión de Información.

Gracias al gran desarrollo alcanzado por los Sistemas Informáticos hoy día la mayoría de las empresas cuentan con potentes herramientas informáticas para facilitar la gestión de la información en todas las áreas de trabajo.

“Un **sistema de gestión información** se puede definir como un conjunto de funciones o componentes interrelacionados que forman un todo, es decir, obtiene, procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización. Igualmente apoya la coordinación, análisis de problemas, visualización de aspectos complejos entre otros.

El sistema de gestión de información en una empresa: es el conjunto de procesos que, operando sobre una colección de datos, recopila, elabora y distribuye (parte de) la información necesaria para la información de dicha empresa. Entre estos componentes los más importantes son:

- El equipo computacional: el hardware necesario para que el sistema de información pueda operar.
- El recurso humano que interactúa con el sistema de información, el cual está formado por las personas que utilizan el sistema.
- El sistema de información en sí (generalmente una aplicación software).

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.” [3].

Los sistemas de gestión de Información constituyen hoy, una alternativa de imprescindible presencia en cada organización. Han llegado a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización.

1.3. La web como Sistema de Gestión de Información.

Desde el surgimiento de la Internet, muchas organizaciones se dispusieron a darse a conocer al mundo utilizando esta tecnología. Las empresas se dedicaron a realizar sitios web en donde daban publicidad a sus productos o a los servicios que estas brindaban. Esto trajo consigo un gran incremento en las ventas ya que el uso de la red de redes permite que la información llegue hasta los más recónditos lugares del mundo.

Fue así como se comenzaron a desarrollar lenguajes dinámicos de programación para la web, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos. A continuación se describen algunos de los más utilizados:

HTML: Lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de Hypertext Markup Language, (en español Lenguaje de Marcas de Hipertexto). Desarrollado por el World Wide Web Consortium (W3C).

Javascript: Lenguaje de programación interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications.

PHP: Lenguaje de programación utilizado para la creación de sitio web. Php es un acrónimo que significa “PHP hypertext pre-processor”, desarrollado por Php Group.es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor.

ASP: Tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitio web dinámicos. Asp significa en inglés (active server pages), fue liberado por Microsoft en 1996.

ASP.NET: Lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.net es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .net.

JSP: Lenguaje para la creación de sitios web dinámicos, acrónimo de java server pages. Está orientado a desarrollar páginas web en java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor.jsp fue desarrollado por Sun Microsystems. Para su funcionamiento se necesita tener instalado un servidor tomcat.

Python: Creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

Ruby: Lenguaje interpretado de muy alto nivel y orientado a objetos. Desarrollado en el 1993 por el programador japonés Yukihiro “Matz” Matsumoto. Su sintaxis está inspirada en Python, Perl. Es distribuido bajo licencia de software libre (OpenSource).

Coldfusion: Lenguaje 100% de scripting en que la idea es lograr resultados con pocas líneas de código. Explota el hecho de que en la práctica muchos programadores lo que hacen es simplemente programar y no quieren tener que ver con Objetos, Clases ni nada de alto vuelo sino sólo con hacer consultas a la base de datos e imprimir el resultado.

“Los Sitios Web, como expresión de sistemas de información, poseen los siguientes componentes:

- Usuarios.
- Mecanismos de entrada y salida de la información.
- Almacenes de datos, información y conocimiento.
- Mecanismos de recuperación de información.

Actualmente, los sistemas de información se encuentran al alcance de las grandes masas de usuarios por medio de Internet; así se crean las bases de un nuevo modelo, en el que los usuarios interactúan directamente con dichos sistemas para satisfacer sus necesidades de información.” [4].

1.4. Aplicaciones de software médicos en el mundo.

La informática médica provee al mundo métodos novedosos, sencillos, robustos y de gran eficiencia para la gestión de información que se obtienen de consultas médicas en centros de salud e investigación médica. Para la mayoría de los médicos la disposición de complejos software es considerada una herramienta indispensable en la actualidad. Estos software son utilizados como medios de apoyo reduciendo así la posibilidad de errores en el diagnóstico y tratamiento de las enfermedades trayendo consigo que la formación del profesional sea cada día mejor y que pueda intercambiar información, y conocimientos con el resto del mundo.

El desarrollo de sistemas de ayuda al diagnóstico avanzado usando inteligencia artificial, el desarrollo de sitios Web interactivos con fines docentes y clínicos, el empleo de herramientas gráficas para el tratamiento de imágenes con procesamiento de información de interés médico, la automatización de historias clínicas, el procesamiento de señales, los sistemas simuladores, y las calculadoras médicas constituyen software que ayudan a prevenir el padecimiento de enfermedades cardíacas y cerebrovasculares.

La mayoría de estos software se limitan a recomendar consejos sobre ejercicios físicos que se pueden realizar, sobre tipos de dietas balanceadas que debe seguir una persona de acuerdo a sus características físicas y su edad, a brindar información acerca de la enfermedad, a ofrecer la posibilidad de realizar algunos cálculos como el factor de riesgo de la enfermedad o calcular el valor del peso ideal. A continuación se relacionan algunos ejemplos de aplicaciones a nivel mundial que han mostrado resultados y que tienen el objetivo de prevenir y tratar las dislipoproteinemias al igual que **alasLIPO v2.0**.

- Calculadoras para el cálculo del Índice de Masa Corporal (IMC).
- GRACE Risk Model (2006).
- Software Obstétrico para Asiente Personal Digital (PDA).
- HOMA calculator (2004).
- UKPDS RISK (2004).
- Sistema de Aproximación al Diagnóstico (SAD).
- BrainLED AlfaWave v1.2.

Citmed 6.0:

Es un software gratuito concebido para la gestión de una consulta médica. Es un producto que se adapta con gran facilidad a los diferentes entornos clínicos. Entre sus funcionalidades principales se encuentra: Agenda de citas. Gestión de fichas e historias clínicas de pacientes. Recetas, petición de analíticas, pruebas complementarias. Proporciona flexibilidad con sus modos de funcionamiento multi-consulta y multi-especialidad. Está diseñado para trabajar en red local, incorpora un sistema de seguridad adecuado a los requisitos para la protección de la privacidad de los datos de los pacientes.



Multicalculadora RCV:

Esta calculadora de riesgo cardiovascular permite evaluar las posibilidades de riesgo que presenta el paciente haciendo uso de los modelos SCORE, Framingham, Regicor y Dórica. Con solo añadir datos tales como edad, sexo, si es fumador, si es diabético, colesterol, presión, etc., y la calculadora hará una estimación de riesgo. Software distribuido bajo licencia **Freeware**.



DiagnosMD (2009):

Primer programa de Inteligencia Artificial en español aplicado a toda la Medicina. Es una potente herramienta propietaria de utilidad continua en la consulta, que ayuda al diagnóstico al combinar un conjunto de datos (síntomas, signos, resultados analíticos anormales, etc.) con el país, sexo y edad del paciente, ofreciendo con criterio un listado de enfermedades posibles, con potentes herramientas para afinar en el diagnóstico. Ayudar a evitar errores en la medicación (contraindicaciones, interacciones entre fármacos, doble terapia, etc.). Permite consultar rápidamente enfermedades, dietas para enfermedades, imágenes, protocolos en Atención Primaria y en Urgencias (con los criterios de ingreso hospitalario), pruebas de laboratorio, fármacos, interacciones, contraindicaciones y plantas medicinales.



GlucoGestor.

Aplicación que ayuda a registrar las medidas de glucosa en sangre que se haya tomado un paciente diabético a lo largo de su vida. Los registros se podrán visualizar tanto en formato gráfico como impreso. Se podrá visualizar un gráfico en el que se indicarán las glucemias que se han registrado en un período de tiempo configurable por el usuario y en los tramos horarios que este elija. Software distribuido bajo licencia **Freeware**.

1.5. Aplicaciones y software médicos en Cuba.

Cuba no está exenta de la aplicación de la informática médica, tanto es así que se han creado numerosos centros de investigación como el Instituto Central de Investigaciones Digitales (ICID), el Centro de Estudios Genéticos y Biotecnología (CIGB), el Centro de Neurociencias de Cuba (CNC), el Centro de Cibernética Aplicada a la Medicina, el Centro de Desarrollo Informático para la salud pública, el Centro Nacional de Bioinformática (Bioinfo), centros que han decidido incorporar aplicaciones de gestión de información para mejorar sus decisiones, y su efectividad logrando así una mayor dinámica, calidad y eficiencia.

Estos centros dedican parte de su trabajo a crear software que sirven de apoyo para la detección, diagnóstico y tratamiento de enfermedades, herramientas para el aprendizaje de la medicina e importantes bases de datos médicas con grandes cantidades de información gestionada.

Otro de los principales aspectos significativos en la aplicación de la informática médica es el desarrollo de la Red de Información de las Ciencias Médicas (INFOMED), la cual posibilita el intercambio entre gran cantidad de centros de salud e investigación de Cuba con el resto del mundo. INFOMED brinda un número importante de revistas, libros, publicaciones y materiales científicos de gran utilidad para sus usuarios, y enlaces a sitios Web nacionales y significativas instituciones internacionales.

Cuba es miembro importante en la Asociación Internacional de Informática Médica (IMIA), ha sido sede de congresos internacionales de informática médica. Como resultado de la aplicación de la Informática Médica en Cuba se ha obtenido como resultado muchos software médicos, como son:

- CoFar.
- Excorde 3C.
- Sistema para el diagnóstico remoto de ECG.
- NeuroCid.
- CardioCid
- Software de registro y almacenamiento para el estudio del sueño.
- Medicid 5-132 (Polígrafo digital).
- CLÍNICO 3.0.
- Imagis (Sistema de transmisión de imágenes médicas digitales).



Lipicid.

LIPICID es una aplicación Windows para el diagnóstico, evaluación y tratamiento de las dislipoproteinemias. Toda la información sobre los pacientes manipulada por LIPICID se almacena en bases de datos tipo FoxPro donde una aplicación DOS denominada LIPINFO analiza dicha información. LIPICID detecta la dislipidemia a través del lipidograma, almacena información relacionada a los antecedentes personales y familiares, refleja los elementos positivos al examen físico tales como xantomas, arco lipoideo, índice de masa corporal, presión arterial; recomienda los exámenes complementarios para el despistaje de causas secundarias de dislipidemias y determina la existencia de aterosclerosis preclínica recomendando finalmente el tratamiento no farmacológico y en una etapa posterior el tratamiento farmacológico. Es un sistema basado en la experiencia de la Clínica de Lípidos del Hospital "Hermanos Ameijeiras", que utiliza, entre otros, los criterios del Segundo Panel de Expertos del NCEP de EE.UU.



HyperWeb.

Aplicación concebida con el objetivo de mejorar las funcionalidades presentadas por Hipertencid. Es un sistema distribuido, basado en una plataforma Web y en técnicas de Inteligencia Artificial (IA), flexible, escalable y extensible. Brinda la posibilidad de acceso simultáneo de usuarios restringido por roles, brinda servicios a nivel de intranet (hospitales) o Internet (misiones médicas cubanas) y recoge el conocimiento más actualizado en el tema de la hipertensión arterial (HTA). Está dirigido a médicos con formación general y especialistas en HTA para generar diagnósticos, tratamientos, realizar un trabajo profiláctico en la mejora del estilo de vida y prevenir la HTA. Es un sistema de apoyo para la toma de decisiones de los médicos que no desecha ninguna de las posibilidades dejando en manos del médico la decisión final. Además brinda toda la información relevante y necesaria para apoyar esta decisión.

La adquisición de las herramientas como las anteriormente mencionadas se hace casi inverosímil ya que la mayoría de estas son productos propietarios de realzados precios en el mercado, además requieren soporte por parte de las empresas que los desarrollan lo que se traduce en gastos a la economía cubana. Una vez estudiadas todas estas herramientas propietarias y libres se llegó a la conclusión de que no existe a nivel mundial ninguna que adhiera todas las funcionalidades requeridas para emitir diagnósticos, tratamientos y ayude a la prevención de las dislipoproteinemias.

El presente trabajo pretende desarrollar un software que integre todas las ventajas de los productos estudiados, dígase calcular las calorías de la dieta (peso ideal e IMC), determinar el riesgo cardiovascular para conocer la probabilidad que presenta el paciente de sufrir un evento cardíaco en un período de tiempo determinado, recomendar alimentaciones adecuadas para disminuir el colesterol y los triglicéridos en el plasma, proponer ejercicios físicos para prevenir la obesidad y mantener un cuerpo y una mente sana, mantener un registro actualizado de la Historia Clínica del paciente, graficar el comportamiento evolutivo de este, así como emitir un diagnóstico y un tratamiento farmacológico y no farmacológico.

1.6. ¿Por qué desarrollar el diseño e implementación de **alasLIPO v2.0**?

El diseño propuesto por el trabajo de diploma “Análisis y Diseño del Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias” sirvió de base para comprender en gran medida las clases necesarias para la implementación. Se realizó un análisis completo del negocio, requerimiento y diseños propuestos. Se realizaron consultas al cliente sobre determinadas funcionalidades que adolecían en ocasiones de un sentido lógico y descripción correcta. Como resultados de la evaluación del diseño se concluyó que el mismo no podía utilizarse para la implementación por varias razones:

- El instrumento que sirve como base del negocio poseía campos que no estaban incluidos en la Base de Datos y hubo que rediseñar la misma completamente.
- Se añadieron nuevos requisitos por el cliente y se redefinieron otros.
- Como consecuencia cambió el negocio y el sistema, lo cual incide directamente en el diseño.

El diseño e implementación de **alasLIPOv2.0** brindará a los médicos una herramienta de apoyo que permita:

- Emitir diagnósticos y tratamientos eficientes.
- Contribuir en la lucha contra las dislipoproteinemias
- Proponer mejores estilos de vida con ejemplos de dietas a seguir.
- Exhortar a los pacientes a realizar ejercicios físicos.
- Mayor agilidad en la gestión de la información relacionada a las dislipoproteinemias.
- Generar reportes a partir de la información almacenada, permitiendo realizar estudios comparativos de casos.
- Generar gráficas de comportamiento de diferentes parámetros.
- Generar posible diagnóstico asociado a las características de cada paciente.
- Generar tratamiento del paciente a partir del diagnósticos predicho.
- Posibilidad de darle seguimiento al paciente a partir de un diagnóstico y tratamiento evolutivo.

Para el desarrollo de la investigación los autores se han basado en los resultados del documento ATP III (para la entrada de datos al sistema y realización de algoritmos), entrevistas a médicos especialistas del Hospital Hermanos Ameijeiras, análisis del LIPICID desarrollado por el ICID e información médica obtenida de las bases de datos de Internet, libros de la especialidad y reutilización de algoritmos implementados en la aplicación anterior.

1.7. Arquitectura de Software en el Sistema Nacional de Salud (SNS).

El SNS define un grupo de requisitos no funcionales que deben ser cumplidos en su mayoría por las entidades involucradas en el desarrollo de productos informáticos, estos requisitos constituyen la base que debe sustentar la arquitectura de cualquier proyecto de informatización en el sector de la salud para responder a las políticas de informatización de este. A continuación se describe un conjunto de requisitos no funcionales que fueron seleccionados de los que el SNS define para todo proyecto de informatización en acuerdo con los compañeros de Infomed para que fueran cumplidos por el sistema alasLIPO v2.0.

Estos Requerimientos No Funcionales están avalados por más de 15 sistemas que integran el Registro Informatizado de Salud desplegado en un cluster de servidores en el nodo central de INFOMED, la reutilización de dicha arquitectura garantizará el aprovechamiento de los recursos tecnológicos con los que se cuenta.

RNF 1. Requerimientos de seguridad.

El sistema deberá contar con un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría (Authentication, Authorization and Accounting, AAA) con Autenticación de firma única (Single Sign On).” [5].

- “La autenticación consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica.” [5].
- Si el usuario autenticado no se encuentra registrado se debe reportar un error de acceso.
- “El sistema permitirá a las personas designadas como administradores establecer las políticas de seguridad en su nivel de acceso.

- Los administradores podrán crear, cuentas de administradores en los niveles inferiores.

RNF 4. Requerimiento de software.

RNF 4.1. Todo el software debe correr sobre el Sistema Operativo Linux distribución Red Hat (White Box). Actualmente se estabiliza una distribución Debian para garantizar la compatibilidad con nuevos paquetes no disponibles para Red Hat.

RNF 4.2. El software base debe responder a las políticas de Software Libre y de Fuente Abierta.

RNF 4.2.1. Servidor http Apache 2.0.

RNF 4.2.2. Lenguaje de programación del lado del servidor PHP versión 4.x o 5.0.

RNF 4.2.3. Servidor de datos MySQL 4.x o 5.x.

RNF 4.4. Los clientes tendrán acceso al sistema a través de cualquier navegador Web y debe ser operable tanto en **Mozilla** como **Internet Explorer**.

RNF 5. Requisito de Hardware.

RNF 5.2. Las Capas de Presentación, Negocio y Datos estarán desplegadas en dos servidores dedicados.

RNF 6. Requisito de diseño y la implementación.

RNF 6.1.1. La Capa de Presentación contendrá la interfaz de usuario y negocio de validación de formularios de entrada de datos o funcionalidades que garanticen la usabilidad del sistema.

RNF 6.1.2. La Capa de Negocio contendrá los servicios que dan cumplimiento a los Requisitos Funcionales del Sistema.

RNF 6.1.3. La Capa de Datos contiene la Base de Datos relacional del Sistema y negocio únicamente que garantice la integridad referencial de los datos.”[5].

La reutilización de esta arquitectura garantizará los beneficios de reusabilidad de servicios, interoperabilidad de aplicaciones, utilización de un medio único de acceso a servicios y el aprovechamiento de los recursos tecnológicos con los que se cuenta. Debido a estas características

de arquitectura descritas, para que pueda ser desplegado cualquier software desarrollado para el SNS en sus servidores centrales, se torna necesario cumplir con los requerimientos de MySQL y desplegarse sobre servidor Linux. Estos requisitos propuestos exigen a cualquier aplicación que se ejecute sobre esta red la utilización de Java Script como tecnología del lado del cliente.

1.8. Análisis de las herramientas seleccionadas.

1.8.1. Proceso de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software.

“RUP (Rational Unified Process) es un proceso de desarrollo de software que captura las mejores prácticas del conocimiento de líderes en ingeniería de software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad. Las mejores prácticas de desarrollo de software están documentadas como principios clave. “[6].

Para el desarrollo de alasLIPO v2.0 se decidió utilizar RUP por las siguientes razones:

- Está concebido para proyectos y equipos de trabajos grandes (Esta es la principal característica del proyecto en que se desarrolla el presente trabajo).
- Exige una documentación completa de los artefactos a realizar.
- Existe una formalidad prefijada.
- Es un proceso mucho más controlado, con numerosas políticas y normas.
- El proyecto está formado por grandes grupos y distribuidos por módulos.

RUP se caracteriza por ser un:

- **Proceso dirigido por Casos de Uso.**

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema, también guían su diseño, implementación y prueba, constituyen un elemento integrador y una guía del trabajo.

- **Proceso centrado en la arquitectura.**

La arquitectura de un sistema es la estructura u organización de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios). La arquitectura se ve influenciada por la plataforma de software, sistema operativo, gestor de bases de datos y protocolos. RUP presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

- **Proceso iterativo e incremental.**

RUP se propone como estrategia, tener un proceso iterativo e incremental donde el trabajo se divida en partes más pequeñas o mini proyectos. Permitiendo un equilibrio entre Casos de Uso y arquitectura durante cada mini proyecto, y así para todo el proceso de desarrollo.

RUP ayuda a mejorar la productividad del equipo de trabajo, definiendo claramente sus actividades, roles y responsabilidades. Propone un conjunto de mejores prácticas a todos los miembros del equipo de proyecto como son: desarrollo de software iterativo (releases y versionamiento), utilización de arquitectura basada en componentes (diseño de arquitectura flexible), modelamiento de requisitos y diseño de software (con la utilización de UML), y verificación de la calidad.

RUP divide el proceso de desarrollo de software en 4 fases concluyendo con un hito bien definido para cada una de estas:

- **Inicio:** Se determina la visión del proyecto.
- **Elaboración:** Se determina la arquitectura óptima.
- **Construcción:** Se obtiene la capacidad operacional inicial.
- **Transición:** Se obtiene la liberación del proyecto.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales, los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo.

1.8.2. Flujos de trabajo. Roles y Artefactos.

Acorde a la metodología RUP, el presente trabajo se va a realizar enmarcado en los flujos de trabajo de: **Análisis y Diseño** (Flujo del cual solo se realizó el diseño) e **Implementación**.

Diseño: El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales. El diseño debe ser suficiente para que el sistema pueda ser implementado sin imprecisiones. Contribuirá a obtener una arquitectura estable y sólida y a crear un plano del modelo de implementación capturando los requisitos o subsistemas individuales, interfaces y clases.

Los roles desempeñados en el desarrollo de alasLIPO v2.0:

- Diseñador.
- Diseñador de base de datos.
- Diseñador de interfaz de usuario.
- Arquitecto de software.

Los artefactos que se obtuvieron durante este flujo de trabajo:

- Clases del diseño.
- Paquetes del diseño.
- Subsistemas del diseño.
- Modelo de diseño.
- Modelo de despliegue
- Documento de arquitectura de software.
- Realización de los CU.
- Modelo de datos.
- Mapa de navegación.
- Prototipo de interfaz de usuario.

Implementación: En este flujo de trabajo se implementan las clases que se obtuvieron en la fase de análisis y diseño en ficheros fuente, binarios, ejecutables. Además se deben hacer los tests de unidad donde cada implementador es responsable de probar los componentes desarrollados e integrar estos en un sistema ejecutable. El resultado final de este flujo de trabajo es un sistema ejecutable.

Los roles desempeñados en el desarrollo de alasLIPO v2.0:

- Programador.
- Arquitecto de software.

Los artefactos que se obtuvieron durante este flujo de trabajo:

- Modelo implementación.
- Elementos de implementación.
- Subsistemas de implementación.

1.8.3. Lenguaje de Modelado Unificado.

El Proceso Unificado de Desarrollo (RUP) utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) como lenguaje de notación, lenguaje que se utilizará para el modelado del sistema alasLIPO v2.0.

“UML es un lenguaje estándar para especificar, visualizar, construir y documentar todos los artefactos de un sistema de software.”[7].

El objetivo del uso de UML es capturar las partes esenciales del sistema a desarrollar. Su uso permite combinar elementos gráficos mediante reglas para finalmente obtener un modelado visual independiente del lenguaje de implementación dando la posibilidad de que los diseños realizados utilizando UML se puedan implementar en cualquier lenguaje de programación que soporte las posibilidades de UML.(Principalmente lenguajes de programación OO). Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Es un método formal de modelado que aporta muchas ventajas, tales como:

- Permite expresar el sistema de modo gráfico, de forma tal que otro pueda entender.
- Proporciona un vocabulario y reglas para permitir la comunicación y se centra en la representación gráfica del sistema.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y viceversa.
- Mayor rigor en la especificación.

1.8.4. Framework Symfony v1.0.18.

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

“Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.”[8].

El mismo fue desarrollado en PHP 5. Ha sido probado en proyectos de gran envergadura, es compatible con casi todos los gestores de base de datos como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar en varios sistemas operativos. Incluye a Propel como mapeador Objeto-Relacional, Creole como capa de abstracción de la Base de Datos. Además puede ser fácilmente configurable con otros plugins probados, tales como el sfGuardPlugins para facilitar el proceso de autenticación. Symfony simplifica el desarrollo de la aplicación al presentar elementos que permiten garantizar la seguridad.

Esta herramienta fue desarrollada con el objetivo de que ajustara a los siguientes requisitos:

- “Fácil de instala y configurar.
- Independiente del sistema gestor de base de datos.
- Sencillo de usar y flexible.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.” [9].

Symfony puede ser completamente personalizado para cumplir con los requisitos, las políticas y reglas dispuestas para la gestión de proyectos y la programación de aplicaciones y automatiza la mayoría de los elementos comunes de los proyectos web, como por ejemplo:

- La capa de internacionalización que incluye, permite la traducción de los datos y de la interfaz.
- La capa de presentación utiliza plantillas y *layouts*
- Los *helpers* incluidos permiten minimizar el código utilizado en la presentación, pues encapsulan grandes bloques de código.
- Los formularios incluyen validación automatizada y relleno automático de datos (*"repopulation"*),
- Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- Las interacciones con Ajax son muy fáciles de implementar mediante los *helpers* que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código.

1.8.5. Lenguaje de programación PHP v5.1.6.

“PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor.”[10]

“PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.” [11].

“Existen tres campos en los que se usan scripts escritos en PHP.

- Scripts del lado del servidor. Este es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto funcione. El intérprete PHP (CGI ó módulo), un servidor web y un navegador. Es necesario correr el servidor web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectándose con el servidor web.
- Scripts en la línea de comandos. Puede crear un script PHP y correrlo sin ningún servidor web o navegador. Solamente necesita el intérprete PHP para usarlo de esta manera.
- Escribir aplicaciones de interfaz gráfica. ” [11].

PHP puede utilizarse en los más populares sistemas operativos con los que se cuenta hoy en día, entre ellos: Windows, Linux, Mac OS. También tiene soporte para la mayoría de los servidores web que existen como: Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, etc.

Es posible usar programación procedimental o programación orientada a objetos usando PHP. Muchas bibliotecas y aplicaciones de gran tamaño (entre ellas la biblioteca PEAR) están escritas usando programación orientada a objetos solamente.

Este lenguaje soporta conexiones con más de 20 servidores bases de datos diferentes y cuenta con una extensión DBX de abstracción de base de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. Por otra parte soporta Estándar Abierto de Conexión con Bases de Datos (ODBC por sus siglas en inglés), de manera que puede conectarse a cualquier base de datos que soporte tal estándar. En alasLIPO v2.0, PHP se usa en combinación con el gestor de bases de datos MYSQL.

Cuenta también con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros.

1.8.6. Gestor de base de datos: MYSQL v5.0.24.

MySQL es un servidor de base de datos muy rápido y sencillo de usar, multiusuario, multiplataforma (Windows, Linux, Mac OS, etc.) y robusto escrito en C/C++; desarrollado, distribuido y con soporte por parte de MySQL AB y puede ser adquirido bajo dos licencias, el usuario puede adquirir una versión OpenSource con términos de GNU General Public License o puede comprar la versión comercial estándar bajo licencia MySQL AB. Dispone de APIs para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.

Proporciona sistemas de almacenamientos transaccionales y no transaccionales. Posee un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. Brinda soporte a grandes bases de datos con millones de registros, cada tabla permite hasta 64 índices. Tiene una excelente integración con PHP.

1.8.7. APACHE 2.0.

Apache HTTP Server es un servidor HTTP de código abierto escrito en C y desarrollado como uno de los proyectos de Apache Software Foundation que funciona sobre la mayoría de los sistemas operativos incluyendo UNIX, MS-Windows, Macintosh y NetWare. Está basado en software libre y es distribuido bajo la licencia Apache de esta misma empresa. Es un servidor bastante seguro, eficiente y extensible, que proporciona servicios de HTTP en sincronización con los actuales estándares de este protocolo. Apache ha sido el más popular servidor web en Internet desde abril de 1996.

Apache dispone de varios módulos que no están incluidos en el núcleo (core) del servidor, algunos de estos módulos permiten que Apache brinde las funcionalidades básicas para un servidor web y otras lo convierten en mucho más que un servidor web básico ya que optimiza el rendimiento y la rapidez del código. Apache es utilizado en el desarrollo de software para el sistema de salud cubano.

1.8.8. Entorno de desarrollo integrado (IDE).

Un entorno de desarrollo integrado o, en inglés, Integrated Development Environment ('IDE'), es un programa compuesto por un conjunto de herramientas que proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. en favor de los programadores. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. En algunos lenguajes, pueden funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

Eclipse es una colección de proyectos OpenSource entre los cuales se encuentra un entorno de desarrollo integrado multiplataforma y distribuido bajo Licencia Publica de Eclipse y escrito con varios lenguajes de programación (Java, C, C++, JSP, etc.). Este IDE de Eclipse emplea módulos (en inglés *plug-in*) para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software.

En su última versión Eclipse incluye una contribución de Zend: **PHP Development Tools**, plugin para que se admitan proyectos PHP, formando parte del grupo de herramientas para desarrollo de otros lenguajes. Entre sus características están que es fácil de usar e intuitivo, integrable con *Web Tools* de Eclipse, extensible y con soporte continuo de desarrolladores PHP. Es una herramienta de desarrollo de código abierto que ofrece todas las funcionalidades básicas de edición de código. Permite modificar el código fuente, completamiento de código, utilizar plantillas, navegar por los diferentes elementos de PHP etc.

1.8.9. Herramienta CASE: Visual Paradigm v6.1.

“Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte

del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.” [12].

“Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.” [13].

Es una herramienta multiplataforma, que brinda muchas ventajas tales como que permite la generación de código (modelo a código, diagrama a código). Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos. Permite realizar ingeniería tanto directa como inversa de bases de datos desde Sistemas Gestores de Bases de Datos a diagramas de Entidad-Relación transformándolos luego en tablas de la base de datos. Es una herramienta generadora de informes. Distribuye automáticamente los diagramas y reorganiza las figuras y conectores de los diagramas UML. Permite la importación y exportación de ficheros XML.

1.8.10. Tecnología del lado del cliente: Java Script.

“**JavaScript** es un lenguaje interpretado que permite incluir macros en páginas Web. Estas macros se ejecutan en el ordenador del visitante de nuestras páginas, y no en el servidor (algo muy interesante, porque los servidores Web suelen estar sobrecargados, mientras que los PC's de los usuarios no suelen estarlo).

JavaScript proporciona los medios para:

- Controlar las ventanas del navegador y el contenido que muestran.
- Programar páginas dinámicas simples.
- Evitar depender del servidor Web para cálculos sencillos.
- Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet.
- Simular el comportamiento de las macros CGI cuando no es posible usarlas.
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.

- Comunicarse con el usuario mediante diversos métodos.” [14].

Es el lenguaje script más utilizado, es además un lenguaje interpretado que puede ser utilizado libremente sin adquirir ninguna licencia, soportado por la mayoría de los navegadores como: Internet Explorer, Firefox, Chrome, Opera y Safari. Diseñado para añadir interactividad a las páginas HTML y que usualmente está embebido en estas. Su sintaxis simple lo ha convertido en una herramienta de programación para los diseñadores HTML que generalmente no son programadores.

Hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, éste es el JavaScript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript; pero también existe un JavaScript que se ejecuta en el servidor, es más reciente y se denomina LiveWire JavaScript.

1.8.11. Subversion.

“Subversion es un sistema gratuito y open source para el control de versiones. Emplea licencia Apache/BSD. Se usa para mantener versiones actuales e históricas y los cambios de archivos tales como los de código fuente, páginas web y/o documentación” [15]. Se le conoce también como **svn** por ser el nombre de la herramienta de línea de comandos.

Subversion puede trabajar a través de redes, lo que permite que personas que estén en diferentes ordenadores puedan usarlo. Permite que varias personas modifiquen y gestionen el mismo conjunto de datos desde sus respectivos sitios, si se produce algún cambio incorrecto de los datos, sólo hace falta deshacerlo.

Ventajas:

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Maneja eficientemente archivos binarios
- Permite selectivamente el bloqueo de archivos.
- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos.

- Los programas asociados a Subversion que se ejecutan por línea de comandos tales como svn pueden ejecutarse tanto en plataformas Unix, Linux, Solaris o Microsoft Windows.

Existen varias interfaces a Subversion, ya sea programas individuales como interfaces que lo integran en entornos de desarrollo ejemplo de algunos de estos son:

- TortoiseSVN. Provee integración con el explorador de Windows. Es la interfaz más popular en este sistema operativo.
- Subclipse. "Plugin" que integra Subversion al entorno Eclipse.
- Subversive. "Plugin" alternativo para Eclipse.
- Cervisia Programa para interacción para Linux, combinada con Quanta Plus puede llegar a ser muy eficaz.
- Easyeclipse: Paquete basado en eclipse es una plataforma de desarrollo, con algunos plugins de código abierto.

1.8.12. DotProject.

“Aplicación web creada en PHP para administración de proyectos está construida por aplicaciones de código abierto, incluye módulos para compañías, proyectos, tareas (con gráficas de Gantt), foros, archivos, calendario, contactos, tickets de soporte. Es multiidioma y soporta distintos niveles de permisos de uso de módulos.”[16].

Utiliza inicialmente MySQL como base de datos (aunque otros motores como PostgreSQL también pueden ser utilizados). Software distribuido bajo licencia Freeware. La plataforma recomendada para utilizar dotProject se denomina LAMP (Linux + Apache+ MySQL+ PHP) aunque también es usado en la plataforma WAMP (Windows + Apache + MySQL+ PHP). Presenta una estructura modular que soporta características personalizables por el mismo. Tiene sus propios elementos de configuración, que pueden gestionarse a través del Entorno de Administración del Sistema.

DotProject brinda considerables ventajas tales como:

- Generar entornos colaborativos

- Trabajar en uno o más proyectos.
- Asignar recursos fácilmente a cada uno de los proyectos y actividades.
- Visualizar el estado de cada proyecto
- Entregar informes completos de las tareas realizadas.
- Conocer el porcentaje de adelanto o retraso de cada proceso, actividad o tarea.

1.8.13. Trac.

“**Trac** es un sistema web libre para la gestión de proyectos y Sistema de seguimiento de errores.” [17].

Trac fue desarrollado en Python y es mantenido por Edgewall Software. Estuvo disponible bajo GNU General Public License hasta mediados de 2005 y a partir de la versión 0.9, se distribuye de acuerdo a una modificación de la licencia BSD Trac License.

Dentro de sus principales características se encuentra que es un sistema multiplataforma que se integra fácilmente con Subversion, permite enlazar información entre una base de datos de errores de software, un sistema de control de versiones y el contenido de un wiki. Sirve como interfaz web de un sistema de control de versiones. Utiliza un sistema de plantillas web propio llamado Genshi y mantiene la compatibilidad con plug-ins basados en ClearSilve.

1.9. Patrones de diseño.

Un patrón es una descripción de un problema y la solución, a la que se le da un nombre, y se puede aplicar a nuevos contextos.

“Un patrón de diseño es:

- Una solución estándar para un problema común de programación.
- Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- Un diseño o estructura de implementación que logra una finalidad determinada.
- Un estilo de programación de alto nivel.
- Una manera más práctica de describir ciertos aspectos de la organización de un programa.
- Conexiones entre componentes de programas.

- La forma de un diagrama de objeto o de un modelo de objeto”. [18].

1.9.1. Patrones GRASP.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Constituyen un apoyo para entender el diseño y aplica el razonamiento para el diseño de una forma sistemática, racional y aplicable. Los 5 patrones básicos se refieren a cuestiones y aspectos fundamentales del diseño, algunos de estos patrones utilizados en este trabajo son:

- **“Experto:** Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- **Creador:** Asignarle a la clase B la responsabilidad de crear una instancia de clase A en uno de los siguientes casos:
 - B agrega los objetos A.
 - B contiene los objetos A.
 - B registra las instancias de los objetos A.
 - B utiliza específicamente los objetos A
 - B contiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A).

B es un creador de los objetos A. Si existe más de una opción, prefiera la clase B que agregue o contenga a la clase A.

- **Alta Cohesión:** Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo por hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande.
- **Bajo Acoplamiento:** Asignar una responsabilidad para mantener bajo acoplamiento. Las clases deben comunicarse con un número pequeño de clases tanto como sea posible.

- **Controlador:** Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.
Utilice la misma clase de controlador con todos los eventos del sistema en el mismo caso de uso”. [19].

1.9.2. Patrones GoF.

Los patrones GoF se clasifican en 3 grandes categorías basadas en su **propósito**: creacionales, estructurales y de comportamiento.

- **Creacionales:** Patrones creacionales tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.
- **Estructurales:** Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.
- **Comportamiento:** Los patrones de comportamiento nos ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

Los patrones GoF se clasifican también en 2 **ámbitos**: Clases y objetos por eso se tienen 6 tipos de patrones:

“Creacionales:

- Creacional de la Clase: Usan la herencia como un mecanismo para lograr la instanciación de la Clase. Por ejemplo el método Factoría.
- Creacional del objeto: Son más escalables y dinámicos comparados de los patrones creacionales de Clases. Por ejemplo la Factoría abstracta y el patrón Singleton.

Estructurales:

- Estructural de la Clase: Usan la herencia para proporcionar interfaces más útiles combinando la funcionalidad de múltiples Clases. Por ejemplo el patrón Adaptador (Clase).
- Estructural de Objetos: Crean objetos complejos agregando objetos individuales para construir grandes estructuras. La composición del patrón estructural del objeto puede ser cambiado en tiempo de ejecución, el cual nos da flexibilidad adicional sobre los patrones estructurales de Clases. Por ejemplo el Adaptador (Objeto), Facade, Bridge, Composite.

Comportamiento:

- Comportamiento de Clase: Usan la herencia para distribuir el comportamiento entre Clases. Por ejemplo Interpreter.
- Comportamiento de Objeto: Permite analizar los patrones de comunicación entre objetos interconectados, como objetos incluidos en un objeto complejo. Ejemplo Iterator, Observer, Visitor". [20].

1.9.3. Tipos de patrones GoF que implementa Symfony.

Symfony, framework que se utiliza para la implementación de la solución del presente trabajo; implementa una serie de patrones GoF entre los que se encuentran:

En la categoría de **creacionales** Symfony utiliza el patrón:

- **Decorator**: Añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades.

En Symfony el contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla.

- **Singleton** (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

- **Command:** Encapsula peticiones en forma de objetos permitiendo así parametrizar los clientes utilizando distintas peticiones, encolar las peticiones y ofrecer la posibilidad de deshacer las operaciones. Permite solicitar operaciones sin tener que saber cómo o quien lleva a cabo esas operaciones.

En la categoría de **comportamiento**:

- **Observer:** Para no recurrir a soluciones fuertemente acopladas (que reducen la posibilidad de reutilización), este patrón define una dependencia “uno-a-muchos” entre objetos, para que, cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan actualizarse convenientemente.
- **Registry (Registro):** Provee un mecanismo para almacenar datos globales, esto permite la persistencia de estos datos durante la vida del sistema.
- **Controller (Controlador Frontal):** Permitir que sólo exista un único punto de entrada en la aplicación, esto ayuda a centralizar las restricciones de seguridad y a delegar las peticiones en otros componentes.

1.10. Patrón de arquitectura Modelo Vista Controlador (MVC).

Un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular recurrente que surge en un cierto contexto. Este esquema se especifica describiendo las componentes, con sus responsabilidades, relaciones, y las formas en que colaboran. Dentro de los patrones de arquitectura se puede encontrar el patrón Modelo Vista Controlador (MVC) y el patrón Modelo de Tres Capas.

El desarrollo de la aplicación informática que acompaña el presente trabajo de diploma se utilizará el framework Symfony, el cual está basado en el patrón clásico de MVC, que está formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes.

1.11. Conclusiones.

De acuerdo al estudio realizado a los sistemas existentes a nivel mundial y nacional se llegó a la conclusión de que no existe uno que integre todas las funcionalidades dedicadas a diagnosticar , tratar , prevenir y dar seguimiento a las dislipoproteinemias de ahí la necesidad de realizar el diseño e implementación de una aplicación (alasLIPO v2.0) donde se adhirieran todas estas. Para obtener un buen desarrollo de esta nueva versión se realizó una fundamentación de la metodología, de la arquitectura propuesta por el SNS para el despliegue de aplicaciones en INFOMED, de los patrones, de las técnicas y herramientas a utilizar, todo esto con el objetivo de garantizar la adecuada prestación de los servicios.

Capítulo 2: Diseño del Sistema.

Introducción.

En el presente capítulo se exponen elementos importantes para una solución satisfactoria tales como los requisitos funcionales y no funcionales así como una descripción resumida de los casos de uso del sistema. Además se realiza el diseño del sistema obteniéndose los diagramas de clases del diseño web, interacción (secuencia), despliegue, clases persistentes y el modelo de datos.

2.1. Características del sistema.

2.1.1. Requisitos funcionales y no funcionales del sistema.

Los requisitos funcionales:

- “Son declaraciones de los servicios que debe proporcionar el sistema.
- Especifica la manera en que éste debe reaccionar a determinadas entradas.
- Especifica cómo debe comportarse el sistema en situaciones particulares.
- Pueden declarar explícitamente lo que el sistema no debe hacer.” [21].

Para dar cumplimiento a las funcionalidades del sistema se detectaron 31 requerimientos funcionales, a continuación se listan los mismos:

R1. Validar usuario y contraseña de los usuarios del sistema.

R2. Calcular el valor del peso ideal.

R3. Calcular el valor de las calorías de la dieta.

R4. Calcular el valor del índice de masa corporal.

R5. Calcular el valor del riesgo cardiovascular.

R6. Registrar los datos del paciente.

R6.1. Generar diagnóstico de la primera consulta asociado al paciente.

R6.2 Generar tratamiento de la primera consulta asociado al diagnóstico del paciente.

R7. Actualizar los datos del paciente.

R8. Buscar paciente según criterios de búsqueda especificados.

R9. Registrar datos de la consulta evolutiva asociada a un paciente.

R9.1. Generar diagnóstico de la consulta evolutiva asociado al paciente.

R9.2. Generar tratamiento de la consulta evolutiva asociado al diagnóstico del paciente.

R10. Visualizar los datos de la historia clínica del paciente.

R10.1. Generar gráficos del estado del paciente.

R10.2. Visualizar los datos del paciente asociados a la primera consulta.

R10.3. Visualizar los datos del paciente asociados a la consulta evolutiva.

R10.4. Imprimir los datos de la historia clínica del paciente referentes a la primera consulta y a la consulta evolutiva.

R11. Generar reporte según criterios indicados.

R11.1 Exportar el resultado del reporte generado.

R11.2 Imprimir el resultado del reporte generado.

R12. Insertar un nuevo medicamento.

R13. Buscar medicamentos existentes.

R14. Visualizar medicamentos existentes.

R15. Eliminar un medicamento.

R16. Actualizar los datos de un medicamento.

R17. Insertar un nuevo documento.

R18. Buscar documentos existentes.

- R19. Visualizar documentos existentes.
- R20. Eliminar un documento.
- R21. Actualizar los datos de un documento.
- R22. Solicitar trasladar un paciente de la consulta.
- R23. Trasladar un paciente de la consulta.
- R24. Buscar datos de los usuarios existentes.
- R25. Visualizar usuarios existentes.
- R26. Asignar permisos a los usuarios.
- R27. Eliminar un usuario.
- R28. Registrar los datos de un usuario.
- R29. Insertar documento con el árbol genealógico del paciente.
- R30. Buscar documentos con el árbol genealógico del paciente.
- R31. Visualizar documentos con el árbol genealógico del paciente.

Los Requisitos No Funcionales:

- “No se refieren a funciones específicas que proporciona el sistema.
- Son restricciones de los servicios o funciones ofrecidas por el sistema (fiabilidad, tiempo de respuestas, capacidad de almacenamiento, etc.)
- Generalmente se aplican al sistema en su totalidad.
- Surgen de las necesidades del usuario (restricciones de presupuesto, políticas de la organización, necesidad de interoperatividad, etc.).
- Definen propiedades y restricciones del sistema: tiempos de respuesta, requisitos de almacenamiento, etcétera.
- También puede referirse al proceso de desarrollo: uso de una herramienta CASE, lenguaje de programación o metodología concreta

- En general, suelen ser más críticos que los requisitos funcionales. Si no son satisfechos, el sistema no resulta útil. ” [21].

El sistema alasLIPO v2.0 debe cumplir ciertas características tales como:

R1. Apariencia o interfaz externa.

La aplicación alasLIPOv2.0 presentará un color que denote seriedad debido a su contenido además deberá ser atractivo a la vista del usuario, se seleccionará como color predominante el verde. Se brindará información concreta y útil, con el objetivo de mantener a todos los usuarios informados y guiados en la prevención y el tratamiento de las dislipoproteinemias. Sólo se incorporarán las imágenes necesarias para la comprensión de términos médicos e imágenes propias de Cuba.

R2. Usabilidad.

En la aplicación se garantizará un acceso fácil y rápido a los usuarios. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora y de un ambiente Web en sentido general.

R3. Rendimiento.

Garantizará tiempos de respuestas y velocidad de procesamiento de la información generalmente rápidos.

R4. Seguridad.

El sistema deberá contar con un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría (Authentication, Authorization and Accounting, AAA) con Autenticación de firma única (Single Sign On). Se deberá garantizar que la información solo pueda ser vista por los usuarios con el nivel de acceso autorizado para ello; permitiendo que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo. El sistema deberá contar con protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. Deberá garantizar confidencialidad de los datos, permitiendo que cada médico solo pueda ver los datos de sus pacientes.

R5. Políticos – Culturales.

En la aplicación se deberá utilizar idioma español. Contará con logotipos e imágenes que se encuentren en correspondencia con el carácter científico y profesional del tema. Algún cambio que se quiera realizar deberá ser previamente consultado con el Dr. Alfredo Nasiff Hadad y el proyecto alasLIPOv2.0.

R6. Legales.

El sistema se desplegará en los servidores de Infomed como herramienta para la ayuda al diagnóstico y tratamiento de las dislipoproteinemias. Se usarán principalmente herramientas de software libre, bajo la licencia GNU/GPL.

R7. Software.

La máquina servidor deberá disponer del servidor Web Apache versión 2.0. Además se debe constar con el sistema gestor de base de datos MySQL 5.0.24. El lenguaje de programación debe ser PHP versión 5.1.6.

Las máquinas clientes podrán contar con sistema operativo Windows XP o superior o cualquier distribución de Linux. Las máquinas clientes para acceder al sistema deben hacerlo a través del navegador web Mozilla o Internet Explorer. Dichas máquinas deben tener instalado el Adobe Acrobat para la impresión y el Microsoft Office Excel para exportar el reporte generado.

R8. Hardware.

La máquina servidor deberán tener como mínimo 80 GB de capacidad de disco duro, microprocesador superior a 1.00 GHz, 1GB mínimo de memoria RAM. Las Capas de Presentación, Negocio y Datos estarán desplegadas en dos servidores.

R9. Restricciones en el diseño y la implementación.

La lógica de presentación constituirá una capa independiente de la lógica de negocio, centrando su función en la interfaz de usuario y validaciones de los datos de entrada. Se utilizará como lenguaje de programación PHP 5.1.6, como Gestor de Base de Datos MySQL 5.0.24, como Framework de

Desarrollo Symfony 1.013 y como IDE Eclipse. Visual Paradigm 3.1 Enterprise Edition como herramienta CASE para el modelado de los artefactos.

R10. Confiabilidad.

El sistema deberá ser administrado solamente por una persona capacitada y será usada por médicos registrados en el SAAA, por tanto, la información que fluirá será la que se emite realmente en una consulta de un médico. Se validará el registro de los datos del sistema disminuyendo el error humano y así la información sea lo más confiable posible.

R11. Requisitos para la documentación de usuarios en línea y ayuda del sistema.

Se realizará un Manual de Usuario para ayudar a los usuarios a trabajar con el sistema. Se realizará una ayuda online para ayudar al usuario a trabajar con el sistema.

R12. Requisito de interfaz interna.

Se tendrá que implementar una interfaz interna que permita autenticar al usuario con el servicio SAAA.

2.1.2. Diagrama de Casos de Uso del sistema.

“El Diagrama de Casos de Uso es un diagrama diseñado para el desarrollo de software dentro del contexto de la Metodología Iterativa Incremental. Brinda valiosa información para el desarrollo de un proyecto, así mismo es un puente de comunicación entre el cliente, el analista y el programador.”[22].

“Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto.”[23].

A partir de los requisitos funcionales identificados en el epígrafe anterior se realizó el diagrama de casos de usos del sistema, donde se especifican las relaciones entre los actores y las funcionalidades de alasLIPO v2.0, con un total de 15 casos de usos y 7 actores identificados.

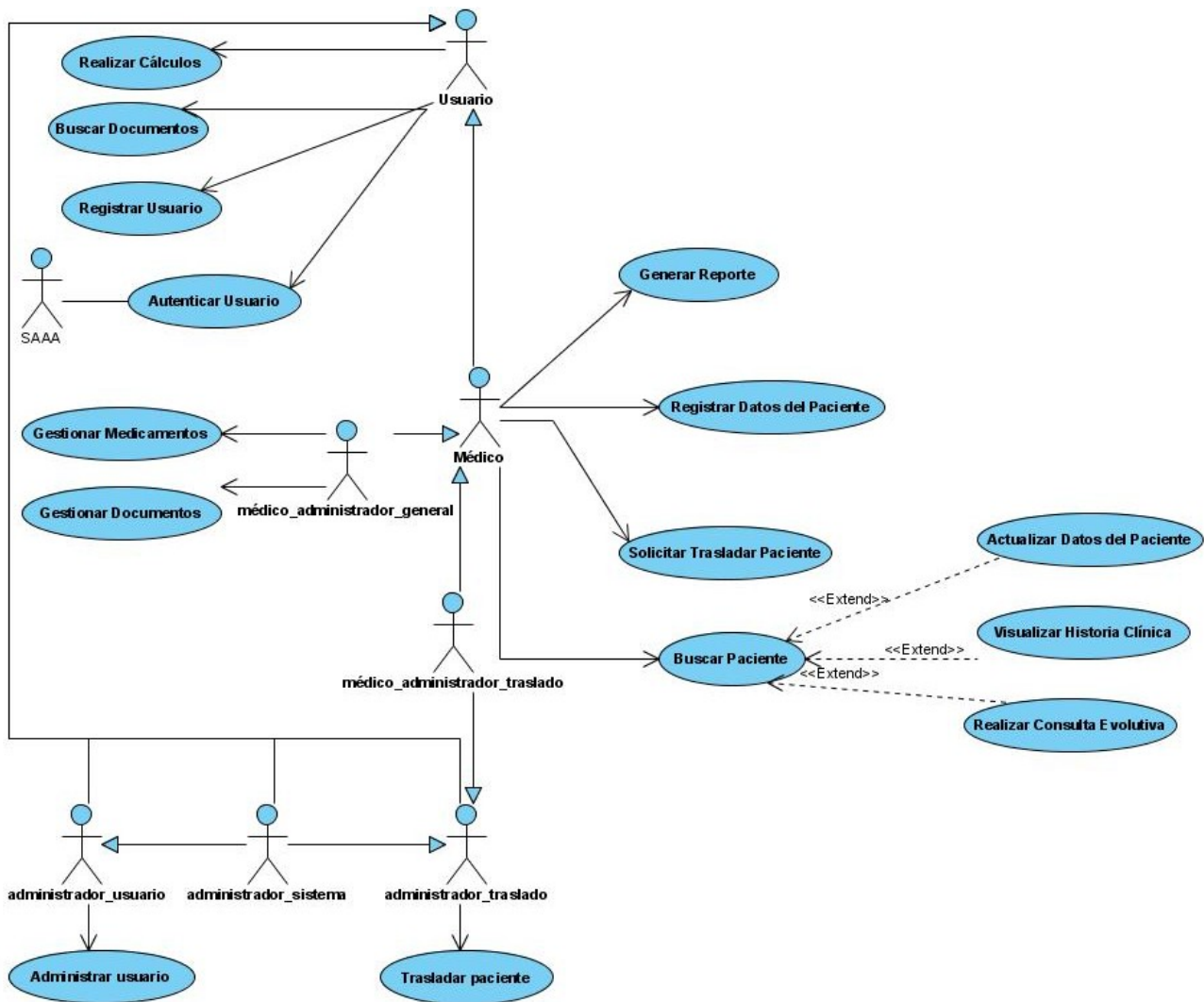


Fig. 1: Diagrama de Casos de Uso del sistema.

2.1.3. Actores del sistema.

Los actores del sistema no siempre representan personas. Pueden ser actores otros sistemas o hardware externos que interactúan con este. Los actores se comunican con el sistema mediante el envío y recepción de mensajes hacia y desde según éste lleve a cabo los casos de uso. Podemos

encontrar y especificar todos los actores examinando a los usuarios que utilizaran el sistema y a otros sistemas que debe interactuar con el.

Actor	Descripción
Usuario	Interactúa con el cálculo del peso ideal, cálculo de factores de riesgos, cálculo de índice de masa corporal y el cálculo de dietas según calorías, además de navegar por el sistema en la parte informativa (documentos publicados, glosario de términos, dieta de las dislipoproteinemias, factores de riesgo cardiovascular).
Médico	Puede realizar las funcionalidades del usuario. Además gestiona toda la información relacionada con la atención al paciente. Puede registrar los datos de los pacientes asociados a la primera consulta, completarlos, buscarlos, imprimirlos. Además registrar y buscar los datos del paciente asociados a la consulta evolutiva, pedir solicitud de traslado de un paciente hacia otra consulta, generar un reporte según criterios indicados.
SAAA (Actor externo)	Gestiona el proceso de autenticación al sistema. Es un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría (Authentication, Authorization and Accounting, AAA) con Autenticación de firma única (Single Sign On).
Médico_administrador_general	Puede realizar las actividades de los roles de médico y de usuario. Además gestiona los medicamentos y los documentos a publicarse en la aplicación.
Administrador_sistema	Puede realizar las actividades del usuario y además las

	funcionalidades de los roles administrador_traslado, y administrador_usuario.
médico_administrador_traslado	Puede realizar las actividades de los roles médico, usuario y administrador_traslado.
administrador_traslado	Puede realizar las actividades del rol de usuario. Es el encargado del traslado de un paciente para otra consulta.
administrador_usuario	Puede realizar las actividades del rol de usuario. Es el encargado de administrar los usuarios existentes en el sistema

Tabla #1: Actores del sistema.

2.1.4. Descripción de los Casos de Uso del sistema.

A continuación se muestran las descripciones textuales resumidas de los casos de uso determinados para satisfacer los requerimientos funcionales de sistema, para el estudio de las descripciones expandidas remitirse al **Anexo 1** "Descripción expandida de los casos de uso".

Nombre del Caso de Uso	Gestionar Medicamentos
Actores	Médico_administrador_general
Propósito	Insertar, actualizar o eliminar medicamentos
Resumen	El CU inicia cuando el Administrador general desea insertar, actualizar o eliminar un medicamento. Luego de que el Administrador general indique los datos del medicamento a insertar, actualizar o eliminar entonces el sistema registra, actualiza o elimina el medicamento correspondientemente.
Referencias	R12, R13, R14, R15, R16

Tabla #2: Descripción resumida del Caso de Uso: Gestionar Medicamento.

Nombre del Caso de Uso	Realizar Cálculos
Actores	Usuario
Propósito	Realizar cálculos de peso ideal, calorías de la dieta, el índice de masa corporal y el factor de riesgo cardiovascular del paciente.
Resumen	El CU inicia cuando el Usuario desea realizar los cálculos del peso ideal, las calorías de la dieta, el índice de masa corporal y el factor de riesgo cardiovascular. Luego de seleccionado el cálculo que desea realizar el usuario e indicando los datos necesarios, entonces el sistema obtiene los resultados asociados a los datos introducidos.
Referencias	R2, R3, R4, R5

Tabla #3: Descripción resumida del Caso de Uso: Realizar Cálculos.

Nombre del Caso de Uso	Generar Reporte
Actores	Médico
Propósito	Generar Reporte
Resumen	El CU se inicia cuando el Médico indica generar un reporte. Luego de indicar los criterios para realizarlo y seleccionar los criterios que se mostrarán en el resultado del mismo el sistema genera el reporte. Luego el sistema permite que se exporte a formato Excel o imprima el resultado del reporte generado.
Referencias	R11, R11.1, R11.2

Tabla #4: Descripción resumida del Caso de Uso: Generar Reporte.

Nombre del Caso de Uso	Registrar Datos del Paciente
Actores	Médico
Propósito	Registrar los datos de un nuevo paciente.
Resumen	El CU se inicia cuando el Médico indica registrar los datos de un nuevo paciente. Luego el sistema muestra la interfaz referente a la inserción de dichos datos, el Médico indica los mismos y entonces el sistema verifica que los datos estén correctos y sean los necesarios, además el sistema verifica que no exista otro paciente registrado con el mismo carnet de identidad o identidad. Luego de cumplirse todas estas condiciones entonces se guardan los datos asociados a dicho paciente y se genera y registra de ser posible un diagnóstico y tratamiento asociado al paciente.
Referencias	R6, R6.1, R6.2

Tabla #5: Descripción resumida del Caso de Uso: Registrar Datos del Paciente.

Nombre del Caso de Uso	Buscar Paciente
Actores	Médico
Propósito	Se buscan los pacientes que coincidan con los criterios de búsqueda indicado por el médico y que pertenezcan al médico en cuestión.
Resumen	El CU se inicia cuando el Médico desea buscar un paciente. Para realizar la búsqueda del paciente el Médico indica los criterios de búsqueda (nombre, primer apellido, segundo apellido y carnet de identidad). Luego el sistema muestra un listado con los pacientes que coincidan con los criterios indicados por el Médico. En caso de que el Médico no indique

	<p>criterios de búsqueda entonces el sistema muestra todos los pacientes existentes. En ambos casos sólo se muestran los pacientes pertenecientes al Médico en cuestión.</p>
Referencias	R8

Tabla #6: Descripción resumida del Caso de Uso: Buscar Paciente.

Nombre del Caso de Uso	Realizar Consulta Evolutiva
Actores.	Médico.
Propósito	Registrar los datos de una consulta evolutiva realizada al paciente.
Resumen	<p>El CU inicia cuando el sistema muestra la interfaz correspondiente a la Consulta Evolutiva y luego indica los datos asociados a dicha consulta y a partir de los datos indicados se genera un nuevo diagnóstico y tratamiento del paciente. También el Médico puede visualizar los cálculos auxiliares asociados a los datos registrados y/o emitir sus observaciones. El sistema registra todos los datos generados e indicados.</p>
Referencias	R9, R9.1, R9.2.

Tabla #7: Descripción resumida del Caso de Uso: Realizar Consulta Evolutiva.

Nombre del Caso de Uso	Actualizar Datos del Paciente
Actores	Médico
Propósito	Actualizar la primera consulta con los datos que aún no se hayan registrado.
Resumen	El CU se inicia cuando el sistema muestra la interfaz

	correspondiente para actualizar los datos del paciente que faltaban de la primera consulta. Luego del Médico indicar los datos a completar el sistema verifica que estén correctos y en caso positivo los guarda. En caso de que los datos no estén correctos entonces se muestra una alerta.
Referencias	R7

Tabla #8: Descripción resumida del Caso de Uso: Actualizar Datos del Paciente.

Nombre del Caso de Uso	Visualizar Historia Clínica
Actores	Médico
Propósito	Se muestra la Historia Clínica (Primera Consulta y Consulta Evolutiva) asociada al paciente.
Resumen	El CU se inicia cuando el sistema muestra la interfaz correspondiente a la visualización de la Historia Clínica y además brinda la posibilidad de visualizar los datos de todas las consultas realizadas al paciente. También permite imprimir la Historia Clínica del paciente, insertar y visualizar el árbol genealógico del paciente que presente los triglicéridos por encima de 400mg/dl y muestra las gráficas con el estado del paciente hasta el momento.
Referencias	R10, R10.1, R10.2, R10.3, R10.4,R29.R.30

Tabla #9: Descripción resumida del Caso de Uso: Visualizar Historia Clínica.

Nombre del Caso de Uso	Autenticar Usuario
Actores	Médico, SAAA
Propósito	Permitir que sólo los Médicos registrados en el SAA puedan

	gestionar la información relacionada a la atención y seguimiento de los pacientes en el sistema.
Resumen	El Médico introduce su usuario y contraseña. Luego el SAAA verifica que es usuario esté autorizado y en caso positivo el sistema brinda acceso a las funcionalidades según los privilegios del usuario.
Referencias	R1

Tabla #10: Descripción resumida del Caso de Uso: Autenticar Usuario.

Nombre del Caso de Uso	Solicitar Traslado de Paciente
Actores	Médico
Propósito	Solicitar el traslado de un paciente de una consulta para otra.
Resumen	El CU se inicia cuando un Médico desea solicitar el traslado de un paciente para su consulta. El Médico indica buscar los pacientes existentes y luego indica cual es el paciente a solicitar su traslado. Luego el sistema registra la solicitud realizada por el Médico en cuestión.
Referencias	R22

Tabla #11: Descripción resumida del Caso de Uso: Solicitar Traslado de Paciente.

Nombre del Caso de Uso	Trasladar Pacientes
Actores	Administrador_traslado
Propósito	Autorizar el traslado de un paciente para otra consulta.
Resumen	El CU se inicia cuando el Administrador de sistema indica realizar el traslado de los pacientes pendientes a trasladar

	para otra consulta.
Referencias	R23

Tabla #12: Descripción resumida del Caso de Uso: Trasladar Pacientes.

Nombre del Caso de Uso	Buscar Documentos
Actores	Usuario
Propósito	Buscar y visualizar documentos existentes.
Resumen	El CU inicia cuando el Usuario desea buscar los documentos existentes. Luego el sistema muestra una interfaz con el listado de los documentos y sus datos.
Referencias	R18, R19

Tabla #13: Descripción resumida del Caso de Uso: Buscar Documentos.

Nombre del Caso de Uso	Gestionar Documentos
Actores	Médico_administrador_general
Propósito	Insertar, actualizar o eliminar documentos
Resumen	El CU inicia cuando el Administrador general desea insertar, actualizar o eliminar un documento. Luego de que el administrador general indique los datos del documento a insertar, actualizar o eliminar entonces el sistema registra, actualiza o elimina el medicamento correspondientemente.
Referencias	R17, R20, R21

Tabla #14: Descripción resumida del Caso de Uso: Gestionar Documentos.

Nombre del Caso de Uso	Administrar Usuarios
Actores	Administrador_usuario

Propósito	Buscar, visualizar, eliminar un usuario y asignar permisos según el nivel de acceso del mismo.
Resumen	El CU inicia cuando el Administrador del sistema desea visualizar, eliminar un usuario o asignar permisos según el nivel de acceso del mismo. Luego el sistema muestra un interfaz con todos los usuarios y da la posibilidad de eliminarlo o asignarle el nivel de acceso. Los usuarios que saldrán en el listado serán según el nivel de acceso del administrador de sistema.
Referencias	R24, R25, R26, R27

Tabla #15: Descripción resumida del Caso de Uso: Administrar Usuarios.

Nombre del Caso de Uso	Registrar Usuario
Actores	Usuario
Propósito	Registrar los datos de un usuario en el sistema.
Resumen	El CU inicia cuando el Usuario desea registrarse en el sistema e introduce los datos necesarios para el registro. Luego el sistema registra los datos indicados por el usuario y lo ubica pendiente de autorización para acceso
Referencias	R28

Tabla #16: Descripción resumida del Caso de Uso: Registrar Usuario.

2.2. Arquitectura y diseño.

2.2.1. Patrón arquitectónico MVC.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email). El modelo se encarga de la abstracción de la lógica relacionada con los datos,

haciendo que la vista y las acciones sean independientes (tipo de gestor de bases de datos utilizado por la aplicación).

Para el desarrollo del sistema se escogió el framework Symfony, framework “que está basado en un patrón clásico del diseño web conocido como arquitectura MVC, formado por 3 niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista”. [7].

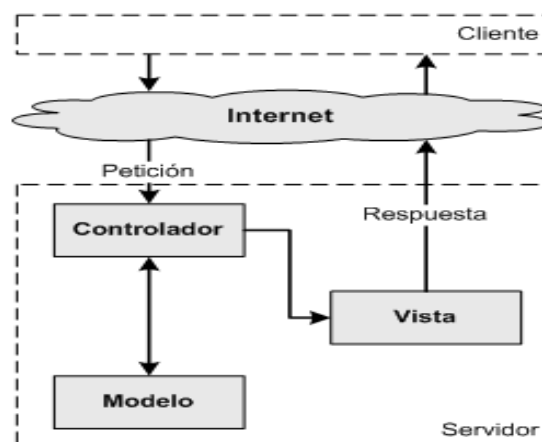


Fig. 1: Patrón Modelo, Vista, Controlador (MVC).

El uso de Symfony resulta bastante útil además de restrictivo ya que obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el mismo. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador.

“La implementación que realiza Symfony de la arquitectura MVC incluye varias clases como son:

- **sfController:** es la clase del controlador. Se encarga de decodificar la petición y transferirla a la acción correspondiente.
- **sfRequest:** almacena todos los elementos que forman la petición (parámetros, cookies, cabeceras)
- **sfResponse:** contiene las cabeceras de la respuesta y los contenidos. El contenido de este objeto se transforma en la respuesta HTML que se envía al usuario.
- El **singleton** de contexto (que se obtiene mediante `sfContext::getInstance ()`) almacena una referencia a todos los objetos que forman el núcleo de Symfony y puede ser accedido desde cualquier punto de la aplicación.

Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo”. [7].

El controlador frontal es un componente generado automáticamente por el framework que, unido con el layout son comunes para todas las páginas de la aplicación. Las clases de la capa del modelo son generadas automáticamente en función de la estructura de datos de la aplicación por la librería Propel y la abstracción de la base de datos es completamente invisible al programador, ya que es realizada mediante un componente llamado Creole.

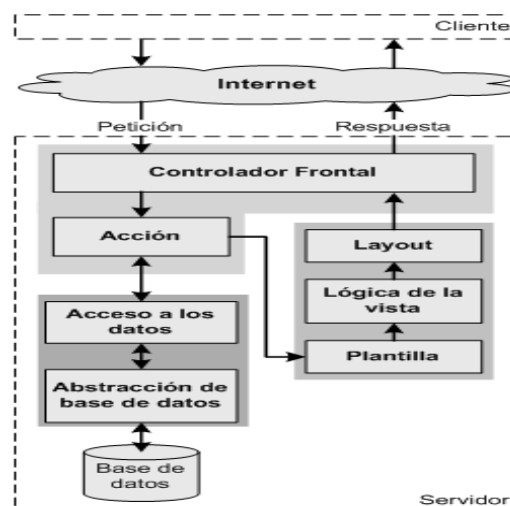


Fig. 3: El flujo de trabajo de Symfony.

2.2.2. Aplicación de patrones GRASP en Symfony.

Se debe conocer la arquitectura del framework para una mejor comprensión del comportamiento que seguirán las clases contenidas en él y su estructura, la arquitectura brinda una vista panorámica del diseño de este sistema. Son muchos los patrones que se utilizan en la implementación con Symfony, a continuación se mencionan algunos ejemplos de los evidenciados, ubicándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC.

Patrón Experto: En el modelo de la arquitectura Symfony existen dos tipos de clases que son fundamentales:

- Las encargadas de la abstracción de datos. (realizan todas las operaciones con la BD).
- Las de acceso a datos. (interactúan con las clases de abstracción de datos y devuelven los objetos necesarios por los controladores).

Por cada tabla Symfony genera 4 tablas: Clase, ClasePeer, BaseClase y BaseClasePeer (ejemplo de clases: Paciente, PacientePeer, BasePaciente, BasePacientePeer). Las clases a las que el framework añade el sufijo "Peer" trabajan directamente con la base de datos y por lo tanto son las encargadas de la abstracción utilizando Propel, en ellas se encuentran los atributos necesarios para este proceso, de ahí la necesidad de que implementen la responsabilidad de efectuar las operaciones con la base de datos, aplicando de esta manera el patrón experto.

Patrón Creador: Con Symfony se pueden crear objetos de varias formas, mayormente mediante métodos estáticos, para ver un ejemplo de esto se debe analizar cuando un "action" desea crear una instancia de una clase modelo, en el ejemplo el "action" obtiene el identificador de paciente (en este caso el número de la historia clínica) y desea ver los datos del mismo, para esto necesita entonces, crear una instancia de este paciente.


```
public function executeVerprimeraconsulta()
{
    $pac=    PacientePeer::retrieveByPK($this->getRequestParameter('hc'));
    $this->paciente=$pac;
}
}
```

Fig. 4: Muestra del método “**Verprimeraconsulta ()**” donde se evidencia el patrón creador.

En el ejemplo el “**action**” que sería el supuesto objeto B quien usa al objeto PacientePeer que es el supuesto objeto A, por tanto puede crear instancias de él y el patrón se cumple perfectamente.

Patrón Bajo Acoplamiento: Las clases que implementan la lógica de negocio y de acceso a datos se encuentran en el modelo, estas clases no tienen asociaciones con las de la vista o el controlador por lo que la dependencia en este caso es baja, cumpliéndose así perfectamente el patrón.

Alta Cohesión: El trabajar con Symfony permite la organización del trabajo en cuanto a la estructura del proyecto, esto proporciona crear y trabajar con clases con una alta cohesión. Ejemplo de esto se evidencia en la clase “**insercionhcActions**”, la cual está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas, teniendo un sentido común y un propósito único, siendo las mismas las encargadas de controlar las acciones de las plantillas.

Patrón Controlador: El controlador se encarga de asignar la responsabilidad de controlar el flujo de eventos del sistema a clases específicas con las que mantiene un modelo de alta cohesión, esto facilita la centralización de actividades (validaciones, seguridad.). Un ejemplo del patrón controlador se puede ver desde la clase sfFrontController, sfWebFrontController, sfContex, los “actions”, y el index.php del ambiente. Symfony implementa el patrón “Front Controller” (Controlador frontal), en el cual existen varias clases controladoras que forman un flujo para atender las peticiones del usuario y de otras clases. La arquitectura del framework (MVC) ayuda desde el principio, existiendo una capa específicamente para los controladores, que son el núcleo del mismo, el puesto de mando.

2.2.3. Aplicación de algunos patrones GoF que Implementa Symfony.

Patrón Decorador: En este método de la clase abstracta “sfView” padre de todas las vistas, tiene cada una un decorador para permitir añadir funcionalidades a las vistas dinámicamente. El archivo llamado layout.php contiene el layout de la página, este archivo se le denomina plantilla global, la cual almacena el código HTML que es común a todas las páginas de la aplicación, el contenido de la plantilla se integra en el Layout.

```
public function
getDecoratorTemplate()
{
    return $this->decoratorTemplate;
}
```

Fig. 5: Método “getDecoratorTemplate()”.

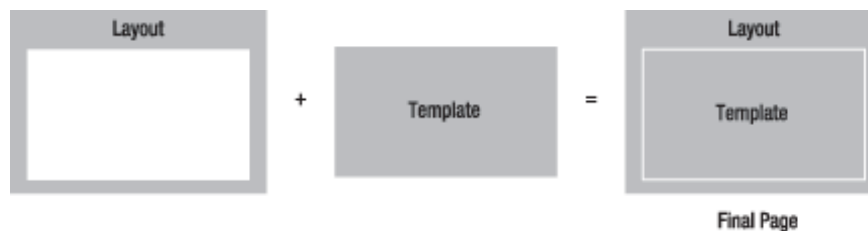


Fig.6: Plantilla decorada con un **Layout**.

Patrón Singleton: La clase sfRouting es una de es una de las que se encuentran en la capa Controlador del framework, esta clase es muy utilizada porque es la encargada de enrutar todas las peticiones que se hagan a la aplicación. El ejemplo tiene solamente un punto de creación, el cual es estático lo que permite entonces la aplicación perfecta del patrón.

```
public static function getInstance ()
{
    if (!isset (self::$instance))
    {
        self::$instance = new sfRouting();
    }

    return self::$instance;
}
```

Fig. 7: Método “getInstance” de la clase “sfRouting”.

Patrón Command: Se evidencia en la clase “sfFrontWebController”, esta clase es la encargada de determinar cual módulo y acción debe responder a las solicitudes de los usuarios.

```
class sfFrontWebController extends sfWebController
{
    /** ...
    public function dispatch()
    {
        try
        {
            if (sfConfig::get('sf_logging_enabled'))
            {
                $this->getContext()->getLogger()->info('{sfController} dispatch request
            }

            // reinitialize filters (needed for unit and functional tests)
            sfFilter::$filterCalled = array();

            // determine our module and action
            $request    = $this->getContext()->getRequest();
            $moduleName = $request->getParameter('module');
            $actionName = $request->getParameter('action');

            // make the first request
            $this->forward($moduleName, $actionName);
        }
    }
```

Fig. 8: Muestra de la clase “sfFrontWebController”.

Patrón Registry: Patrón sumamente útil, es un medio simple y eficiente de compartir datos y objetos sin tener que preocuparse de mantener numerosos parámetros o hacer uso de variables globales. La aplicación de este patrón se observa en la clase “sfConfig”, clase encargada de almacenar todas las variables de uso global en la aplicación. En el ejemplo se muestra el código de la clase “sfConfig” donde se crean los métodos globales y cómo el archivo “**config**” de la aplicación utiliza los métodos globales definidos en esta.

```

class sfConfig
{
    protected static
        $config = array();

    /** (...)
    public static function get($name, $default = null)
    { (...) }

    /** (...)
    public static function has($name)
    { (...) }

    /** (...)
    public static function set($name, $value)
    { (...) }

    /** (...)
    public static function add($parameters = array())
    { (...) }

    /** (...)
    public static function getAll()
    { (...) }

    /** (...)
    public static function clear()
    { (...) }
}

```

Fig.9: Clase “sfConfig”.

```
// include project configuration
include(SF_ROOT_DIR.DIRECTORY_SEPARATOR.'config'.DIRECTORY_SEPARATOR.'config.ph

// symfony bootstrapping
require_once($sf_symfony_lib_dir.'/util/sfCore.class.php');
sfCore::bootstrap($sf_symfony_lib_dir, $sf_symfony_data_dir);

sfConfig::set('paginado',15);

sfConfig::set('reporte_dir', SF_ROOT_DIR . DIRECTORY_SEPARATOR . 'web' . DIRECT

// Plugin dir
sfConfig::set('sf_xspchart_root_dir', SF_ROOT_DIR . DIRECTORY_SEPARATOR . 'plug

// Core pChart library dir
sfConfig::set('sf_xspchart_lib_dir', sfConfig::get('sf_xspchart_root_dir') . DI

// Dir for temporary files/pchar - system path
sfConfig::set('sf_xspchart_sys_web_tmp_dir', sfConfig::get('sf_web_dir') . DIRE
```

Fig. 10: Muestra del código del archivo “config” de la aplicación.

Patrón Front Controller: La aplicación de este patrón se manifiesta en la capa de Controladores, donde existe una estructura de clases bien definida, una de sus principales funciones es definir un solo punto de entrada para las peticiones de los usuarios, permitiendo un mejor control del flujo de eventos del sistema. La clase “sfController” se encarga de decodificar la petición y transferirla a la acción correspondiente.

```

abstract class sfController
{
protected
    $context                = null,
    $controllerClasses     = array(),
    $maxForwards           = 5,
    $renderMode            = sfView::RENDER_CLIENT,
    $viewCacheClassName    = null;

/** ...
public function componentExists($moduleName, $componentName)
{ ... }

/** ...
public function actionExists($moduleName, $actionName)
{ ... }

/** ...
protected function controllerExists($moduleName, $controllerName,
{ ... }

/** ...
public function forward($moduleName, $actionName)
{ ... }

/** ...
public function getAction($moduleName, $actionName)
{ ... }
}
    
```

Fig. 11: Muestra de la Clase “sfController”.

2.3. Pautas del diseño.

Para lograr una agradable apariencia y facilitar el uso de la aplicación se definieron algunas pautas de diseño como son:

- Los formularios que sean creados deben estar centrados con respecto a la sección donde es ubicado.
- Las tablas creadas para los formularios deben tener borde=0.
- La primera fila del formulario debe contener el nombre de la operación que se pretende realizar con el formulario. El estilo a utilizar para dicha fila es " hModulo", resaltando la palabra que resume la finalidad del formulario con el estilo "hModuloListado".

```
<td nowrap="nowrap" class="hModulo" align="left">
<?php echo image_tag('arrow.gif','title=""') ?>
<span>&nbsp;   Gestionar-</span> <span class="hModuloListado">Reporte.</span></td>
```

Fig. 12: Encabezado de formulario “Gestionar Reporte”.

- La palabra que resume la finalidad del formulario debe terminar con un guión (-).

❖ **Generar - Reporte.**

Fig. 13: Diseño de las finalidades de los formularios.

- La opción **cerrar** en la interfaces estará ubicada en el extremo inferior derecho.

❖ **Trasladar - Pacientes**

Resultados de la búsqueda: 1 pacientes encontrados.					
Paciente	Carnet de identidad	Médico al que pertenece	Médico que solicita traslado	Aceptar	Rechazar
Saidel Perez Dopaso	85041503748	Saidel Perez Dopaso	Liusmila Nieto Cervantes	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Cerrar

Fig. 14: Opción “cerrar” en la interfaz “Trasladar - Pacientes”.

- Cuando se le solicita datos al usuario se hará a través de una tabla de dos columnas. Esta tabla debe estar centrada en el formulario. En la columna izquierda se ubicará el nombre del dato solicitado al usuario, alineado a la derecha y en la columna de la derecha el componente adecuado para la solicitud. El estilo a utilizar en las filas del formulario es “**encabezadotabla**” y el del componentes “**entradaplana**”.

```
<table width="100%" border="0">
  <tr>
    <td class="encabezadotabla"><div align="right">
      <span class="hModuloListado">Peso:</span></div></td>
    <td class="encabezadotabla"><?php echo input_tag('peso',
      '', array('title'=>'Introduzca su peso', 'onkeypress'=>'return
      solonumeros(event)', 'maxlength'=>'3', 'size'=>'10')); ?>
      <span class="entradaobligatoria">*</span> Kgs</td>
  </tr>
</table>
```

Fig. 15: Uso de los estilos “encabezadotabla” y “entradaplana”.

- Los **textarea** de tipo rich deben tener size=1x10.

```
<tr>
  <td class="verdeclaro">
    <strong><h5>Observaciones del doctor</h5></strong>
    <div align="center"><?php echo textarea_tag('comentario','',
      'rich=true size=1x10 tinymce_options=language:"es"' );?>
    </div>
  </td>
</tr>
```

Fig. 16: Muestra de código para establecer la dimensión de 1x10 al “**textarea**”.



Fig. 17: “**Textarea**” con dimensión 1x10.

- La relación que existe entre el tamaño del nombre de campo, el componente y la ubicación de los mismo es la siguiente:

- Nombre de campo corto y componente largo (ejemplo textarea): poner nombre de campo encima del componente y alineado a la izquierda.

Localización:

Fig. 18: Diseño para los nombres de campo corto y componente largo.

- Nombre de campo y componente checkbox o listbox deben ir en una misma línea.

Xantomas de los pliegues palmares

Fig. 19: Diseño para los nombres de campo y componente checkbox o listbox.

- Los botones para efectuar operaciones sobre el formulario se ubicarán en el centro o a la derecha del extremo inferior de cada interfaz el formulario. El estilo utilizado para los botones es “sbttn”.

```
<table align="center">
<tr>
<td>
<?php echo submit_tag('Guardar','class="sbttn" type="submit"') ?>
<?php echo button_to('Cancelar', '',array('class' => 'sbttn_cancelar',)) ?>
</td>
</tr>
</table>
```

Fig. 20: Uso del estilo “sbttn” para los botones.

Peso: * Kgs

Talla: * m (Ej:1.82)

Fig. 21: Diseño de los botones en la interfaz “Calcular calorías de la dieta”.

- Para representar campos que deben ser de entrada obligatoria se colocará al lado derecho del componente en el cual el usuario entrará los datos un asterisco con el estilo “entradaobligatoria”.

```

<tr>
  <td class="encabezadotabla"><div align="right">
    <span class="hModuloListado">Peso:</span></div></td>
  <td class="encabezadotabla"><?php echo input_tag('peso','',
    array('title'=>'Introduzca su peso',
  </tr>
  
```

Fig. 22: Uso del estilo “entradaobligatoria”.

Peso: * Kgs

Fig. 23: Textfield con diseño para la entrada obligatoria de datos.

- Los mensajes de error ocurridos durante la validación del formulario se mostrarán en la parte superior del campo validado en el cual ocurrió el error.

Este paciente ya existe

Carnet de identidad: 86112404498 *

Fig. 24: Mensaje de error por campo incorrecto.

- Para tablas donde se solicitan datos de localidad, estos deben estar organizados horizontal o verticalmente pero siempre presentando primero (orden de izquierda a derecha o de arriba hacia abajo) los de mayor extensión, para ir discriminando.

Ejemplo

- País
- Provincia
- Municipio

- Dirección

País: Cuba

Provincia: Villa Clara *

Municipio: Encrucijada *

Dirección particular: calle 45 *

Fig. 25: Diseño de tablas para solicitud datos de localidad.

2.4. Vistas arquitectónicas.

En 1995 Philippe Kruchten propuso su célebre modelo de “4+1” vistas arquitectónicas las cuales fueron vinculadas al Rational Unified Process (RUP). Las vistas muestran los diferentes aspectos del sistema que son modelados. Una vista no es un gráfico, pero es una abstracción consistente de un número de diagramas.

Vistas arquitectónicas:

- **Vista de Casos de Uso:** Muestra la funcionalidad del sistema percibido por actores externos.
- **Vista Lógica:** Muestra como la funcionalidad es diseñada dentro del sistema, define la estructura y el comportamiento del sistema.
- **Vista Concurrente o de Procesos:** Muestra la concurrencia en el sistema dividido en procesos y procesadores. Da cuenta de los aspectos de comunicación e integración.
- **Vista de Despliegue:** Muestra la arquitectura física del sistema.
- **Vista de Componentes o Implementación:** Muestra la organización de componentes del código y su implementación.

No todas las arquitecturas de software requieren todas las vistas del modelo 4+1, algunas vistas pueden ser omitidas, pero todos los escenarios se pueden encontrar en ellas. La finalidad de estas

vistas es, unificar los criterios de modelado, la necesidad de facilitar la comunicación entre los diseñadores y establecer estándares variados que permitan una mayor comprensión del sistema que se esté modelando.

2.5. Vista Lógica.

La Vista Lógica describe el diseño más importante de las clases y su organización en paquetes y subsistemas, y la organización de éstos en capas. Es representada por uno o varios diagramas de clases que son un subconjunto del modelo de diseño. Esta muestra cómo la funcionalidad es diseñada en el interior del sistema, en términos de la estructura estática y comportamiento dinámico del sistema. Es un subconjunto del Modelo de Casos de Uso y su realización es obligatoria.

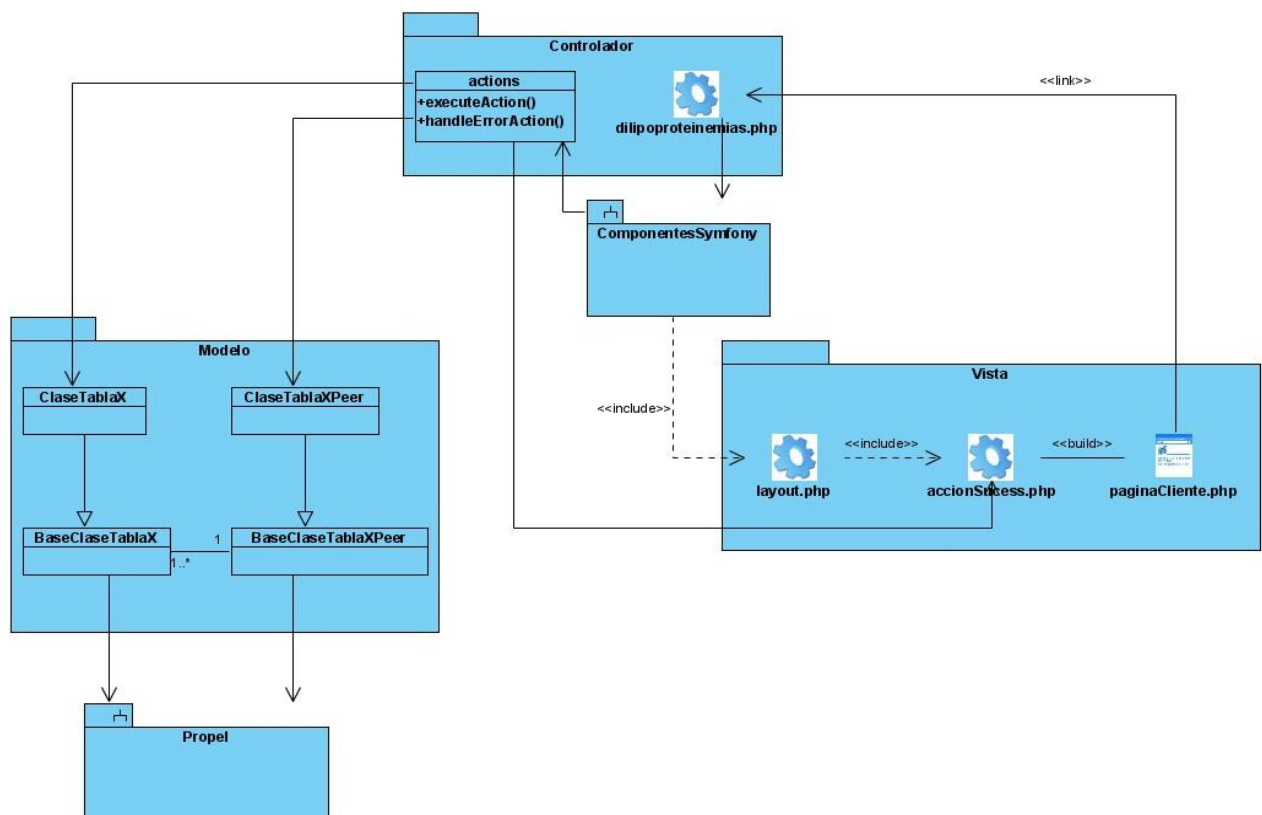


Fig. 26: Representación de la Vista Lógica.

2.5.1. Diagramas de clases del diseño.

Un diagrama de clases del diseño representa las clases del diseño y sus objetos, así como los subsistemas del diseño.

A continuación se describen cada uno de los paquetes y subsistemas que conforman cada uno de los diagramas de diseño del sistema.

Paquete controlador: Una parte importante de su trabajo es común a todos los controladores de la aplicación. Entre las tareas comunes se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador se ha dividido en un controlador frontal (**dislipoproteinemias.php**), que se encarga de realizar las tareas comunes y las acciones (**actions.php**), que incluyen el código específico del controlador de cada página. Habrá un actions.php por cada uno de los módulos pero el controlador frontal será el mismo para todo el sistema.

Paquete vista: Las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabeceras de la página, el layout genérico, el pie de página y la navegación global. En la mayor parte de las veces sólo cambia el interior de la página. Por este motivo, la vista se separa en un **layout** y en una plantilla (el nombre de las plantillas está compuesto por el nombre de la acción que la origina seguido por sufijo que puede ser SUCCESS o ERROR). El layout será global en toda la aplicación o a la mayoría de las páginas. La plantilla sólo se encarga de visualizar las variables definidas en el paquete del controlador.

Paquete del modelo: Solo contiene las clases encargadas del acceso a los datos almacenados en el gestor de base de datos, las cuales utilizan el ORM (Object Relational Model) Propel para el acceso a los mismos.

En función de lograr una mejor comprensión de los diagramas de clases de diseño que a continuación se mostrarán se hace necesario una serie de aclaraciones:

- Las clases **BaseClase** y **BaseClasePeer**, tienen una relación de asociación sin navegabilidad porque ambas pueden crear instancias una de la otra. La multiplicidad expresa que una instancia de la clase **BaseClasePeer** puede crear 1 o varias instancias de la clase **BaseClase** cuando se ejecuta.
- Los estereotipos de **ServerPage** se utilizan para las páginas web que tienen lógica de negocio para diferenciarlas de las clases “claseActions” que no son páginas web sino clases con extensión .php.
- La relación de dependencia estereotipada `<<include>>` entre algunas clases está justificado precisamente por los elementos propios de la programación en PHP, una clase incluye el código de la otra.
- Se usan paquetes para distribuir las clases según el patrón Modelo-Vista-Controlador.

Subsistema Propel: Su función es gestionar el acceso a la base de datos y gestionar el modelo del sistema. Implica que el acceso y la modificación de los datos almacenados en la base de datos se realicen mediante objetos, nunca de forma explícita, permitiendo un alto nivel de abstracción y fácil portabilidad. Propel tiene incluido tareas para generar automáticamente las sentencias SQL necesarias para crear las tablas de la base de datos. La librería Propel se encarga de esta generación automática, ya que crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario. La abstracción de la base de datos es completamente transparente para el programador, ya que se realiza de forma nativa mediante PDO (PHP Data Objects).

Subsistema ComponenteSymfony: Representa todas las clases del framework Symfony que serán utilizadas durante el funcionamiento del sistema. Dígase validadores de formularios, helpers de objetos y formularios, plantillas, componentes de seguridad, etc.

A continuación se muestra el diagrama de casos del diseño correspondiente al caso de uso “**Solicitar Traslado**”. Para el estudio de los diagramas de clase del diseño del sistema remitirse al **Anexo 2** “Diagramas de clase del diseño”.

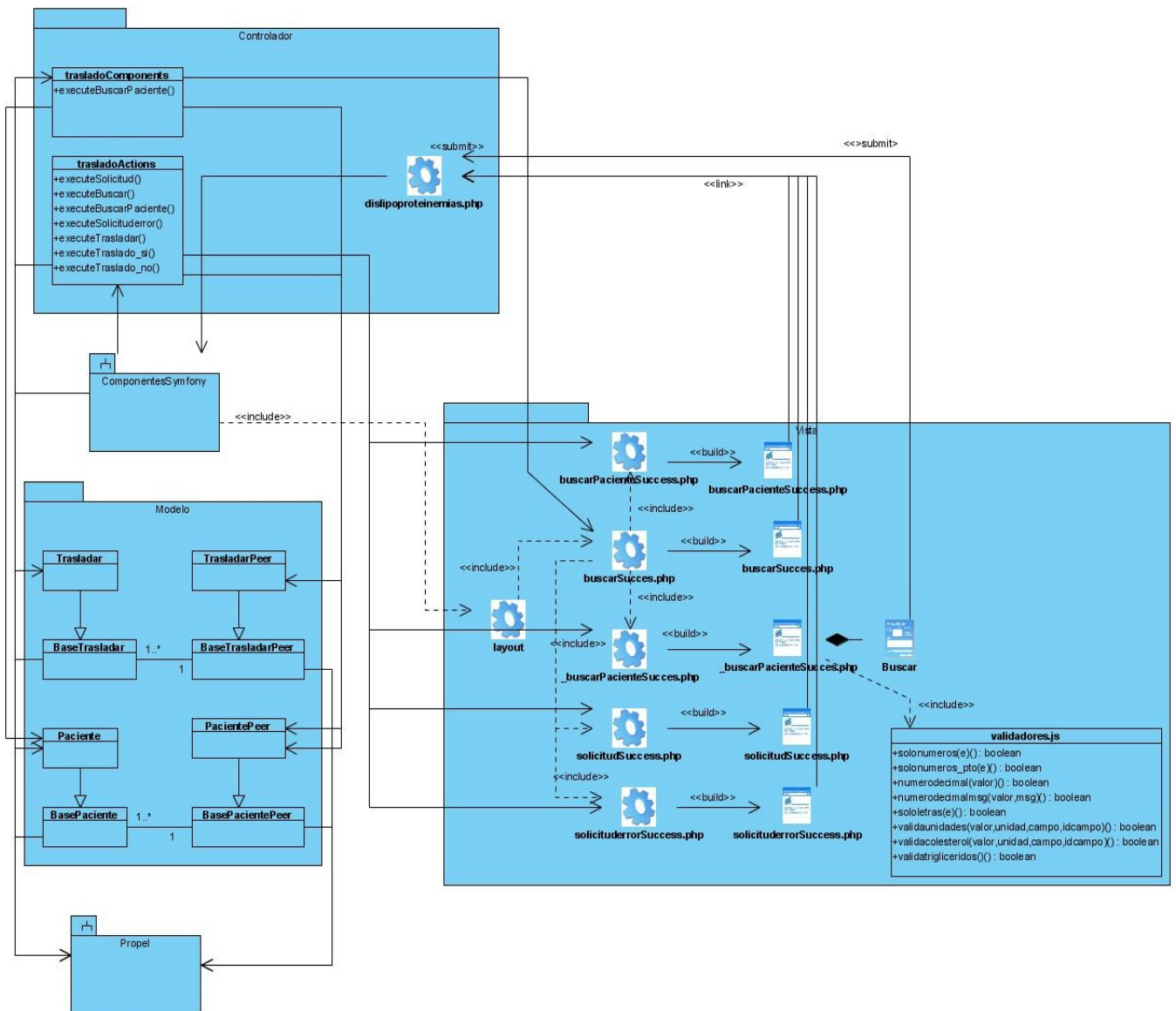


Fig. 27: Diagrama de clase de diseño: Caso de uso “Solicitar Traslado Paciente”.

2.5.2. Descripción de las clases de diseño.

A continuación se muestra la descripción de cada una de las funcionalidades de la clase “**trasladoActions**”. Para el estudio de las descripciones de las clases que contienen la lógica de

negocio del sistema remitirse al **Anexo 3** “Descripción de las clases que contienen la lógica de negocio del sistema”.

Nombre de la clase: trasladoActions
Nombre del método: executeSolicitud().
Descripción: Este método permite a los médicos realizar una solicitud de traslado para cada paciente. Primero se verifica que el paciente no tenga una solicitud pendiente en la base de datos, en el caso de presentarla se muestra un mensaje informando de su solicitud, en caso contrario se crea la solicitud de traslado.
Nombre del método: executeBuscar().
Descripción: Se encarga de ejecutar la página “buscar” del módulo traslado.
Nombre del método: executeBuscarPaciente().
Descripción: Este método permite a los médicos buscar los pacientes existentes en la base de datos según los criterios de búsqueda: nombre, primer apellido, segundo apellido, carnet de identidad y muestra los pacientes que pertenezcan al médico que realiza la búsqueda que coincidan con estos criterios.
Nombre del método: executeSolicituderror().
Descripción: Se encarga de ejecutar la página “Solicituderror” del módulo traslado.
Nombre del método: executeTrasladar().
Descripción: Este método solo podrá ser ejecutado por los administradores del sistema, los cuales tienen tres tipos de niveles, nivel nacional, provincial y municipal. El método trasladar realiza una búsqueda de todos los pacientes en la base de datos en dependencia del nivel del administrador el cual solo podrá ver los pacientes a los cuales se le ha solicitado traslado y que pertenecen a su

nivel.
Nombre del método: executeTraslado_si().
Descripción: Este método permite al administrador trasladar al paciente cambiando en la tabla “Paciente” el código del médico anterior por el código del médico solicitante, elimina la solicitud de traslado y luego redirecciona para la página de traslado.
Nombre del método: executeTraslado_no().
Descripción: Este método permite al administrador rechazar la solicitud de traslado realizada al paciente, elimina la solicitud y redirecciona para la página de traslado.

Tabla 17: Descripción de la clase “**trasladoActions**”

2.5.3. Diagramas de interacción.

Los diagramas de interacción muestran en detalle a los objetos y mensajes entre objetos de un determinado escenario para un caso de uso con el objetivo de modelar el comportamiento dinámico del sistema y verificar la coherencia del sistema validándolo con el modelo de clases.

Los diagramas de interacción se expresan de dos formas:

- Diagrama de colaboración: Diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.
- Diagrama de secuencia: Diagrama de interacción que destaca el orden temporal de los mensajes.

2.5.4. Diagramas de secuencia.

Los diagramas de secuencia son uno de los diagramas más efectivos para modelar la interacción entre objetos en un sistema. Contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar estos escenarios. Muestra los objetos como líneas de vida a lo

largo de la página y con sus interacciones en el tiempo representadas como mensajes, dibujados como flechas desde la línea de vida origen hasta la línea de vida destino.

Al ser una aplicación web se requerirá del uso de una extensión de UML para el modelado de este tipo de sistemas, con el estereotipo “Server Page” para representar la página web que tiene código que se ejecuta en el servidor; “client page”, que representa una página web con formato HTML; y “form”, como el grupo de elementos de entrada que son parte de una página cliente.

A continuación se mostrara el diagrama de secuencias correspondiente al caso de uso “trasladar”. Para el estudio de los diagramas de secuencia remitirse al **expediente de proyecto donde se encuentran publicados los** diagramas de secuencia.

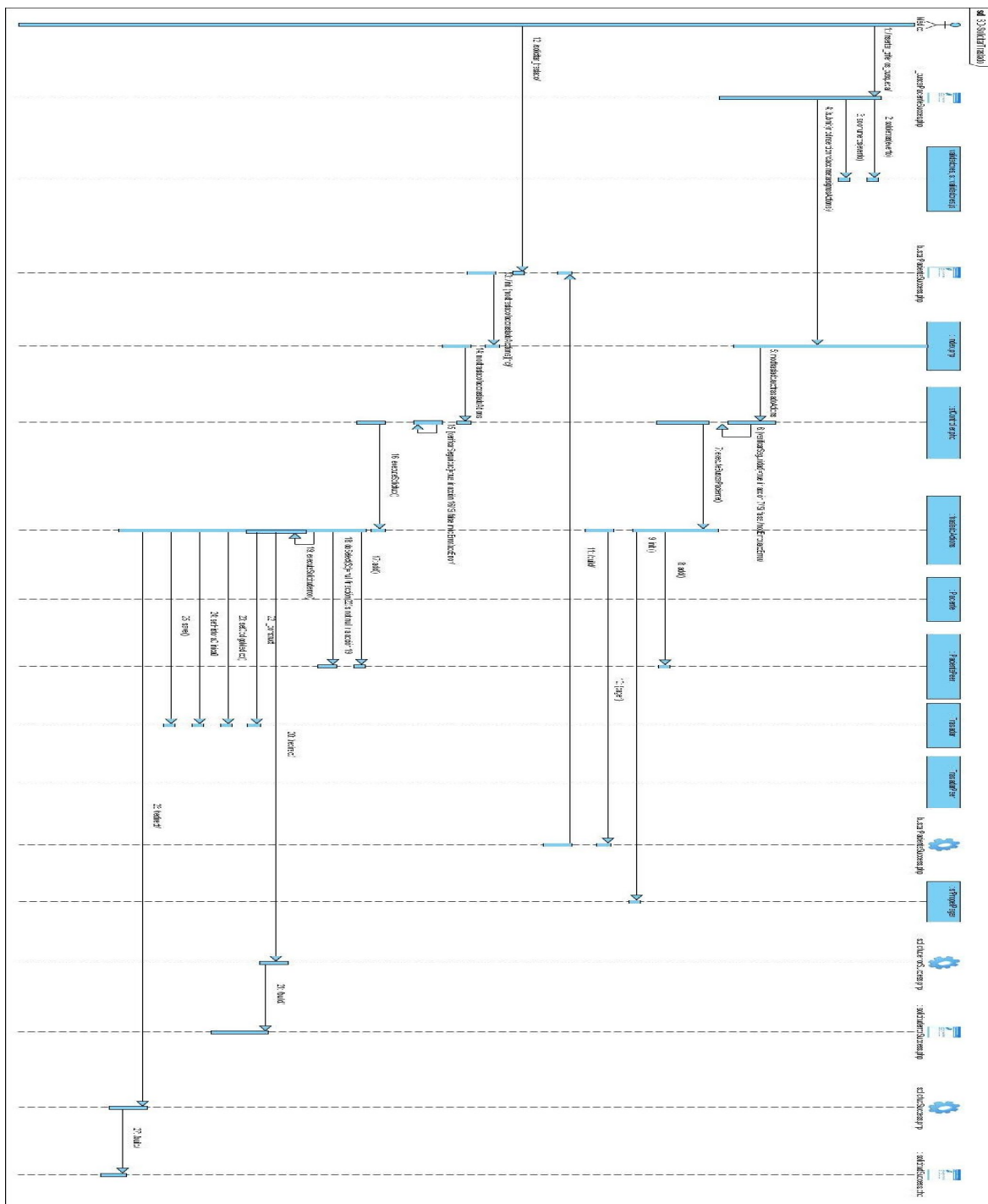


Fig. 28: Diagrama de secuencia para el caso de uso “Solicitar Traslado”.

2.5.5. Diagrama de clases persistentes.

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Las clases persistentes referencian directamente las entidades lógicas y sus atributos. Para definir las clases persistentes se aplicaron las siguientes reglas:

- Cuando una clase que está formada por otras clases es persistente, automáticamente las clases componentes también son persistentes. Lo contrario no se cumple necesariamente.
- Cuando una clase hija de una jerarquía es persistente, automáticamente son persistentes sus ancestros en el árbol de jerarquía. Lo contrario no se cumple necesariamente.
- Cuando se define como persistente a una clase que agrupa a objetos de un mismo tipo de clase base (se refiere a las clases listas, colecciones, registros), entonces automáticamente son persistentes todas las clases hijas a partir de la clase base, incluyendo a la clase base.
- Cuando hay herencia múltiple, esta debe ser resuelta antes si el medio de almacenamiento ha utilizar no soporta este concepto. La solución más factible es que la clase hija herede de la clase, de la que redefine sus métodos y añada un atributo pasivo del tipo de la otra clase de la que heredaba. Si se redefinía comportamiento de más de una clase padre, hay que escoger de cual se quedaría heredando y añadir un atributo pasivo por cada una de las clases padres de las que no hereda. Los métodos que se redefinen que ya no se reciben por herencia, en su implementación incluirán las relaciones con las clases padres.

El siguiente diagrama de clases persistentes que a continuación se muestra fue generado aplicando ingeniería inversa a partir del modelo de datos , debido a que el diseño e implementación se realizaron paralelamente, por que a petición del cliente y la facultad se necesitaba obtener un resultado con premura.

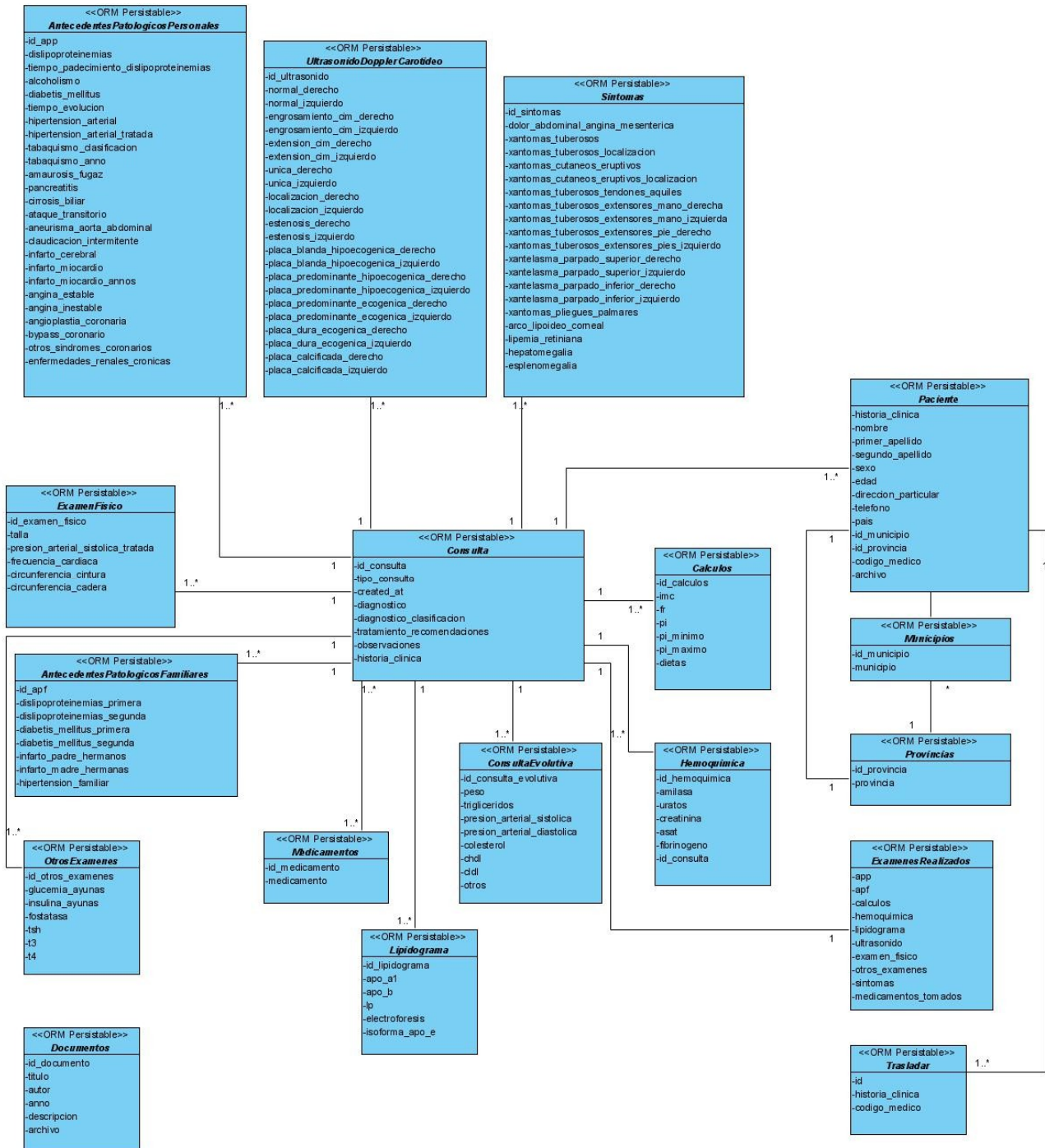


Fig. 29: Diagrama de clases persistentes.

2.5.6. Modelo de datos.

El modelo de datos describe la representación lógica y física de la persistencia de los datos utilizados por la aplicación. En los casos en que la aplicación utilizara un sistema de gestión de base de datos relacional (sus siglas en inglés RDBMS), el modelo de datos también puede incluir procedimientos almacenados, disparadores, restricciones etc. que definen la interacción de los componentes de aplicación.

El modelo de datos estará compuesto por las entidades que pasarán a ser las tablas de la base de datos que será utilizada por el sistema. Las entidades de color naranja se corresponden con las clases persistentes definidas anteriormente, las de color verde son las que fueron agregadas por el plugin sfGuardPlugin para gestionar la seguridad del sistema y por el plugin sfSimpleForumPlugin para el almacenamiento de la información propia del foro de discusión.

2.5.7. Descripción de las tablas.

A continuación se muestra la descripción de las tablas:

- antecedentes_patologicos_familiares
- antecedentes_patologicos_personales
- calculos

Para el estudio de la descripción de las restantes tablas de la base de datos remitirse al **Anexo 4** “Descripción de las tablas de la base de datos”.

antecedentes_patologicos_familiares		
Descripción: Tabla que almacena los datos asociados a los antecedentes patológicos familiares de un paciente.		
Atributo	Tipo	Descripción
id_apf	Int	Identificador de la tabla “antecedentes_patologicos_familiares”.
dislipoproteinemias_primera	Int	Dislipoproteinemias (Familiares 1ra línea).
dislipoproteinemias_segunda	Int	Dislipoproteinemias (Familiares 2da línea).
diabetis_mellitus_primera	Int	Diabetes Mellitus (Familiares 1ra línea).
diabetis_mellitus_segunda	Int	Diabetes Mellitus (Familiares 2da línea).
infarto_padre_hermanos	Int	Infarto del corazón o muerte súbita (Padre o hermanos menores 55 años).
infarto_madre_hermanas	Int	Infarto del corazón o muerte súbita (Madre o hermanas menores 65 años).
hipertension_familiar	Int	Hipertensión arterial (Familiares 1ra línea).

id_consulta	Int	Identificador de la tabla “consulta” (llave foránea).
-------------	-----	---

Tabla 18: Descripción de la tabla “**antecedentes_patologicos_familiares**”

antecedentes_patologicos_personales		
Descripción: Tabla que almacena los datos asociados a los antecedentes patológicos personales de un paciente.		
Atributo	Tipo	Descripción
id_app	int	Identificador de la tabla “antecedentes_patologicos_personales”.
dislipoproteinemias	int	Dislipoproteinemias (Si o No).
tiempo_padecimiento_dislipoproteinemas	int	Tiempo de padecimiento (años).
alcoholismo	int	Ingestión de bebidas alcohólicas (Nunca, Diaria o Semanal, Ocasional).
diabetis_mellitus	int	Diabetes Mellitus (No, Tipo 1, Tipo 2).
tiempo_evolucion	int	Tiempo de evolución (años).
hipertension_arterial	int	Hipertensión arterial (¿Desde cuando?) (años).
hipertension_arterial_tratada	int	Tratamiento actual (Si o No).
tabaquismo_clasificacion	int	Tabaquismo (Nunca fumó, Ex fumador, Fumador actual).
tabaquismo_anno	int	Tiempo fumando (años).

amaurosis_fugaz	int	Otros (Amaurosis Fugaz).
pancreatitis	int	Otros (Pancreatitis).
cirrosis_biliar	int	Otros (Cirrosis biliar).
ataque_transitorio	int	Otros (Ataque transitorio de isquemia).
aneurisma_aorta_abdominal	int	Otros (Aneurisma de la aorta abdominal).
claudicacion_intermitente	int	Otros (Claudicación intermitente).
infarto_cerebral	int	Otros (Infarto cerebral isquémico).
infarto_miocardio	int	Cardiopatía isquémica (Infarto del miocardio).
infarto_miocardio_anno	int	Cardiopatía isquémica (Año(s) del primero).
angina_estable	int	Cardiopatía isquémica (Angina estable).
angina_inestable	int	Cardiopatía isquémica (Angina inestable).
angioplastia_coronaria	int	Cardiopatía isquémica (Angioplastia coronaria).
bypass_coronario	int	Cardiopatía isquémica (Bypass coronario).
enfermedades_renales_cronicas	int	Cardiopatía isquémica (Enfermedades renales crónicas).

otros_sindromes_coronarios	entero	Cardiopatía isquémica (Otros síndromes coronarios).
id_consulta	int	Identificador de la tabla “consulta” (llave foránea).

Tabla 19: Descripción de la tabla “**antecedentes_patologicos_familiares**”

Cálculos		
Descripción: Tabla que almacena los datos asociados a los cálculos auxiliares (Índice de Masa Corporal, Peso Ideal, Calorías de la Dietas, Riesgo Cardiovascular) de un paciente.		
Atributo	Tipo	Descripción
id_calculos	int	Identificador de la tabla “calculos”.
imc	Float	Índice de Masa Corporal.
fr	Float	Factor de Riesgo Cardiovascular.
pi	Float	Peso Ideal.
pi_minimo	Float	Peso Ideal Mínimo.
pi_maximo	Float	Peso Ideal Máximo.
dietas	Float	Calorías de la Dieta.
id_consulta	Int	Identificador de la tabla “consulta” (llave foránea).

Tabla 20: Descripción de la tabla “**calculos**”.

2.5.8. Validación teórica.

2.5.8.1. Integridad de los datos.

La integridad de los datos consiste en la corrección y completamiento de los datos en una base de datos ya que en los procesos de inserción, eliminación o actualización pueden haber pérdidas, añadirse datos no válidos, se pueden modificar datos ya existentes por valores incorrectos; pueden perderse datos también debido a un error del sistema o a un fallo en el suministro de energía o por que estos cambios son aplicados parcialmente. El mantenimiento de la integridad, junto con el de la confidencialidad y disponibilidad, constituye el objetivo principal de la seguridad de la información.

En el diseño de las tablas es muy importante la identificación de valores válidos para una columna y la determinación de cómo forzar la integridad de los datos en la columna. La integridad de datos se divide en 4 categorías integridad de entidad, integridad de dominio, integridad referencial integridad definida por el usuario, para el desarrollo de la aplicación se aplicaron 3 de ellas, las cuales se describen detalladamente a continuación:

- **Integridad de entidad.**

La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones UNIQUE, o restricciones PRIMARY KEY.” [24].

Todas las tablas de la base de datos tienen definido una fila como identificador usando la restricción PRIMARY KEY. Como ejemplo de lo anteriormente planteado se puede evidenciar que las tablas antecedentes_patologicos_familiares, antecedentes_patologicos_personales y calculos tienen como PRIMARY KEY los atributos antecedentes_patologicos_familiares, antecedentes_patologicos_familiares y calculos correspondientemente.

consulta		
+id_consulta	int	Nullable = false
tipo_consulta	int	Nullable = false
created_at	datetime	Nullable = true
diagnostico	text	Nullable = false
diagnostico_clasificacion	varchar(30)	Nullable = false
tratamiento_recomendaciones	text	Nullable = false
observaciones	text	Nullable = true

Fig. 31: Tabla donde se evidencia la definición de PRIMARY KEY.

- **Integridad de dominio.**

“La integridad de dominio viene dada por la validez de las entradas para una columna determinada. Puede exigir la integridad de dominio para restringir el tipo mediante tipos de datos, el formato mediante reglas y restricciones CHECK, o el intervalo de valores posibles mediante restricciones FOREIGN KEY, restricciones CHECK, definiciones DEFAULT, definiciones NOT NULL y reglas.” [24].

Los atributos de cada una de las tablas presentan un tipo de dato definido en correspondencia con el valor que deben almacenar cada uno de estos. Los tipos de datos definidos son int, varchar, float, datetime, text entre otros tipos. Si el valor del atributo no puede ser vacío se establece como definición NOT NULL. En caso de que una tabla presente multiplicidad de uno a muchos con otra, entonces el atributo definido como PRIMARY KEY de la tabla con multiplicidad uno pasaría a formar parte como FOREIGN KEY de la tabla con multiplicidad mucho. La restricción FOREIGN KEY se evidencia también cuando existe multiplicidad de uno a uno donde la PRIMARY KEY de una de las tablas pasaría a formar FOREIGN KEY de la otra tabla.

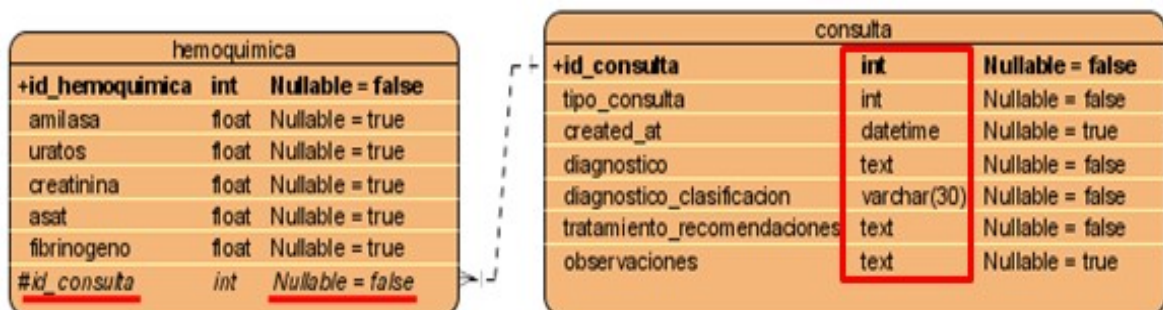


Fig.32: Uso de los diferentes tipos de datos, de la definición NOT NULL, y la restricción FOREIGN KEY.

- **Integridad referencial.**

“La integridad referencial protege las relaciones definidas entre las tablas cuando se crean o se eliminan filas.” [24].

“La integridad referencial garantiza que los valores de clave sean coherentes en las distintas tablas. Para conseguir esa coherencia, es preciso que no haya referencias a valores inexistentes y que, si cambia el valor de una clave, todas las referencias a ella se cambien en consecuencia en toda la base de datos.” [24].

Durante el proceso de inserción y eliminación en las tablas se verifica que los campos que pertenecen a las relaciones entre dicha tabla y alguna otra de la base de datos sean válidos.

2.5.8.2. Normalización de la base de datos.

La normalización consiste en transformar datos complejos en un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. Este proceso se puede entender también como una serie de reglas que se aplican a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional, con el objetivo de evitar la redundancia de los datos, los problemas de actualización e integridad de los datos en las tablas y lograr disminuir su uso de espacio en disco.

Durante el proceso de normalización de la BD se realizaron una serie de transformaciones de forma gradual, garantizando con las mismas que ocurrieran la menor cantidad de anomalías de actualización. A continuación se muestran ejemplos de transformaciones al realizar el análisis correspondiente a cada una de las formas normales por las que iba transitando la base de datos.

“**Primera Forma Normal:** (1FN).

- Prohíbe los atributos multivaluados, los atributos compuestos y sus combinaciones.

- Prohíbe tener un conjunto de valores, una tupla de valores o una combinación de ambas como valor de un atributo para una tupla individual.
- Prohíbe las “relaciones dentro de relaciones” o “relaciones como atributos de tuplas”.
- Sólo se permiten valores atómicos.
- Prohíbe atributos compuestos que por sí mismos son multievaluados o sea relaciones anidadas.

En la relación **Consulta** el atributo medicamentos es un atributo multievaluado y esto viola los requerimientos para alcanzar la primera forma normal. Para solucionar esto debemos crear dos nuevas relaciones **Consulta** y **Medicamento** y establecer entre estas dos relaciones una multiplicidad de mucho a mucho generando una relación **Consulta_medicamento** con el identificador de consulta de medicamento como llave primaria de esta relación:

Consulta(id_consulta, tipo_consulta, created_at, diagnostico, diagnostico_clasificacion, tratamiento_recomendaciones, observaciones, historia_clinica, medicamentos).

Consulta(id_consulta, tipo_consulta, created_at, diagnostico, diagnostico_clasificacion, tratamiento_recomendaciones, observaciones, historia_clinica).

Medicamento(id_medicamento, medicamento).

Consulta_medicamentos(id_consulta, id_medicamento).

Segunda Forma Normal: (2FN).

Para cumplir con la 2FN la base de datos debe encontrarse en 1FN, además debe cumplir con el concepto de dependencia funcional total.

Dependencia funcional total: Sea $X \rightarrow Y$. Si la eliminación de cualquier atributo A de X hace que la dependencia deje de ser válida se dice que esta dependencia es total o que Y depende totalmente de X. O sea $\forall A \in X, (X - \{A\})$ no determina a Y, entonces la dependencia es total (los atributos no llaves dependen de todos los atributos que componen la llave y no de solo una parte de ellos).

En la relación **Paciente** se encuentra el identificador de consulta que posibilita que existan dependencias parciales por lo que una posible solución es crear dos nuevas relaciones Paciente (llave

primaria identificador de paciente(historia_clínica)) y Consulta(llave primaria identificador de consulta(id_consulta)).

Paciente(historia_clinica, nombre, primer_apellido, segundo_apellido, sexo, edad, direccion_particular , telefono, pais, id_provincia, id_municipio, codigo_medico, archivo, id_consulta, tipo_consulta, created_at, diagnostico, diagnostico_clasificacion, tratamiento_recomendaciones, observaciones.).

Paciente(historia_clinica, nombre, primer_apellido, segundo_apellido, sexo, edad, direccion_particular , telefono, pais, id_municipio, id_provincia, codigo_medico, archivo).

Consulta(id_consulta, tipo_consulta, created_at, diagnostico, diagnostico_clasificacion, tratamiento_recomendaciones, observaciones.).

Tercera Forma Normal: (3FN).

Para cumplir con la 3FN la base de datos debe encontrarse en 2FN y eliminar las dependencias transitivas.

Dependencia Transitiva: sea $X \rightarrow Y$ una dependencia funcional en un esquema de relación R, se dice que es transitiva si existe un conjunto de atributos Z que no sea un subconjunto de cualquier llave de R y se cumple tanto $X \rightarrow Z$ como $Z \rightarrow Y$ ($X \rightarrow Y$ donde Y no primo, X ni subconjunto ni superconjunto de alguna llave)."[25].

En la relación **Paciente** anteriormente mencionada existen dos identificadores (id_provincia, id_municipio) que evidencian la existencia de dependencias funcionales transitivas, por lo que la solución es eliminar de la relación **Paciente** la dependencia con la relación **Municipio**.

Paciente(historia_clinica, nombre, primer_apellido, segundo_apellido, sexo, edad, direccion_particular , telefono, pais, id_provincia, codigo_medico, archivo, id_consulta, tipo_consulta, created_at, diagnostico, diagnostico_clasificacion, tratamiento_recomendaciones, observaciones.).

La base de datos que se obtuvo para la realización del sistema se encuentra en 3FN.

2.6. Vista de despliegue.

La vista de despliegue describe varios nodos físicos para las configuraciones más típicas de las plataformas y la asignación de las tareas de la Vista de Procesos a los nodos físicos. Se obtiene cuando se realiza el flujo de trabajo de Análisis y Diseño. Describe la situación de los componentes en una posible implantación del sistema de acuerdo a los requisitos iniciales. Esta vista se realiza sólo si el sistema es distribuido a través de más de un nodo.

El modelo de despliegue que a continuación se muestra fue propuesto y aprobado por el grupo de arquitectura de Softel.

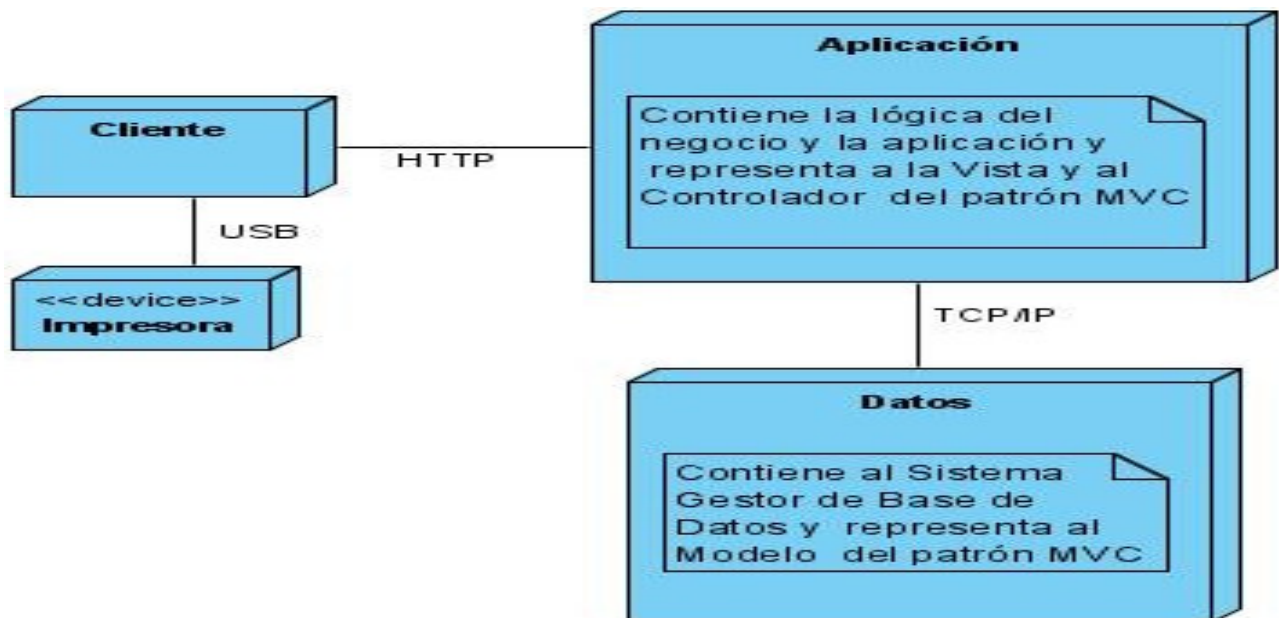


Fig.33: Modelo de despliegue.

2.6.1. Descripción de los nodos físicos.

Un nodo es un objeto físico que representa un recurso informático, este recurso generalmente dispone de datos persistentes y capacidad de proceso. Las conexiones entre nodos muestran las líneas de comunicación con las que el sistema tendrá que interactuar.

En el nodo **Ciente** se encontrará el sistema operativo Windows o Linux y los navegadores web Mozilla Firefox o Internet Explorer mediante los cuales los clientes tendrán acceso al sistema.

En el nodo **Aplicación** estarán agrupados los archivos a través de los cuales el usuario logra acceder al sistema, además se encuentra contenida toda la información específica de cada registro, sus clases; así como almacena además la configuración general del proyecto, las clases y librerías externas, todo el código común de las aplicaciones del proyecto y los plugins de instalación de la aplicación.

El nodo **Impresora** es un dispositivo externo conectado al nodo cliente usado para la impresión de documentos.

En el nodo **Datos** serán almacenados los datos de la aplicación, se guardará el modelo de objetos del sistema.

2.6.2. Características físicas de la conexión.

<<Conexión TCP/IP>>: TCP/IP es el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre sí. TCP/IP, se encargará de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware. Protocolos como TCP e IP proporcionan las reglas para la comunicación, contienen los detalles referentes a los formatos de los mensajes, describen como responde una PC cuando llega un mensaje y especifican de qué manera una PC maneja un error u otras condiciones anormales.

<<Conexión HTTP>>: Protocolo de transferencia de hipertexto (HTTP, Hyper Texto Transfer Protocol (Protocolo de transferencia de Hyper Texto)) es el protocolo usado en cada transacción de la Web (WWW). Este protocolo se puede entender como el lenguaje utilizado por dos computadoras para

comunicarse entre sí. Hyper Texto se refiere al contenido de las páginas escrito en un lenguaje especial (HTML), los browser (navegadores) se comunican con los servidores de internet mediante este protocolo se envían las páginas en el lenguaje HTML y lo interpretan y nos muestran su contenido.

<<**Conexión USB**>>: Protocolo para la conexión entre la estación de trabajo y el dispositivo impresora.

2.7. Conclusiones.

En este capítulo se obtuvieron los artefactos del diseño de la solución. Se realizaron las clases del diseño con estereotipos Web, los diagramas de secuencia, el diseño de la Base de Datos y el diagrama de despliegue que muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Los anteriores elementos servirán de entrada para el flujo de implementación.

Capítulo 3: Implementación del sistema.

Introducción.

En el presente capítulo se presentan los diagramas de componentes elaborados a partir de los diagramas de clases de diseño. Se muestra el código fuente de uno de los principales métodos con la descripción del mismo. Se argumenta sobre la validación y seguridad aplicada a la aplicación. Además se presentan las interfaces de la aplicación más significativas de alasLIPO v2.0.

3.1. Vista de implementación.

La Vista de implementación contiene la organización de los módulos en términos de paquetes y capas, pueden incluirse también la trazabilidad de la vista lógica. Esta vista toma en cuenta los requerimientos que facilitan la programación, los niveles de reutilización y las limitaciones impuestas por el entorno de desarrollo. Es representada por un diagrama de componentes o especificaciones de paquetes que son básicamente un subconjunto del modelo de despliegue. Para modelarla se dispone de dos elementos, los paquetes que representan una partición física del sistema y los componentes que representan la organización de los módulos de código fuente.

Symfony establece una estructura de carpetas para sus proyectos y almacena los archivos del proyecto en una estructura estandarizada de tipo árbol. Algunos subdirectorios agrupan componentes propios de Symfony, que son clases que implementan el núcleo del framework, y por tanto no se modifican en la construcción del sistema. Dentro de un proyecto, las operaciones se agrupan de forma lógica en aplicaciones. Cada aplicación está formada por uno o más módulos.

A continuación se muestra la Vista de Implementación del sistema y la descripción de cada uno de los elementos que la conforman:

- **“apps:** Contiene un directorio por cada aplicación del proyecto.
- **lib:** Almacena las clases y librerías externas, Se suele guardar todo el código común a todas las aplicaciones del proyecto. El subdirectorio model/ guarda el modelo de objetos del proyecto.

- **config:** Almacena la configuración general del proyecto.
- **web:** Contiene los únicos archivos accesibles desde Internet. Los únicos archivos accesibles desde Internet son los que se encuentran en este directorio.
- **pluggins:** Almacena los plugins instalados en la aplicación.” [7].

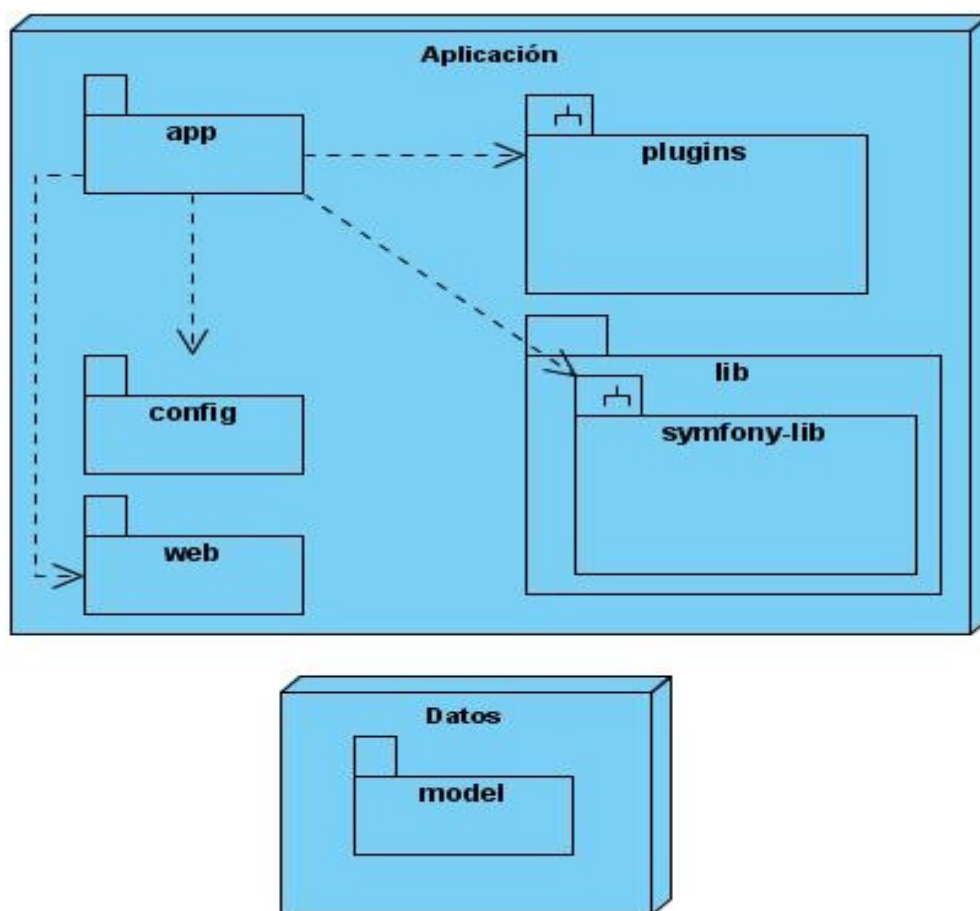


Fig. 34: Vista de Implementación.

La vista de implementación del sistema queda constituida por un grupo de componentes, y subsistemas de implementación que modelan el empaquetado físico real del sistema, estos representan a las carpetas del directorio del proyecto descritas anteriormente, y contienen a los ficheros modelados en término de componentes.

3.2. Diagrama de componentes.

Los diagramas de componentes representan cómo un sistema es dividido en componentes y muestra las dependencias entre estos. Los componentes físicos incluyen archivos, cabeceras, librerías compartidas, módulos, ejecutables, o paquetes. Pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Debido a que estos son muy parecidos a los diagramas de casos de usos son utilizados para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

A continuación se muestra el diagrama de componentes correspondiente al caso de uso “**Solicitar Traslado**” y la descripción de los principales elementos que lo conforman. Para el estudio de los diagramas de secuencia remitirse al **Anexo 5** “Diagramas de componentes”.

Paquete Vista: Paquete que agrupa todos los componentes relacionados con la vista de la aplicación tales como:

- **Componente layout.php:** Componente que implementa la clase layout del diseño. Contiene los elementos que se muestran de forma idéntica a lo largo de toda la aplicación.
- **Componentes (Success):** Componentes que implementan el código correspondiente a cada plantilla que utiliza el módulo.

Paquete Controlador: Paquete que agrupa todos los componentes `actions.class.php` de cada uno de los módulos y el componente `dislipoproteinemias.php`, estos son implementados por las clases `Actions` del diseño. Estos componentes incluyen el código específico del controlador para cada página del módulo.

Componente security.yml: Archivo de configuración que permite restringir el acceso a determinadas acciones del módulo.

Paquete Modelo: Se especifican las clases generadas por el subsistema Propel que es el ORM que utiliza Symfony el cual proporciona persistencia para los objetos y un servicio de consultas. Propel a su vez utiliza el componente Creole como sistema de abstracción de la base de datos, sistema similar a

los PDO (PHP Data Object) y proporciona una interfaz entre el código PHP y el código SQL de la base de datos, permitiendo cambiar fácilmente de sistema gestor de base de datos.

Componente BD (alasLIPO): Encapsula todos los datos del sistema.

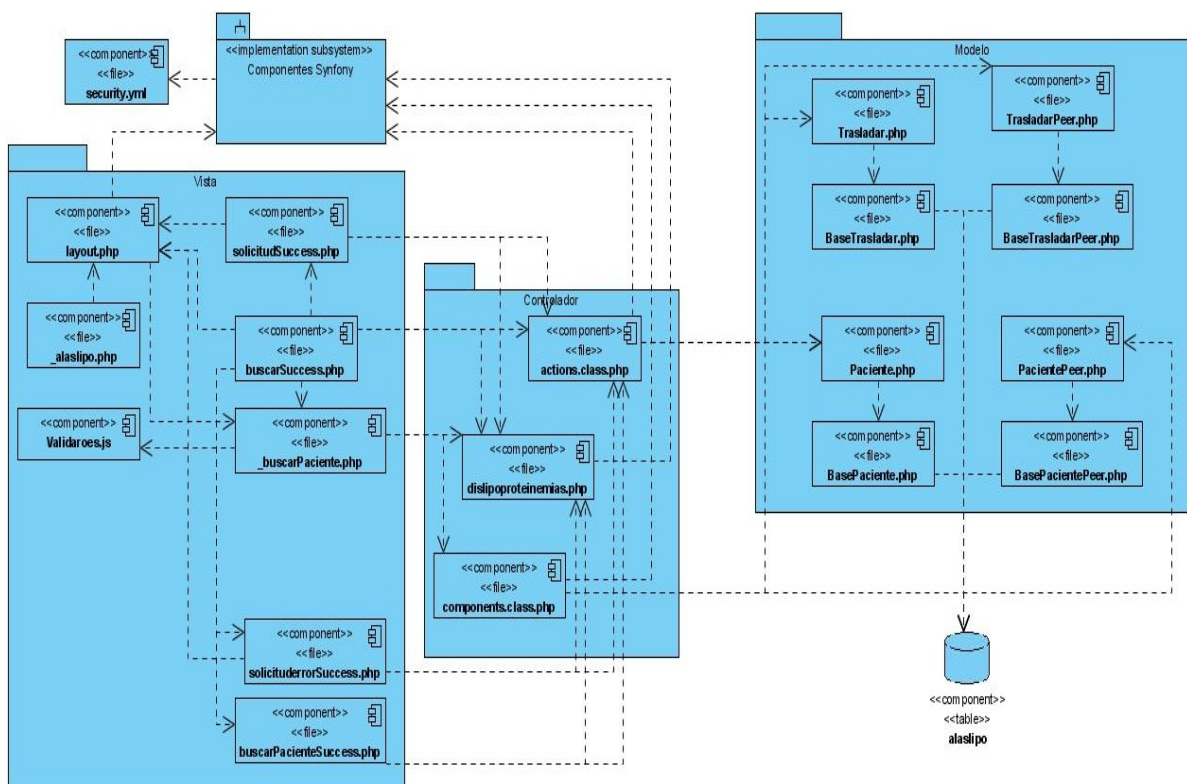


Fig. 35: Diagrama de componentes “Solicitar Traslado”.

3.3. Código fuente.

A continuación se muestra el código fuente de uno de los principales métodos del sistema “executeDiagnostico_comp ()”. Este método es el encargado de obtener el diagnóstico del paciente a partir de los resultados de los exámenes realizados al paciente registrados por el médico, además genera un tratamiento farmacológico y no farmacológico asociado a este diagnóstico.

```

//Generar el diagnóstico y tratamiento de la primera consulta.
public function executeDiagnostico_comp()
(
    $con=$this->getRequestParameter('con');

    $examenesrealizados= new Criteria();
    $examenesrealizados->add(ExámenesRealizadosPeer::ID_CONSULTA,$con);
    $ex = ExámenesRealizadosPeer::doSelect($examenesrealizados);
    $ultrasonido=$ex[0]->getUltrasonido();
    $sintomas=$ex[0]->getSintomas();

// búsquedas en la tabla otros exámenes
    $as= new Criteria();
    $as->add(OtrosExámenesPeer::ID_CONSULTA,$con);
    $as=OtrosExámenesPeer::doSelect($as);
    $glucemia=$as[0]->getGlucemiaAyunas();

//búsquedas en la tabla examen físico
    $c = new Criteria();
    $c->add(ExamenFisicoPeer::ID_CONSULTA,$con);
    $ex = ExamenFisicoPeer::doSelect($c);
    $talla = $ex[0]->getTalla();

//búsquedas en la tabla consulta evolutiva
    $cx = new Criteria();
    $cx->add(ConsultaEvolutivaPeer::ID_CONSULTA,$con);
    $ex12 = ConsultaEvolutivaPeer::doSelect($cx);
    $presion = $ex12[0]->getPresionArterialSistolica();
    $presiond= $ex12[0]->getPresionArterialDiastolica();
    $peso = $ex12[0]->getPeso();
    $colesterol= $ex12[0]->getColesterol();
    $trigliceridos=$ex12[0]->getTrigliceridos();
    $cldl=$ex12[0]->getCldl();
    $chdl=$ex12[0]->getChdl();

// búsquedas en la tabla consulta
    $c1 = new Criteria();
    $c1->add(ConsultaPeer::ID_CONSULTA,$con);
    $ex1 = ConsultaPeer::doSelect($c1);
    $hc = $ex1[0]->getHistoriaClinica();

// búsquedas en la tabla paciente
    $c2 = new Criteria();
    $c2->add(PacientePeer::HISTORIA_CLINICA,$hc);
    $pac = PacientePeer::doSelect($c2);
    $edad = $pac[0]->getEdad();
    $sexo = $pac[0]->getSexo();
    $sexo1='femenino';
    if ($sexo==1)
    (
        $sexo1='masculino';
    )

// búsquedas en la tabla Lipidograma
    $c3=new Criteria();
    $c3->add(LipidogramaPeer::ID_CONSULTA,$con);
    $ex2 = LipidogramaPeer::doSelect($c3);
    $isoforma=$ex2[0]->getIsoformaApoE();

// búsquedas en la tabla Antecedentes patológicos personales (APP)
    $c4 = new Criteria();
    $c4->add(AntecedentesPatologicosPersonalesPeer::ID_CONSULTA,$con);
    $ex3 = AntecedentesPatologicosPersonalesPeer::doSelect($c4);
    $fuma = $ex3[0]->getTabaquismoClasificacion();
    $ataque_corazon=$ex3[0]->getInfartoMiocardio();
    $ataque_transitorio_isquemia=$ex3[0]->getAtaqueTransitorio();
    $infarto_cerebral=$ex3[0]->getInfartoCerebral();
    $diabetis=$ex3[0]->getDiabetisMellitus();
    $aneurisma=$ex3[0]->getAneurismaAortaAbdominal();
    $claudicacion=$ex3[0]->getClaudicacionIntermitente();
    $alcoholismo=$ex3[0]->getAlcoholismo();
    $hipertencion=$ex3[0]->getHipertensionArterial();
    $anginae=$ex3[0]->getAnginaEstable();

```



```

$anginaI=$ex3[0]->getAnginaInestable();
$bypass=$ex3[0]->getBypassCoronario();
$angioplastia=$ex3[0]->getAngioplastiaCoronaria();
$otros_coronarios=$ex3[0]->getOtrosSindromesCoronarios();
$infarto_miocardio=$ex3[0]->getInfartoMiocardio();

// búsquedas en la tabla Antecedentes patológicos familiares (APF)
$c5=new Criteria();
$c5->add(AntecedentesPatologicosFamiliaresPeer::ID_CONSULTA,$con);
$ex4=AntecedentesPatologicosFamiliaresPeer::doSelect($c5);
$infarto_madre=$ex4[0]->getInfartoMadreHermanas();
$infarto_padre=$ex4[0]->getInfartoPadreHermanos();
$exten=false;

//verificación si el examen ultrasonido esta realizado
if ($ultrasonido==1)
{
    // búsquedas en la tabla ultrasonido
    $c6=new Criteria();
    $c6->add(UltrasonidoDopplerCarotideoPeer::ID_CONSULTA,$con);
    $ex5=UltrasonidoDopplerCarotideoPeer::doSelect($c6);
    $extenosis=$ex5[0]->getEstenosisDerecho();
    $extenosisI=$ex5[0]->getEstenosisIzquierdo();

    if ($extenosis>50 || $extenosisI>50)
    {
        $exten=true;
    }
}

$xantomas=false;

//verificación si el examen síntomas esta realizado
if ($síntomas==1)
{
    // búsquedas en la tabla síntomas
    $c7= new Criteria();
    $c7->add(SíntomasPeer::ID_CONSULTA,$con);
    $ex6=SíntomasPeer::doSelect($c7);
    $tendon=$ex6[0]->getXantomasTuberososTendonesAquiles();
    $manod=$ex6[0]->getXantomasTuberososExtensoresManoDerecha();
    $manoi=$ex6[0]->getXantomasTuberososExtensoresManoIzquierda();
    $pied=$ex6[0]->getXantomasTuberososExtensoresPieDerecho();
    $piei=$ex6[0]->getXantomasTuberososExtensoresPiesIzquierdo();
    $xantomaspliegues=$ex6[0]->getXantomasPlieguesPalmares();
    $xcutaneoseruptivos=$ex6[0]->getXantomasCutaneosEruptivos();
    $xtuberosos=$ex6[0]->getXantomasTuberosos();
    $arco=$ex6[0]->getArcoLipideoCorneal();

    if ($tendon==1 || $manod==1 || $manoi==1 || $pied==1 || $piei==1)
    {
        $xantomas=true;
    }
}
else
{
    $xantomaspliegues=0;$xcutaneoseruptivos=0;$xtuberosos=0; $arco='No tiene arco lipideo corneal';
}

// búsquedas en la tabla cálculos
$c9= new Criteria();
$c9->add(CalculosPeer::ID_CONSULTA,$con);
$ex9=CalculosPeer::doSelect($c9);
$factor=$ex9[0]->getFr();

$dietas = new Dietas($talla,$peso);
$imc = $dietas->indice_masa_corporal();

```



```

else ($hombre='')
$this->factores_mayores=$comienzo.$fumal.$presion1.$chdl1.$infarto_padre1.$hombre."</br>";

//verificar el tipo de riesgo según los datos buscados
//riesgo elevado
if($diabetis !=0 ||$aneurisma == 1 || $claudicacion== 1 || $infarto_cerebral == 1 ||
$ataque_transitorio_isquemia == 1 || $exten || $factor>20 || $glucemia>7.7 || $anginae==1 ||
$anginal==1 || $bypass==1 || $angioplastia==1 || $otros_coronarios==1 || $infarto_miocardio==1)
{
    $this->texto='</br><strong>El riesgo de cardiopatía isquémica "dura" a
    10 años es elevado por: </strong></br>';
    $clasificacion='alto riesgo';
    if($diabetis ==1 ||$diabetis ==2 )
    {$diabetis1="- Diabetes mellitus.</br>";}else    {$diabetis1='';}
    if($anginae==1 || $anginal==1 || $bypass==1 || $angioplastia==1 || $otros_coronarios==1
    || $infarto_miocardio==1)
    {$coronarios="- Síndromes coronarios.</br>";}else    {$coronarios='';}
    if($aneurisma ==1){$aneurisma1="- Aneurisma de la aorta abdominal.</br>";}
    else {$aneurisma1='';}
    if($claudicacion == 1){$claudicacion1="- Claudicación intermitente.</br>";}
    else {$claudicacion1='';}
    if($ataque_transitorio_isquemia ==1){$ataque_transitorio_isquemial="- Ataque transitorio
    de isquemia.</br>";}
    else
    {$ataque_transitorio_isquemial='';}
    if($infarto_cerebral ==1){$infarto_cerebral1= "- Infarto cerebral.</br>";}
    else {$infarto_cerebral1='';}
    if ($exten){$exten1="- Más del 50% de estenosis carotídea.</br>";}
    else
    {$exten1='';}
    if($glucemia>7.7){$glucemial="- Glucosa en el plasma elevada.</br>";}
    else
    {$glucemial='';}
    if($factor>20){$valor ="- Además porque tiene riesgo de cardiopatía
    isquémica mayor del 20% de acuerdo al score de Framingham";}
    else {$valor='';}
    $this->diag=$diabetis1.$coronarios.$claudicacion1.$aneurisma1.$ataque_transitorio_isquemial.
    $infarto_cerebral1.$exten1.$glucemial.$valor."</br>";
}
else
{
//riesgo moderado
    $contador=0;
    if($fuma==2){$contador=$contador+1;}
    if ($presion>140 && $presiond>90){$contador=$contador+1;}
    if ($chdl<40){$contador=$contador+1;}
    if ($infarto_padre==1){$contador=$contador+1;}
    if ($infarto_madre==1){$contador=$contador+1;}
    if ($edad>45 && $sexo==1){$contador=$contador+1;}
    if ($edad>55 && $sexo==0){$contador=$contador+1;}

    if ($contador>=2 || ($factor<=20 && $factor>=10))
    {
        $this->texto='</br><strong>El riesgo de cardiopatía isquémica "dura" a
        10 años es moderado por presentar dos o más factores de riesgo mayores.
    </strong></br>';
        $clasificacion='riesgo moderado';
        $this->diag='';
    }
}
if($this->texto=="")
{
//riesgo bajo
    $this->diag='';
    $clasificacion='bajo riesgo';
    $this->texto='</br><strong>El riesgo de cardiopatía isquémica "dura" a
    10 años es bajo:</strong>';
}
}

//verificar padecimientos
//verificar existencia de enfermedad hipertrigliceridemia

```

```

$hiper_trigl=false;

if($trigliceridos>150)
{
    if($diabetis || $imc>=24.9 || $alcoholismo==1)
    {
        $hiper_trigl=true;
    }
}

$hiper_familiar=false;
$pntos=0;
if($edad<18 && $cldl>95)
{ $pntos+=1; }
if(($edad>60 && $sexo=='masculino') || ($edad>55 && $sexo=='femenino'))
{
    if($ataque_corazon==1) { $pntos+=2; }
    if($infarto_cerebral==1) { $pntos+=1; }
}
if($xantomas) { $pntos+=6; }
if($arco!='No tiene arco lipoideo corneal' && $edad<45) { $pntos+=4; }

if($cldl>=155 && $cldl<=189) { $pntos+=1; }
if($cldl>=190 && $cldl<=249) { $pntos+=3; }
if($cldl>=250 && $cldl<=329) { $pntos+=5; }
if($cldl>=330) { $pntos+=8; }
if($pntos>=8) { $hiper_familiar=true; }

//verificar existencia de enfermedad disbetalipoproteinemia
$disbeta =false;
if($xantomaspliegues==1 || $xtuberosos==1 || $xcutaneoseruptivos==1 || $colesterol>300 ||
$trigliceridos>400 || $isoforma!=0)
{ $disbeta=true; }

$df="";
$hf="";
$ht="";
$principio="";
$this->enfermedades='';
//respuesta según enfermedad

if($disbeta)
{
    $principio="<br><strong>Debe considerarse alguno de los tipos de dislipoproteinemias
siguientes:</strong><br>";
    $df="- Disbetalipoproteinemia familiar.<br>";
}
if($hiper_familiar && $trigliceridos<200)
{
    $principio="<br><strong>Debe considerarse alguno de los tipos de dislipoproteinemias
siguientes:</strong><br>";
    $hf="- Hipercolesterolemia familiar.<br> ";
}
}
if($hiper_trigl)
{
    $principio="<br><strong>Debe considerarse alguno de los tipos de dislipoproteinemias
siguientes:</strong><br>";
    $ht="- Hipertrigliceridemia familiar.<br> ";
}
$this->enfermedades=$principio.$df.$hf.$ht;

//resultado final del diagnóstico
$final=$this->otros.$this->otros_diag.$this->factores_mayores.$this->texto.$this->diag.
$this->enfermedades;
$this->con=$con;
$this->setFlash('con', $con);

/*****TRATAMIENTO *****/
/*****

```

```

//*****Tratamiento no farmacologico*****
    $this->firt="<strong>El paciente debe:</br></strong>";
//tratamiento para la presión arterial
    $this->tra_presion="";
    if($presion>140 && $presiond>90)
    {
        if($hipertension==0)($this->tra_presion="- Chequear su presioacute;n arterial.</br>");
        else
        {
            $this->tra_presion="- Consultar a su m&eacute;acute;dico para evaluar tratamiento futuro de la
            presioacute;n arterial.</br>";
        }
    }

//tratamiento para la hipertrigliceridemia
    $this->trat_hiper_trigl="";
    if($hiper_trigl)
    {
        if($trigliceridos>150)($tra_trig="- Realizar tratamiento para disminuir los triglic&eacute;ridos
        reducir los triglic&eacute;ridos en caso de ser necesario debe reducir el peso corporal, los
        niveles de alcohol y tratar la diabetes mellitus. Adem&eacute;s es necesario reducir el consumo
        de azuac&eacute;ar.</br>");
        else ($tra_trig='');
        $this->trat_hiper_trigl=$tra_trig;
    }

//tratamiento para pacientes alcoholicos y fumadores
    if($alcoholismo==1)($this->tra_alc="</br>- Controlar la ingesti&eacute;n de bebidas
    alcoholicas.</br>");
    else ($this->tra_alc='');
    if($fuma==2)($this->tra_fuma="</br>- Erradicar el h&aacute;bito de fumar y consultar
    una cl&iacute;nica de deshabitua&eacute;n si lo requiere debido a que est&eacute;
    h&aacute;bito reduce la expectativa de vida y favorece el desarrollo de la aterosclerosis
    y las enfermedades tumorales.</br> ";$this->padecimiento=1; )
    else ($this->tra_fuma='');

//diets necesarias seg&uacute;n peso

    $diets_caloricas=$diets->diets_caloricas();

    if($imc>=24.9)
    {
        $this->indice_masa_corporal=0;
    }
    elseif($imc<18.5)
    {
        $this->indice_masa_corporal=1;
    }

    else
    {
        $this->indice_masa_corporal=2;
    }

    if($diets_caloricas<1200)
    {
        $this->dieta_recomendada=1200;
    }
    else if($diets_caloricas<1500)
    {
        $this->dieta_recomendada=1500;
    }
    else if($diets_caloricas<1800)
    {
        $this->dieta_recomendada=1800;
    }
    else if($diets_caloricas<2000)
    {
        $this->dieta_recomendada=2000;
    }
    else if($diets_caloricas<2500)
    {
        $this->dieta_recomendada=2500;
    }

```



```

        $valor_cldl=true;
        $this->farmacologico=1;
    }
}
elseif($clasificacion=='riesgo moderado')
{
    if($cldl>=130)
    {
        $valor_cldl=true;
        $this->farmacologico=1;
    }
}
else
{
    if($cldl>=160)
    {
        $valor_cldl=true;
        $this->farmacologico=1;
    }
}
if ($valor_cldl || $trigliceridos<200)
{
    $this->convination=0;
    $this->farmacos="Estatina";
    $this->farmacos1="Niacina";
    $this->farmacos2="Resina (Ezetimibe) y PPG";
}
else if ($valor_cldl || ($trigliceridos<500 && $trigliceridos>200) || $nohdl>200 )
{
    $this->convination=0;
    $this->farmacos="Estatina";
    $this->farmacos1="Niacina";
    $this->farmacos2="Fibrato";
}
if ($trigliceridos>500 && !$valor_cldl)
{
    $this->farmacologico=1;
}
else
{
    $obeso= "Mantener el peso corporal";
}

if($this->farmacologico==1)
{
    $tratamiento=$obeso." la dieta recomendada es ".$this->dietarecomendada.
    $this->trat_hiper_trigl.$this->tra_presion."Este paciente presenta ciertos padecimientos
    por lo que se le recomienda:<br>".$this->tratamiento_primeros."Los medicamentos que debe
    tomar son ".$a href=../../../../insercionhc/mostrarfarmacos>".$this->farmacos.", ".
    $this->farmacos1.", ".$this->farmacos2."</a>" ;
}
else
{
    $tratamiento=$obeso." la dieta recomendada es ".$this->dietarecomendada."
    </a>".$this->trat_hiper_trigl.$this->tra_presion."Este paciente presenta ciertos
    padecimientos por lo que se le recomienda:<br>".$this->tratamiento_primeros ;
}

//Actualizar en la base de datos el diagnostico y tratamiento
$con = Propel::getConnection();

$diagnostico= new Criterias();
$diagnostico->add(ConsultaPeer::ID_CONSULTA,$con);
$actualizar= new Criterias();
$actualizar->add(ConsultaPeer::DIAGNOSTICO_CLASIFICACION,$clasificacion);
BasePeer::doUpdate($diagnostico,$actualizar,$con);
$actualizar1= new Criterias();
$actualizar1->add(ConsultaPeer::DIAGNOSTICO,$final);
BasePeer::doUpdate($diagnostico,$actualizar1,$con);
$actualizar2=new Criterias();
$actualizar2->add(ConsultaPeer::TRATAMIENTO_RECOMENDACIONES,$tratamiento);
BasePeer::doUpdate($diagnostico,$actualizar2,$con);
}
}

```

Fig. 36: Código fuente del método “executeDiagnostico_comp ()”.

3.4. Integración con Ajax.

Ajax esta conformada por la unión de un grupo de tecnologías tales como:

- **XHTML y CSS:** Crear una presentación basada en estándares.
- **DOM:** Interacción y manipulación dinámica de la presentación.
- **XML, XSLT y JSON:** Intercambio y manipulación de información.
- **XMLHttpRequest:** Intercambio asíncrono de información.
- **JavaScript:** Unir todas tecnologías anteriormente mencionadas.

Hoy en día las aplicaciones mediante el uso de JavaScript presentan numerosas interacciones en el lado del cliente, efectos visuales complejos y comunicaciones asíncronas con los servidores pero la programación de estas funcionalidades se hace una tarea muy engorrosa. Afortunadamente Symfony incluye una serie de **helpers** que automatizan muchos de los usos comunes de JavaScript en las plantillas. Los programadores solo tienen que ocuparse de incluir las bibliotecas de funciones adecuadas en el proyecto que permiten hacer llamadas **Ajax** con muy poco esfuerzo por parte del programador y Symfony se encarga de lidiar con la sintaxis necesaria y con las posibles incompatibilidades entre navegadores. Los **helpers** de **Ajax** permiten al usuario actualizar partes de la página web pinchando sobre un enlace, enviando un formulario o modificando un elemento de formulario. Todos estos *helpers* disponen de múltiples opciones que proporcionan una mayor flexibilidad, sobre todo mediante el uso de las funciones de tipo callback.

Para el desarrollo de la aplicación se aprovechó la librería prototype librería que Symfony incluye y haciendo uso de esta librería se usaron helpers como:

- **link_to_remote():** Muestra un enlace que llama a una función remota.

En el ejemplo que a continuación se mostrará, al pulsar sobre el enlace "editar.png" se realizará una llamada en segundo plano a la acción insercionhc/gestionarMed. La respuesta devuelta por el servidor se mostrará automáticamente en el elemento de la página cuyo atributo id sea igual a 'med'.\$idm.


```
<?php echo link_to_remote(image_tag('editar.png','title="Editar"'),
    array(
    'update'=>'med'.$idm,
    'url'=>'insercionhc/gestionarMed',
    'with'=>"'idMed=' + document.getElementById('idmed".$idm."' ).value",
    ))
```

Fig.37: Muestra de código fuente donde se emplea el helper “**link_to_remote()**”.

- **form_remote_tag():** Envía los datos del formulario al servidor y actualiza un elemento de la página con la respuesta del servidor.

El helper `form_remote_tag()` crea una etiqueta `<form>` de apertura. En el ejemplo el envío del formulario consiste en el envío en segundo plano de una petición de tipo POST a la acción `insercionhc/actualizarMed` y con las variables definidas como parámetro de la petición. La respuesta del servidor reemplaza los contenidos del elemento cuyo atributo `id` sea igual a `'med'.$med->getIdMedicamento()`.

```
<?php echo form_remote_tag(array(
    'update'=>'med'.$med->getIdMedicamento(),
    'url'=>'insercionhc/actualizarMed'
))
?>
```

Fig. 38: Muestra de código fuente donde se emplea el helper “**form_remote_tag()**”.

- **observe_field():** Reacciona a los cambios producidos por el usuario sobre alguno de los campos de un formulario.

En el ejemplo la acción `herramientas/mnp` se ejecuta cada vez que el usuario modifica el valor del campo de formulario que se está observando (en este caso, `'mun'`) sin necesidad de enviar el formulario. La acción puede acceder a los valores seleccionados en cada momento por el usuario en el select `'provincia'`. Por la necesidad de enviar el `id` de la provincia, se especifica con la expresión JavaScript en el parámetro `with` el valor del mismo.

```
<?php echo observe_field('provincia', array(
    'update' => 'mun',
    'url' => 'herramientas/mnp',
    'loading' => "Element.show('indicador')",
    'complete' => "Element.hide('indicador')",
    'script' => true,
    'with' => "'idp='+document.getElementById('provincia').value",
)) ?>
```

Fig. 39: Muestra de código fuente donde se emplea el helper “**observe_field()**”.

3.5. Mapa de navegación.

“Un mapa de navegación es la representación gráfica de la organización de la información de una estructura web. Expresa todas las relaciones de jerarquía y secuencia y permite elaborar escenarios de comportamiento de los usuarios. También grafica, de modo que todos los profesionales participantes en un proyecto lo tengan claro, diferencias entre páginas dinámicas, administrables o estáticas.”[26].

A continuación se muestra el mapa de navegación correspondiente al sistema alasLIPO v2.0.

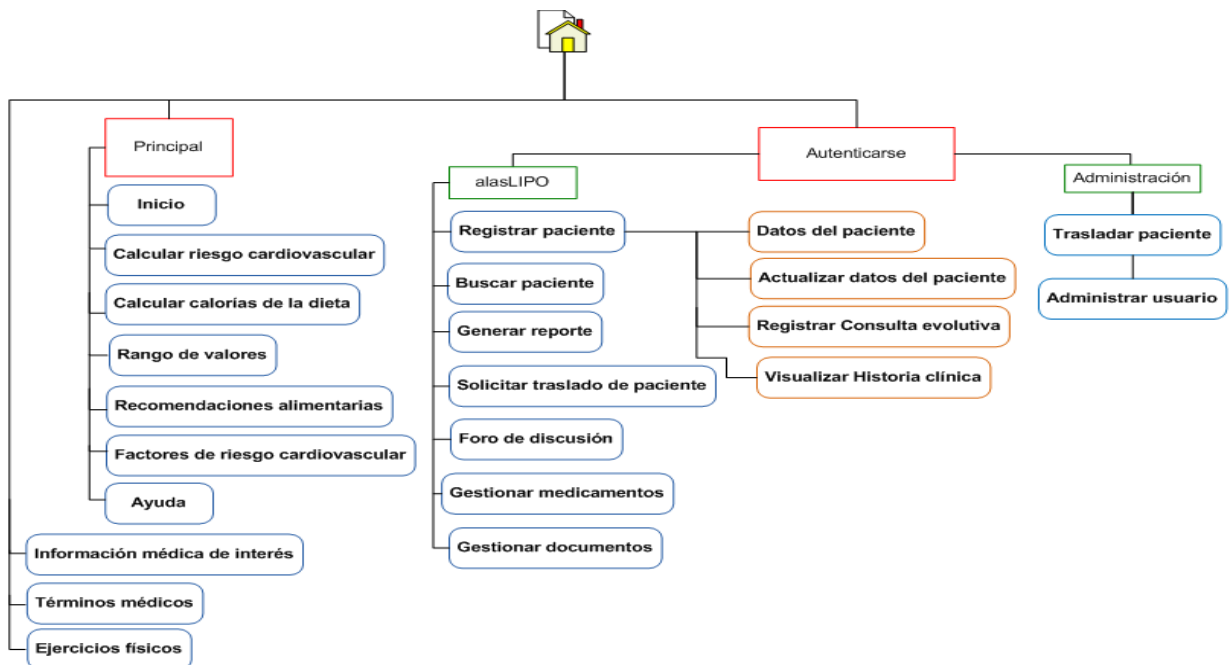


Fig. 40: Mapa de navegación.

3.6. Interfaces principales.

Para la realización de la aplicación se desarrollaron interfaces amigables y seguras, acorde con el personal que trabajará con la misma.

alasLIPO
SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPOPROTEINEMIAS

Contacto Ayuda Acerca de

Martes, 26 de mayo de 2009

Principal

- Inicio
- Calcular riesgo cardiovascular
- Calcular calorías de la dieta
- Recomendaciones alimentarias
- Factores de riesgo cardiovascular
- Ayuda

Acceso

Usuario:

Contraseña:

Entrar

[Registrarse](#)

Hipertensión Arterial

alasLIPO es un sistema para apoyar al médico en la asistencia a pacientes que sufren trastornos del metabolismo de los **lípidos** y las **lipoproteínas** del **plasma**. Esta herramienta de trabajo no constituye un sustituto de la labor asistencial del médico. Este sistema adicionalmente ofrece información teórica relacionada con las **dislipoproteinemias**, dirigida a médicos y pacientes interesados en conocer acerca de este tema.

hacer dieta sin pasar hambre ●●●

Cuatro normas a seguir.

- 1.- Distribuir su alimentación en tres grandes "tomas": desayuno, almuerzo y comida, meriendas en los horarios intermedios y además antes de dormir.
- 2.- Balancear sus comidas para lograr el aporte nutricional requerido. Escoja métodos de cocción sanos. Prefiera el consumo de verduras, leguminosas, frutas, cereales, pescados, leche desnatada.
- 3.- En una dieta sana los hidratos de carbono simples (azúcar) deben ser incluidos.
- 4.- Iniciar todas las comidas con una ensalada, para saciar con mayor facilidad al centro del apetito que se encuentra en el sistema nervioso central.

CONOCETE POR DENTRO

Información médica de interés

Términos médicos

Ejercicios físicos

© Universidad de las Ciencias Informáticas, 2009.

Fig.41: Pagina principal de la aplicación.

Se selecciona la opción “**Calcular riesgo cardiovascular**”. Se muestra la interfaz donde se realiza este cálculo. Para realizar este cálculo se deben llenar los campos que se especifican en el formulario, teniendo cuidado de no dejar de llenar los que tienen asterisco, pues son de carácter obligatorio.

Una vez presionado el botón “**Calcular**” se mostrará una nueva interfaz con los resultados obtenidos para los parámetros que fueron introducidos.

The screenshot displays the 'alás LIPO' web application interface. At the top, there is a navigation bar with 'Contacto', 'Ayuda', 'Acerca de', and 'Sistema'. The main header includes the logo 'alás LIPO' and the text 'SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPOPROTEINEMIAS'. A sidebar on the left contains a menu with options like 'Inicio', 'Calcular riesgo cardiovascular', and 'Ayuda'. The main content area is divided into two sections: 'Calcular - Riesgo cardiovascular' and 'Mostrar - Riesgo Cardiovascular'.

Calcular - Riesgo cardiovascular: This section contains a definition of cardiovascular risk and a diagram illustrating the relationship between LDL (labeled 'Colesterol malo') and HDL (labeled 'Colesterol bueno'). The form includes the following fields:

- Edad: años
- Sexo:
- ¿Es fumador?:
- Colesterol: mg/dL (Ej:0.5)
- cHDL: mg/dL (Ej:0.5)
- Presión arterial sistólica: mm/Hg
- ¿Cumple tratamiento para la hipertensión?:

 Buttons for 'Calcular' and 'Cancelar' are at the bottom right of the form.

Mostrar - Riesgo Cardiovascular: This section shows the results of the calculation. It includes the same definition of risk and a diagram with cartoon characters for HDL and LDL. The results are:

- Resultado:** Usted tiene una **cardiopatía isquémica "dura"** en los próximos 10 años menor que 1%, de acuerdo a los valores que indicó anteriormente.
- Si Usted tiene un riesgo cardiovascular a 10 años:**
 - menor al 10% se encuentra en **riesgo bajo**.
 - entre 10% y 20% se encuentra en **riesgo moderado**.
 - mayor al 20% se encuentra en un **riesgo alto**.
- Usted indicó los siguientes valores:**
 - Edad: 22 años
 - Sexo: Femenino
 - Fumador: No
 - Colesterol: 123 mg/dL
 - cHDL: 123 mg/dL
 - Presión arterial sistólica: 120 mm/Hg
 - ¿Cumple tratamiento para la hipertensión?: No

 A 'Cerrar' button is located at the bottom right of the results section.

At the bottom of the page, there is a copyright notice: '© Universidad de las Ciencias Informáticas, 2009.'

Fig.42: Interfaz “Calcular – Riesgo cardiovascular”. Resultado del cálculo.

Se selecciona la opción “**Calcular calorías de la dieta**”. Se muestra la interfaz donde se realiza este cálculo. Para realizar este cálculo se deben llenar los campos que se especifican en el formulario, teniendo cuidado de no dejar de llenar los que tienen asterisco, pues son de carácter obligatorio. En la interfaz se muestra información ofrecida por especialistas en estos temas e información sobre el estado nutricional según índice de masa corporal. Finalmente después de haber introducido todos los datos presionar el botón “Calcular” y se realizará el cálculo deseado.

Una vez presionado el botón “**Calcular**” se mostrarán los resultados obtenidos para:

- **Índice de masa corporal.**
- **Peso ideal**
- **Calorías de la dieta**

Cuando se selecciona la opción de visualizar el resultado para Calorías de la dieta en el resultado existen enlaces a páginas donde se ofrecen posibles dietas a seguir para mejorar la calidad de vida.

alas LIPO
SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPROTEINEMIAS

Contacto Ayuda Acerca de Salir

Dr(a). dayana canova
Nivel de acceso: Doctor
Martes, 26 de mayo de 2009

Principal

Inicio

Calcular riesgo cardiovascular

Calcular calorías de la dieta

Recomendaciones alimentarias

Factores de riesgo cardiovascular

Ayuda

Calcular - Calorías de la dieta

Peso: [] * Kgs

Talla: [] * m (Ej:1.82)

Calcular Cancelar

Los campos marcados con (*) son obligatorios.

Resultado del cálculo de las calorías de la dieta

Estado nutricional según índice de masa corporal.

Bajo peso: <18.5 Kgs/m²

Normal: 18.5 - 24.9 Kgs/m²

Sobrepeso: 25.0 - 29.9 Kgs/m²

Obesidad: 30.0 - 39.9 Kgs/m²

Obesidad extrema: >40 Kgs/m²

Índice de masa corporal

Su índice de masa corporal es de 13.31 Kgs/m²

Peso ideal

Su peso ideal oscila entre 44.45 y 59.82 Kgs, lo que corresponde al rango normal del índice de masa corporal.

Calorías de la dieta

De acuerdo a su peso ideal se le recomienda una dieta de 2500 calorías.

© Universidad de las Ciencias Informáticas, 2009.

Fig.43: Interfaz “Calcular – Calorías de la dieta”. Resultado del cálculo.

Se selecciona la opción “**Registrar paciente**”. Se muestra la interfaz correspondiente para insertar los datos personales del paciente. Se deben llenar los campos que se especifican en el formulario, teniendo cuidado de no dejar de llenar los que tienen asterisco, pues son de carácter obligatorio.

alás LIPO
SISTEMA DE AYUDA MÉDICA PARA LA
ATENCIÓN DE LAS DISLIPROTEINEMIAS

Contacto Ayuda Acerca de Salir

Dr(a). Alejandro Pérez Lara
Nivel de acceso: Doctor

Miércoles, 27 de mayo de 2009

Principal

Inicio

Calcular riesgo cardiovascular

Calcular calorías de la dieta

Recomendaciones alimentarias

Factores de riesgo cardiovascular

Ayuda

alásLIPO

Registrar paciente

Buscar paciente

Generar reporte

Solicitar traslado de paciente

Foro de discusión

Insertar - Datos personales del paciente

Nombre: *

País: Cuba

Primer apellido: *

Provincia: ---Seleccione--- *

Segundo apellido: *

Municipio: ---Seleccione--- *

Edad: * años

Dirección particular: *

Sexo: M

Teléfono:

Carnet de identidad: *

Identidad: *

Los campos marcados con (*) son obligatorios.
El campo "Identidad" sólo se llena cuando el paciente no posee nacionalidad cubana.

© Universidad de las Ciencias Informáticas, 2009.

Fig.44: Interfaz “Insertar-Datos personales del paciente”.

Esta interfaz presenta los formularios asociados a los exámenes de un paciente que se deben registrar. Se debe seleccionar el formulario para insertar los datos del mismo. Una vez introducidos los datos necesarios el sistema realizará los cálculos del factor de riesgo, índice de masa corporal, peso

ideal, y calorías de la dieta. Una vez realizados estos cálculos se emitirá el posible diagnóstico y tratamiento farmacológico y no farmacológico para el paciente.

The screenshot displays the 'alas LIPO' web application interface. At the top, there is a header with the logo 'alas LIPO' and the subtitle 'SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPOPROTEINEMIAS'. To the right of the header, there are navigation links: 'Contacto', 'Ayuda', 'Acerca de', and 'Salir'. Below these links, the user information is shown: 'Dr(a). Alejandro Pérez Lara' and 'Nivel de acceso: Doctor'. The date 'Miércoles, 27 de mayo de 2009' is also present.

On the left side, there is a vertical menu with the following items: 'Principal', 'Inicio', 'Calcular riesgo cardiovascular', 'Calcular calorías de la dieta', 'Recomendaciones alimentarias', 'Factores de riesgo cardiovascular', 'Ayuda', 'alasLIPO', 'Registrar paciente', 'Buscar paciente', 'Generar reporte', 'Solicitar traslado de paciente', and 'Foro de discusión'.

The main content area is titled 'Insertar - Datos del paciente.' and contains several expandable sections:

- Antecedentes patológicos personales**
- Antecedentes patológicos familiares**
- Examen físico**
- Signos asociados a las dislipoproteinemias**
- Determinación de las lipoproteinemias**
- Otros exámenes**
- Medicamentos**
- Hemoquímica**
- Ultrasonido doppler carotídeo**
- Cálculos auxiliares**
- Diagnóstico, tratamiento y observaciones**

The 'Diagnóstico' section contains the following text:

Alteraciones de las lipoproteínas:
- El cLDL moderadamente elevado.

Otros diagnósticos:
- Pese Corporal con criterios de bajo peso (<18.5 Kgs/m²).

Los factores de riesgo mayores son:
- Antecedente en familiares de primera línea de cardiopatía isquémica en edades tempranas de la vida.

El riesgo de cardiopatía isquémica "dura" a 10 años es elevado por:
- Síndromes coronarios.
- Infarto cerebral.
- Glucosa en el plasma elevada.

The 'Tratamiento y recomendaciones' section contains:

El paciente debe:
- Mantener el peso corporal, por lo que necesita una dieta de 2500 calorías.

Este paciente presenta ciertos padecimientos por lo que se le recomienda:
- Consultar a un especialista en endocrinología para la glucemia elevada.
- Se recomienda **Estatina**, **Niacina** o **Resina (Ezetimibe)** y **PPG** (Ver los enlaces correspondientes a cada medicamento para conocer como administrar los fármacos.)

The 'Observaciones' section is currently empty and contains a rich text editor toolbar with various icons for text formatting and alignment. Below the editor, there is a 'Ruta:' label and two buttons: 'Guardar' (with a green checkmark icon) and 'Cancelar' (with a red X icon).

At the bottom right of the interface, there is a 'Cerrar' button. The footer of the page reads '© Universidad de las Ciencias Informáticas, 2009.'

Fig.45: Interfaz "Insertar-Datos del pacientes". Diagnóstico y tratamiento.

Se selecciona la opción **"Buscar Paciente"**. Se muestra la interfaz correspondiente para especificar los criterios de búsqueda por los cuales se desea que la búsqueda se realice. Finalmente una vez especificados uno, varios o todos los criterios de búsqueda presionar el botón **"Buscar"**, y se mostrará un listado con todos los pacientes que coincidan con los criterios especificados o un mensaje donde se especifique que no se encontró ningún paciente que coincidiera con los criterios especificados.

Una vez encontrado el paciente deseado se podrá seleccionar tres opciones a realizar:

- **Visualizar Historia Clínica.**
- **Registrar Consulta Evolutiva.**
- **Actualizar Historia Clínica.**

The screenshot shows the 'alás LIPO' web application interface. The header includes the logo 'alás LIPO' and the text 'SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPOPROTEINEMIAS'. There are navigation links for 'Contacto', 'Ayuda', 'Acerca de', and 'Salir'. The user is identified as 'Dr(a). dayana canova' with a 'Nivel de acceso: Doctor' and the date 'Miércoles, 27 de mayo de 2009' is displayed.

The main content area is titled 'Buscar - Paciente' and contains a search form with the following fields: 'Nombre:', 'Primer apellido:', 'Segundo apellido:', and 'Carnet de identidad:'. Below the form are 'Buscar' and 'Cancelar' buttons. A message states: 'Si usted no introduce algún criterio de búsqueda se le mostrará un listado con todos sus pacientes.'

The search results are displayed in a table with the following data:

Resultados de la búsqueda: 2 pacientes encontrados.			
Nombre y apellidos	Visualizar Historia clínica	Registrar Consulta evolutiva	Actualizar datos del paciente
Dayana Canova Ramirez	06112404498		
Jaidel Perez Dopazo	06112404465		

The footer of the application reads: '© Universidad de las Ciencias Informáticas, 2009.'

Fig.46: Interfaz "Buscar - Paciente". Resultado.

Se selecciona la opción "**Visualizar Historia Clínica**" y se muestra la Historia Clínica del paciente.

Se muestran:

- Datos personales del paciente.
- Listado de consultas realizadas al paciente, con las informaciones pertinentes a estas consultas.
- Diagnóstico, tratamiento, recomendaciones y observaciones emitidas al paciente en su última consulta.
- Gráficas de comportamiento de la enfermedad en el paciente.

alas LIPO
SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPOPROTEINEMIAS

Contacto Ayuda Acerca de Salir
Dr(a). dayan canao
Nivel de acceso: Médico
Lunes, 8 de junio de 2009

Principal
Inicio
Calcular riesgo cardiovascular
Calcular calorías de la dieta
Recomendaciones alimentarias
Factores de riesgo cardiovascular
Rango de valores
Ayuda

Visualizar - Historia clínica

Historia clínica del paciente

Historia clínica	Nombre y apellidos	Sexo	Edad	Fecha de ingreso	Dirección particular
75112405353	Livan Rivera Lopez	M	33	2009-06-08 15:19:49	sgdfgf. Fomento, Sancti Spiritus.

Gráfica de estado del paciente
Colesterol | cLDL | cHDL | Triglicéridos | IMC | PAS | PAD

Índice de masa corporal

Fecha	IMC
08/06/2009	10.46
08/06/2009	10.46
08/06/2009	26.16

Cerrar

Consultas realizadas: 4

Tipo de consulta	Fecha	Ver consulta
Primera consulta	2009-06-08 15:19:49	
Consulta evolutiva	2009-06-08 15:22:53	
Consulta evolutiva	2009-06-08 15:23:44	
Consulta evolutiva	2009-06-08 15:24:35	

Los datos que se muestran a continuación corresponden a la última consulta registrada.

Diagnóstico
Los factores de riesgo cardiovascular que presenta son:
- El colesterol elevado. Ha aumentado desde la última consulta.
- El cLDL moderadamente elevado. Ha aumentado desde la última consulta.
- El valor de los triglicéridos es elevado. Ha aumentado desde la última consulta.

Tratamiento y recomendaciones
Observaciones

Imprimir Lerrar

Fig.47: Interfaz "Visualizar- Historia clínica".

Se selecciona la opción "Registrar Consulta Evolutiva" y se muestra la interfaz correspondiente a Registrar Consulta Evolutiva. Una vez especificados todos los campos se mostrarán los resultados para los cálculos de peso ideal, índice de masa corporal, riesgo cardiovascular y calorías de la dieta, se emitirá además un nuevo diagnóstico y un nuevo tratamiento farmacológico y no farmacológico.

The screenshot shows the 'alás LIPO' web application interface. At the top, there is a header with the logo 'alás LIPO' and the text 'SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPOPROTEINEMIAS'. To the right of the header, there are navigation links: 'Contacto', 'Ayuda', 'Acerca de', and 'Salir'. Below these links, the user's name 'Dr(a). dayana canova' and 'Nivel de acceso: Médico' are displayed, along with the date 'Jueves, 28 de mayo de 2009'. The main content area is titled 'Insertar - Consulta evolutiva'. It features a dropdown menu for 'Seleccione unidad de medida:' set to 'mg/dL'. Below this, there are several input fields for medical data: 'Lolesterol:' (with a red asterisk), 'Triglicéridos:' (with a red asterisk), 'cHDL:' (with a red asterisk), 'cLDL:' (with a red asterisk), 'Peso:' (with a red asterisk and 'kg' unit), 'Presión arterial sistólica:' (with a red asterisk and 'mm/Hg' unit), and 'Presión arterial diastólica:' (with a red asterisk and 'mm/Hg' unit). There is also an 'Otros:' field. At the bottom of the form, there are 'Guardar' and 'Cancelar' buttons. A note at the bottom of the form states 'Los campos marcados con (*) son obligatorios.' The footer of the page contains the copyright information '© Universidad de las Ciencias Informáticas, 2009.'

Fig.48: Interfaz "Insertar – Consulta evolutiva" (CE).

The screenshot shows the 'Visualizar' interface of the 'alas LIPO' system. The header includes the logo 'alas LIPO' and the text 'SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPOPROTEINEMIAS'. The user is identified as 'Dr(a). dayana canova' with 'Nivel de acceso: Médico' and the date 'Sábado, 30 de mayo de 2009'. The interface is divided into several sections:

- Principal:** A sidebar menu with options: Inicio, Calcular riesgo cardiovascular, Calcular calorías de la dieta, Recomendaciones alimentarias, Factores de riesgo cardiovascular, Ayuda, alasLIPO, Registrar paciente, Buscar paciente, Generar reporte, Solicitar traslado de paciente, and Foro de discusión.
- Visualizar - Cálculos auxiliares, diagnóstico, tratamiento y recomendaciones y observaciones.** The main content area, which includes:
 - Diagnóstico:** A section titled 'Los factores de riesgo cardiovascular que presenta son:' with a list item '- El cLDL moderadamente elevado. Ha aumentado desde la última consulta.'
 - Tratamiento y recomendaciones:** A section with fields for 'Riesgo cardiovascular', 'Índice de masa corporal', 'Paso ideal', and 'Calorías de las dietas'.
 - Recomendaciones:** A rich text editor with a toolbar containing icons for text formatting, alignment, and linking.
 - Ruta:** A text input field.
 - Buttons:** 'Guardar' (with a green checkmark icon) and 'Cancelar' (with a red X icon).
- Footer:** '© Universidad de las Ciencias Informáticas, 2009.'

Fig.49: Interfaz “Visualizar calculos auxiliares, diagnósticos, tratamiento, recomendaciones y observaciones” (CE).

Se selecciona la opción “**Generar reporte**”. Se muestra la interfaz correspondiente con los formularios a especificar para poder generar el reporte deseado. Para generar un reporte se debe seleccionar o especificar los campos que se encuentran en los formularios. Si no se especifica ningún campo presionar el botón “**Generar**” se generará un reporte mostrando un listado de todos los pacientes con su nombre, número de Historia Clínica, edad, sexo dirección, país, provincia y municipio de cada uno de estos.

Una vez obtenido el reporte deseado, se podrá imprimir (exportar a pdf) o exportar a Excel con solo presionar en los enlaces correspondientes a estas acciones.

alás LIPO
SISTEMA DE AYUDA MÉDICA PARA LA ATENCIÓN DE LAS DISLIPOPROTEINEMIAS

Contacto Ayuda Acerca de Salir
Dr(a). Alejandro Pérez Lara
Nivel de acceso: Doctor
Miércoles, 27 de mayo de 2009

Principal
Inicio
Calcular riesgo cardiovascular
Calcular calorías de la dieta
Recomendaciones alimentarias
Factores de riesgo cardiovascular
Ayuda

alásLIPO
Registrar paciente
Buscar paciente
Generar reporte
Solicitar traslado de paciente
Foro de discusión

❖ Generar - Reporte.

Datos personales del paciente
 Antecedentes patológicos personales
 Antecedentes patológicos familiares
 Riesgo cardiovascular, triglicéridos y colesterol
 Personalizar resultado del reporte

Con los criterios que se visualizan en "Datos personales del paciente", "Antecedentes patológicos personales", "Antecedentes patológicos familiares" y "Riesgo cardiovascular, triglicéridos y colesterol" usted puede generar un reporte de los pacientes que se correspondan con los mismos.

Con los parámetros que se visualizan en "Personalizar resultado del reporte" usted puede escoger cuál de ellos desea visualizar en el resultado del reporte.

❖ Resultado del reporte

Resultados 1 - 1 de 1

Nombre y apellidos	HC	Edad	Sexo	Dirección	País	Provincia	Municipio
Dayana Canova Ramirez	86112404497	22	M	Calle 50c entre 1a y 1b #126A	Cuba	La Habana	Artemisa

Cerrar

© Universidad de las Ciencias Informáticas, 2009.

Fig.50: Interfaz “Generar - Reporte”. Resultado del reporte.

3.7. Validación de la aplicación.

La aplicación alásLIPO v2.0 se validó a nivel de cliente donde se creó utilizando el lenguaje Javascript, una clase validadores.js que se incluye en las páginas que necesitan validar los datos antes de enviarlos al servidor, en la misma se crearon funciones para verificar que los valores de entrada sean correctos .

Hay validaciones para diferentes tipos de datos por ejemplo para números enteros, decimales, letras; rangos de números, cadenas de texto, y además validaciones para calcular valores (ejemplo Cldl). Existen otras validaciones que inhabilitan ciertos campos en dependencia de los valores de entrada de otros, impidiendo así el proceso de inserción en la base de datos con datos incorrectos.

La aplicación consta además de validación a nivel de servidor a través de Symfony que incluye un conjunto de clases que fueron utilizadas para el desarrollo de la aplicación tales como `sfStringValidator`, `sfNumberValidator`, `sfFileValidator`, `sfPropelUniqueValidator`, `sfFileValidator`, `sfGuardUserValidator` (pertenece al plugin `sfGuardUser`) entre otras. Además pueden realizarse validaciones usando métodos de la clase acción donde:

“La clave para un correcto funcionamiento de la validación es respetar la convención de nombres para los métodos de la acción:

- `validateNombreAccion` es el método de validación, que devuelve true o false. Se trata del primer método buscado cuando se solicita la acción `NombreAccion`. Si no existe, la acción se ejecuta directamente.
- `handleErrorNombreAccion` es el método llamado cuando el método de validación falla. Si no existe, entonces se muestra la plantilla `Error`.
- `executeNombreAccion` es el método de la acción. Debe existir para todas las acciones.” [7].

3.8. Seguridad de la aplicación.

La aplicación contará con un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría (Authentication, Authorization and Accounting, AAA) con Autenticación de firma única (Single Sign On) donde para poder acceder a la aplicación los usuarios necesitarán estar autenticados previamente haciendo coincidir el usuario y la contraseña introducidos con las credenciales que están registradas en el servidor de Infomed encargado del almacenamiento de esta información. El sistema brindará la posibilidad de realizar funcionalidades (calcular factor de riesgo cardiovascular, visualizar documentos, recomendaciones alimentarias, mapa de navegación, ejercicios

físicos entre otras funcionalidades) sin la necesidad de autenticarse pero cada acción antes de ser ejecutada, pasará por un filtro que verificará si el usuario actual presenta privilegios de acceso.

En Symfony, los privilegios están compuestos por dos partes:

- Las acciones seguras requieren que los usuarios estén autenticados.
- Las credenciales son privilegios de seguridad agrupados bajo un nombre y que permiten organizar la seguridad en grupos.

Los permisos fueron definidos mediante el uso de credenciales, para cada una de las acciones en cada módulo de la aplicación. Para ello se creó un archivo YAML, `security.yml` en el directorio `config` de cada módulo aprovechando la escalabilidad de los archivos de configuración que brinda Symfony.

Específicamente la seguridad en `alasLIPOv2.0` es implementada a partir del plugin **`sfGuardplugin`** el cual ofrece un modelo de seguridad basado en usuario, grupo y permisos, además de facilitar varios módulos para la administración de los mismos y otras opciones de gestión de usuarios más avanzadas que las que proporciona por defecto Symfony.

“`sfGuardplugin` brinda el modelo (usuario, grupo y objetos de permisos) y los módulos (backend y frontend) para asegurar la aplicación.

`SfGuardplugin` añade 4 módulos al proyecto:

- **`sfGuardAuth`** - para el frontend.
- **`sfGuardGroup`** - para el backend. Gestiona los grupos de usuarios.
- **`sfGuardPermission`** - para el backend. Gestiona los permisos (roles o credenciales).
- **`sfGuardUser`** - para el backend. Gestiona los usuarios de la aplicación.

Para el acceso a la aplicación se definieron 6 roles según las funcionalidades que estos podrán realizar. Los roles definidos fueron `medico_administrador_general`, `medico`, `medico_administrador_traslado`, `administrador_traslado`, `administrador_usuario` y `administrador_sistema`. Para el acceso a los datos se definieron 3 niveles, nacional, provincial y municipal. Estos niveles podrán ser asignados a los roles `medico_administrador_traslado`,

administrador_traslado administrador_usuario y administrador_sistema permitiendo que solo puedan gestionar la información según su nivel.

3.9. Conclusiones.

En el transcurso del presente capítulo se obtuvieron a partir de los diagramas de clases de diseño los diagramas de componentes correspondientes. Se describió y mostró el código fuente de una de los principales métodos con que cuenta el sistema. Se argumentó sobre la validación y seguridad de la que dispone la aplicación. Además se presentaron y describieron las interfaces más significativas de la solución informática alasLIPO v2.0.

Conclusiones.

Una vez culminado el trabajo es posible afirmar que se les dio cumplimiento a los objetivos trazados para el mismo.

Se realizó el diseño de las clases, cumpliendo estándares y con la programación orientada a objetos, el cual sirvió de base para la posterior implementación.

Se implementó las clases diseñadas cumpliendo las normas propuestas por el sistema nacional de salud para el despliegue de aplicaciones en servidores de Infomed y utilizando herramientas propuestas por el grupo de desarrollo, en su mayoría distribuidas bajo licencias de software libre en correspondencia con las políticas de la Universidad y del país.

Recomendaciones.

- Se recomienda aplicar algoritmos y técnicas modernas de la inteligencia artificial en la predicción del diagnóstico y tratamiento de los pacientes dislipoproteinémicos.
- Se recomienda para una nueva versión del sistema incorporar la posibilidad de gestionar datos históricos.

Referencias bibliográficas

- [1] Dr. Alfredo Nasiff Hadad, Luis Miguel Pérez Pérez. Primer consenso nacional de dislipoproteinemias: Guía para la prevención, detección, diagnóstico y tratamiento. Publicado (17-04-08). Consultado (23-10-08). [Disponible en: http://bvs.sld.cu/revistas/end/vol17_4_06/end01406.htm]
- [2] Lic. María Isabel Clavero Quintana, Lic. Tania Díaz Padilla. La gestión de información: una herramienta para el bibliotecario escolar. Publicado (08-09-2008) Consultado (19-01-09). [Disponible en: <http://www.cfg.rimed.cu/revista/viewArticle.php?id=154>]
- [3] Ing. Marlene M.M, Ing Gonzalo Q.A. Sistema de información. Publicado (2006). Consultado (21-03-09). [Disponible en: www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r14811.DOC]
- [4] Lic. Keilyn Rodríguez Perojo, Lic. Rodrigo Ronda León. El web como sistema de información. Publicado (14-01-06). Consultado (15-03-09). [Disponible en: http://bvs.sld.cu/revistas/aci/vol14_1_06/aci08106.htm]
- [5] IT INSTITUTE Advanced Information Technology Center. ¿Qué es RUP? Publicado (2006). Consultado (15-03-09). [Disponible en: http://it-institute.org//index.php?option=com_content&task=view&id=145&Itemid=75]
- [6] Ing. Alfredo Sánchez Rodríguez, Ing. Frank Popa Sourd. Normas para el desarrollo de aplicaciones para la salud en Cuba. Aprobado (28-11-2007). Consultado (25-1-09).
- [7] Ing. Luis Antonio Salazar. Prolegómenos Sobre el Lenguaje de Modelado Unificado (UML). Publicado (2003). Consultado (06-02-09). [Disponible en: <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/prolego.pdf>]

- [8] Fabien Potencier, Francois Zaninotto.Symfony la guía definitiva. Creado (30-12-08). Consultado (23-04-09). [Disponible en:
http://www.librosweb.es/symfony_1_2/pdf/symfony_1_2_guia_definitiva_2caras.pdf
- [9] Javier Eguíluz. Symfony en pocas palabras. Consultado (12-12-08). [Disponible en:
http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html
- [10] ¿Qué es PHP? Publicado (08- 05- 09). Consultado (09-12-08). [Disponible en:
<http://www.php.net/manual/es/intro-whatis.php>
- [11] ¿Qué se puede hacer con PHP? Publicado (08- 05- 09). Consultado (09-12-08). [Disponible en:
<http://www.php.net/manual/es/intro-whatcando.php>
- [12] Cristina Lazalde Miranda, Maya Yanin Miranda. Herramientas Case. Consultado (29-6-09). [Disponible en:
<http://www.mitecnologico.com/Main/HerramientasCase>
- [13] Visual Paradigm for UML. Consultado (29-6-09). [Disponible en:
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(Iglesia_Anglicana\)_%5BMac_OS_X_cuenta_14717_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/)
- [14] ¿Qué es JavaScript? Consultado (05-5-09). [Disponible en:
<http://www.lcc.uma.es/~eat/services/html-js/manual14.html>
- [15] Blanca Cubas Cruz ¿Qué es Subversion? Publicado (Julio-08). Consultado (8-4-2009). [Disponible en:
<http://www.cesnavarra.net/cesdigital/Lists/Noticias%20CESDigital/DispFormCES.aspx?List=5ec0dfc7%2D7911%2D470b%2D8b6b%2D71ba72783fdd&ID=48>
- [16] ¿Qué es dotProject? Publicado (07-06-2006). Consultado (12-3-2009). [Disponible en:
<http://www.desarrolloweb.com/scripts/dotproyect-gestion-proyectos-php.html>

- [17] Plataforma colaborativa (trac). Publicado (19-11-08). Consultado (11-05-09). [Disponible en:
http://wiki.canaima.softwarelibre.gob.ve/wiki/index.php/Plataforma_Colaborativa_%28trac%29
- [18] Barbara Liskov. Design Patterns. Publicado (2005). Consultado (15-11-08). [Disponible en:
<http://ocw.mit.edu/NR/ronlyres/Electrical-Engineering-and-Computer-Science/6-170Fall-2005/96A8B7AF-3C20-4D92-8F07-EFA537FD82C2/0/lec18.pdf>
- [19] Craig Larman. UML y Patrones. Consultado (11-12-08). [Disponible en:
<http://bibliodoc.uci.cu/pdf/reg00061.pdf>
- [20] Patrones de Diseño en aplicaciones Web con Java J2EE. Publicado (2005). Consultado (05-03-09). [Disponible en: http://java.ciberaula.com/articulo/disenio_patrones_j2ee/
- [21] Grupo de investigación Kybele & Universidad Rey Juan Carlos. Departamento de Lenguajes y Sistemas Informáticos II. Requisitos de software. Ingeniería de software. Consultado (05-03-09). [Disponible en:
<http://kybele.escet.urjc.es/documentos/ISG/Estructurado/%5BISG-2006-07%5DRequisitosSoftware.pdf>
- [22] Universidad de San Carlos de Guatemala. Facultad de Ingeniería. Escuela de Ciencias y Sistemas. Diagrama de Casos de Uso. Publicado (2005). Consultado (28-04-09). [Disponible en: <http://www.cif.acuareladelsur.org/tutoriales/casosdeuso.pdf>.
- [23] Ivar Jacobson, Grady Booch, James Rumbaugh. El proceso unificado de desarrollo de software (volumen 1). Consultado (30-04-09). [Disponible en :
<http://bibliodoc.uci.cu/pdf/reg00060.pdf>
- [24] Libros en pantalla de SQL Server 2008. Publicado (enero de 2009).Integridad de los datos.). Consultado (30-04-09). [Disponible en: <http://msdn.microsoft.com/es-es/library/ms184276.aspx>

- [25] Conferencia 5. Teoría de diseño de bases de datos relacionales (BDR). [En línea] 2006-2007. Universidad de las Ciencias Informáticas. La Habana, Cuba.
- [26] Arquitectura de Información. Mapas de navegación. Publicado (04-05-08). Consultado (29-05-09).[Disponible en: <http://arquitecturadeinformacion.cl/como/mapa.html>]

Bibliografía

- Aja Quiroga Lourdes. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. Publicado (10-05-02). [Disponible en: <http://eprints.rclis.org/1841/1/gestion.pdf>.
- Aldana Álvarez Dagoberto. Revistas médicas cubanas vol. 17. Publicado (Diciembre-08) [Disponible en: <http://bvs.sld.cu/revistas/end/indice.html>.
- Arquitectura, normas y tecnologías para el desarrollo de aplicaciones informáticas para la Salud Pública en Cuba. Junio/2007.
- Centro de Información Cardiovascular. Publicado (enero-09). (Calculadora de IMC). [Disponible en: http://www.texasheartinstitute.org/HIC/Topics_Esp/HSmart/bmi_calculator_span.cfm
- Índice de Masa Corporal. [Disponible: <http://www.indicemasacorporal.org/>
- Inox. DotProject. Publicado (agosto-08). [Disponible en: <http://tutoriales.inox.com.mx/vhost/tutoriales/completo/index.html>
- IVANEX. Sistema de selección de las rutas optimas. Publicado (20-04-08) [Disponible en: <http://ivanex.wikidot.com/metodologia>
- JavaScript Introduction. [Disponible en: http://www.w3schools.com/JS/js_intro.asp
- M. Molinero Luis (Alce Ingeniería). Publicado (agosto-03). Modelos de Riesgo Cardiovascular. Estudio de Framingham. Proyecto SCORE. [Disponible en: <http://www.seh-lelha.org/modelries.htm>

- Nasiff Hadad Alfredo, Luis Miguel Pérez Pérez. Primer consenso nacional de dislipoproteinemias: Guía para la prevención, detección, diagnóstico y tratamiento. Publicado (17-04.08). [Disponible en:
http://bvs.sld.cu/revistas/end/vol17_4_06/end01406.htm.
- Potencier Fabien, Zaninotto Francois. Symfony la guía definitiva. Publicado (2003). [Disponible en:
http://www.librosweb.es/symfony_1_2/pdf/symfony_1_2_guia_definitiva_2caras.pdf
- Página Oficial de Subversion. Consultado (12-01-09). [Disponible en:
<http://subversion.tigris.org/>
- Página oficial de Zend. Eclipse PHP Development Tools. [Disponible en:
<http://www.zend.com/en/community/pdt>.
- Pagina oficial de PHP. ¿Qué es PHP? [Disponible en:
<http://www.php.net/manual/es/intro-what-is.php>.
- Pagina oficial de PHP. ¿Qué se puede hacer con PHP? [Disponible en:
<http://www.php.net/manual/es/intro-whatcando.php>
- Página oficial de Apache. Apache Http Server. [Disponible en:
http://projects.apache.org/projects/http_server.html
- Página oficial de MySQL. [Disponible en:
<http://dev.mysql.com/doc/refman/5.0/es/features.html>
- Página oficial de MySQL. Información general. [Disponible en:
<http://dev.mysql.com/doc/refman/5.0/es/introduction.html>
- Página oficial de Eclipse. [Disponible en:
<http://www.eclipse.org/proposals/php-ide/>.
- Página oficial de Apache. [Disponible en:
<http://httpd.apache.org/modules/>

- Rivas Miriam. Informática Médica (No. 10). [Disponible en: <http://www.informaticamedica.org.ar/numero10/monografia2.htm>].
- Rodríguez Perojo Keilyn, Ronda León Rodrigo. El web como sistema de información. Publicado (14-01-06). [Disponible en: http://bvs.sld.cu/revistas/aci/vol14_1_06/aci08106.htm].
- Sacristán Luis. PHP en Eclipse. Publicado (01-02-07). [Disponible en: <http://sentidoweb.com/2007/02/01/pdt-php-en-eclipse.php>].
- Schmuller. Aprendiendo Joseph. UML en 24 horas. Publicado (03-02-01). [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00004.pdf>].

ANEXOS

Anexo 1. Descripción expandida de los casos de uso.

Descripción del caso de uso: “Gestionar medicamentos”.

Nombre del Caso de Uso	Gestionar Medicamentos
Actores	Médico_administrador_general
Propósito	Insertar, actualizar o eliminar medicamentos
Resumen	El CU inicia cuando el Administrador general desea insertar, actualizar o eliminar un medicamento. Luego de que el Administrador general indique los datos del medicamento a insertar, actualizar o eliminar entonces el sistema registra, actualiza o elimina el medicamento correspondientemente.
Referencias	R12, R13, R14, R15, R16
Precondiciones	Que el Administrador general esté autenticado en el sistema.
Poscondiciones	Se actualizó el listado de medicamentos.
Prioridad	Secundario
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Administrador general indica Gestionar Medicamentos.	2. El sistema busca los medicamentos existentes y los muestra dando la posibilidad de insertar un nuevo medicamento, eliminar o actualizar los datos de un medicamento existente.
3. El Administrador general realiza una de las siguientes operaciones: <ul style="list-style-type: none"> • Insertar un nuevo medicamento. 	4. El sistema en dependencia de la operación indicada por el Administrador general realiza lo

<ul style="list-style-type: none"> • Eliminar un medicamento existente. • Actualizar un medicamento existente. 	<p>siguiente:</p> <ul style="list-style-type: none"> • Si el Administrador general va a insertar un nuevo medicamento entonces se ejecuta la sección "Insertar medicamento". • Si el Administrador general va a eliminar un medicamento existente, entonces se ejecuta la sección "eliminar medicamento". • Si el Administrador general va a actualizar un medicamento, entonces se ejecuta la sección "Actualizar medicamento".
Sección "Insertar Medicamento".	
Acciones del Actor	Respuesta del Sistema
<p>1. El Administrador general indica el nombre del medicamento:</p> <ul style="list-style-type: none"> • Nombre del medicamento 	<p>2. El sistema verifica que el campo este lleno.</p>
	<p>3. El sistema verifica que el dato introducido sea correcto.</p>
	<p>4. El sistema verifica que no exista un medicamento con el mismo nombre.</p>
	<p>5. El sistema registra el medicamento indicado por el Administrador general.</p>
	<p>6. El sistema muestra el listado de medicamentos actualizado.</p>
Flujo Alterno Sección "Insertar Medicamento".	
Acciones del Actor	Respuesta del Sistema
	<p>3.1. Si el Administrador general dejó el campo vacío, entonces el sistema emite un mensaje de alerta indicando que el campo está vacío.</p>
	<p>4.1. Si el Administrador general introdujo el valor incorrecto, entonces el sistema emite un</p>

	mensaje de alerta indicando error.
	5.1 Si el nombre indicado coincide con el nombre de un medicamento existente, entonces el sistema muestra un mensaje de alerta indicando que el medicamento ya existe.
Prototipo Sección “Insertar Medicamento”	
Sección “Eliminar Medicamentos”.	
Acciones del Actor	Respuesta del Sistema
1. El Administrador general indica del listado mostrado el medicamento que desea eliminar.	2. El sistema muestra un mensaje de alerta indicando si desea eliminar el medicamento.
3. El Administrador general indica que desea eliminar el medicamento.	4. El sistema elimina el medicamento indicado por el Administrador general.
	5. El sistema muestra el listado de los medicamentos actualizado.
Flujo Alternativo Sección “Eliminar Medicamento”.	
Acciones del Actor	Respuesta del Sistema
3.1 Si el Administrador general indica que no desea eliminar el medicamento entonces sale de la sección.	
Prototipo Sección “Eliminar Medicamento”	
Sección “Actualizar Medicamentos”.	
Acciones del Actor	Respuesta del Sistema

1. El Administrador general indica el medicamento que desea actualizar.	2. El sistema da la posibilidad de actualizar el campo correspondiente al medicamento indicado. <ul style="list-style-type: none"> • Nombre del medicamento.
3. El Administrador general indica el nuevo nombre del medicamento.	4. El sistema verifica que el campo este lleno.
	5. El sistema verifica que el dato introducido sea correcto.
	6. El sistema verifica que no exista un medicamento con el mismo nombre.
	7. El sistema actualiza el nombre del medicamento indicado por el Administrador general.
	8. El sistema muestra el listado de los medicamentos actualizado.
Flujo Alternativo Sección “Actualizar Medicamento”.	
Acciones del Actor	Respuesta del Sistema
3.1. Si el Administrador general no desea actualizar el nombre del medicamento entonces sale de la sección.	
	5.1. Si el Administrador general dejó el campo vacío, entonces el sistema emite un mensaje de alerta indicando que el campo está vacío.
	6.1. Si el Administrador general indicó un valor incorrecto, entonces el sistema emite un mensaje de alerta indicando error.
	7.1 Si el nombre indicado coincide con el nombre de un medicamento existente, entonces se muestra un mensaje de alerta

	indicando que ese medicamento ya existe.
Prototipo Sección “Actualizar Medicamento”	

Descripción de caso de uso: “Realizar cálculos”.

Nombre del Caso de Uso	Realizar Cálculos
Actores	Usuario
Propósito	Realizar cálculos de peso ideal, calorías de la dieta, el índice de masa corporal y el factor de riesgo cardiovascular del paciente.
Resumen	El CU inicia cuando el Usuario desea realizar los cálculos del peso ideal, las calorías de la dieta, el índice de masa corporal y el factor de riesgo cardiovascular. Luego de seleccionado el cálculo que desea realizar el usuario e indicando los datos necesarios, entonces el sistema obtiene los resultados asociados a los datos introducidos.
Referencias	R2, R3, R4, R5
Precondiciones	Que el Usuario indique realizar alguno de los cálculos.
Poscondiciones	Se realizó el cálculo del factor de riesgo cardiovascular o el cálculo de peso ideal, índice de masa corporal y calorías de la dieta.
Prioridad	Opcional
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Usuario indica realizar una de las	2. El sistema realiza una de las siguientes operaciones en dependencia de lo indicado por

siguientes operaciones:	el Usuario:
<ul style="list-style-type: none"> • Calcular Calorías de la dieta. • Calcular Riesgo cardiovascular. 	<ul style="list-style-type: none"> • Si seleccionó “Calcular Calorías de la dieta” entonces se ejecuta la sección “Calcular Calorías de la dieta”. • Si seleccionó “Calcular Riesgo cardiovascular” entonces se ejecuta la sección “Calcular Riesgo cardiovascular”.
Sección “Calcular calorías de la dieta”.	
Acciones del Actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente al cálculo de las calorías de la dieta, el peso ideal y el índice de masa corporal.
2. El Usuario inserta los datos e indica realizar el cálculo. <ul style="list-style-type: none"> • Peso • Talla 	3. El sistema verifica que todos los campos estén llenos.
	4. El sistema verifica que los datos introducidos sean correctos.
	5. El sistema calcula el índice de masa corporal mediante la siguiente fórmula: <ul style="list-style-type: none"> • $IMC = \text{Peso kg} / \text{Talla m}^2$.
	6. El sistema calcula los valores del rango del peso ideal mediante la siguiente fórmula: <ul style="list-style-type: none"> • $PI_{inferior} = \text{Talla m}^2 * 18.5$ • $PI_{superior} = \text{Talla m}^2 * 24.9$
	7. El sistema calcula las calorías de la dieta

	<p>asociadas al Usuario.</p> <ul style="list-style-type: none"> • Peso*25 si el IMC da sobrepeso. • Peso*35 si el IMC da normo peso. • Peso*45 si el IMC da bajo peso.
	8. El sistema muestra el resultado del índice de masa corporal, del peso ideal y de las calorías de la dieta correspondientes al Usuario y da la posibilidad visualizar la dieta correspondiente a las calorías calculadas.
9. El Usuario indica visualizar la dieta correspondiente a sus calorías.	10. El sistema muestra en una interfaz la dieta correspondiente a las calorías del Usuario y da la posibilidad de imprimirla.
11. El Usuario indica imprimir la dieta.	12. El sistema imprime la dieta.
Flujo Alterno Sección “Calcular calorías de la dieta”.	
Acciones del Actor	Respuesta del Sistema
	4.1 Si el Usuario dejó campos vacíos entonces el sistema emite un mensaje de alerta indicando los campos vacíos.
	5.1. Si el Usuario indicó valores incorrectos entonces el sistema emite un mensaje de alerta indicando los campos incorrectos.
9.1 Si el Usuario no desea visualizar la dieta correspondiente a sus calorías sale de la sección.	
11.1 Si el Usuario no desea imprimir la dieta sale de la sección.	
Prototipo Sección “Calcular calorías de la dieta”.	
Sección “Calcular Riesgo cardiovascular”	

Acciones del Actor	Respuesta del Sistema
	1. El sistema muestra la interfaz correspondiente al cálculo del riesgo cardiovascular.
<p>2. El Usuario inserta los datos e indica realizar el cálculo.</p> <ul style="list-style-type: none"> • Edad • Sexo (F o M) • Es Fumador (Si o No) • Colesterol • cHDL • Presión Arterial Sistólica • ¿Cumple tratamiento para la Hipertensión? (Si o No) 	3. El sistema verifica que todos los campos estén llenos.
	4. El sistema verifica que los datos introducidos sean correctos.
	5. El sistema calcula el factor de riesgo cardiovascular mediante las tablas representadas en el Anexo1.
	6. El sistema muestra el resultado del factor de riesgo cardiovascular correspondiente al Usuario.
Flujo Alterno Sección “Calcular Riesgo cardiovascular.”	
Acciones del Actor	Respuesta del Sistema
	4.1 Si el Usuario dejó campos vacíos entonces el sistema emite un mensaje de alerta indicando los campos vacíos.
	5.1. Si el Usuario indicó valores incorrectos entonces el sistema emite un mensaje de alerta indicando los campos incorrectos.

Prototipo Sección “Calcular Riesgo cardiovascular.”**Descripción del caso de uso: “Generar reporte”.**

Nombre del Caso de Uso	Generar Reporte
Actores	Médico
Propósito	Generar Reporte
Resumen	El CU se inicia cuando el Médico indica generar un reporte. Luego de indicar los criterios para realizarlo y seleccionar los criterios que se mostrarán en el resultado del mismo el sistema genera el reporte. Luego el sistema permite que se exporte a formato Excel o imprima el resultado del reporte generado.
Referencias	R11, R11.1, R11.2
Precondiciones	Que el Médico esté autenticado en el sistema.
Poscondiciones	Se generó un reporte, y se exportó el resultado del mismo.
Prioridad	Crítico
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Médico indica realizar el reporte	2. El sistema muestra la interfaz correspondiente para generar reporte.
3. El Médico indica el criterio por el cual desea realizar el reporte. <ul style="list-style-type: none"> • Datos Personales. <ul style="list-style-type: none"> - Provincia 	

<ul style="list-style-type: none">- Municipio- Edad- Sexo• Antecedentes Patológicos Personales.• Antecedentes Patológicos Familiares.• Otros<ul style="list-style-type: none">- Riesgo bajo- Riesgo moderado- Riesgo alto- Triglicéridos- Colesterol <p>El médico selecciona los parámetros a mostrar en el resultado del reporte</p> <ul style="list-style-type: none">• Datos personales<ul style="list-style-type: none">- Edad- Sexo- Dirección particular- País- Provincia- Municipio- Teléfono• Examen físico<ul style="list-style-type: none">- Peso- Presión arterial sistólica- Presión arterial diastólica- Frecuencia cardiaca por	
--	--

<p>minutos</p> <ul style="list-style-type: none"> - Circunferencia de la cintura - Circunferencia de la cadera • Determinación de la lipoproteinemias <ul style="list-style-type: none"> - Colesterol - Triglicéridos - cHDL - cLDL - Apo A-I - ApoB - Lp(a) • Otros exámenes <ul style="list-style-type: none"> - Glucemia en ayunas 	
	4. El sistema verifica si se indicó algún criterio para generar el reporte.
	5. El sistema genera el reporte a partir de los criterios indicados por el médico.
	6. El sistema muestra los datos indicados por los que se generó el reporte y el resultado del reporte generado en una tabla con los datos nombre, apellidos, número de HC y los parámetros indicados por el Médico para mostrar en el resultado del mismo. El sistema da la posibilidad de exportar a formato Excel o imprimir el resultado del reporte generado.
<p>7. El Médico selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Exportar el resultado del reporte a formato Excel. • Imprimir el reporte. 	<p>8. El sistema en dependencia de la opción seleccionada por el médico realiza lo siguiente:</p> <ul style="list-style-type: none"> • Si el Médico indica exportar un reporte a formato Excel entonces el sistema

	<p>exporta el reporte generado a formato Excel.</p> <ul style="list-style-type: none"> • Si el Médico indica imprimir el reporte entonces el sistema lo imprime.
Flujo Alterno	
Acciones del Actor	Respuesta del Sistema
	5.1 Si no se indicó ningún criterio para generar el reporte entonces el sistema genera el reporte con todos los pacientes existentes pertenecientes al médico en cuestión generado en una tabla con los datos nombre, apellidos, número de HC.
	5.2 Se ejecuta la acción 7 del flujo normal de los eventos.
Prototipo	

Descripción del caso de uso: "Registrar Datos Del Paciente".

Nombre del Caso de Uso	Registrar Datos del Paciente
Actores	Médico
Propósito	Registrar los datos de un nuevo paciente.
Resumen	El CU se inicia cuando el Médico indica registrar los datos de un nuevo paciente. Luego el sistema muestra la interfaz referente a la inserción de dichos datos, el Médico indica los mismos y entonces el sistema verifica que los datos estén correctos y sean los necesarios, además el sistema verifica que no exista otro paciente

	registrado con el mismo carnet de identidad o identidad. Luego de cumplirse todas estas condiciones entonces se guardan los datos asociados a dicho paciente y se genera y registra de ser posible un diagnóstico y tratamiento asociado al paciente.
Referencias	R6, R6.1, R6.2
Precondiciones	Que el Médico esté autenticado en el sistema.
Poscondiciones	Se registraron o no los datos asociados a un paciente.
Prioridad	Crítico
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Médico indica registrar los datos de un paciente.	2. El sistema muestra la interfaz correspondiente para registrar los datos personales del paciente.
3. El Médico registra los siguientes datos personales del paciente: <ul style="list-style-type: none"> • Nombre • Primer Apellido • Segundo Apellido • Edad • Sexo • Carnet de identidad • País • Provincia • Municipio • Dirección Particular • Teléfono • Identidad 	4. El sistema verifica que los campos siguientes estén llenos: <ul style="list-style-type: none"> • Nombre • Primer Apellido • Segundo Apellido • País • Dirección Particular • Edad • Sexo <p>En caso de que en el campo "País" indique Cuba los siguientes campos además son obligatorios:</p> <ul style="list-style-type: none"> • Provincia • Municipio

	<ul style="list-style-type: none"> • Carnet de Identidad <p>En caso de que en el campo "País" no indique Cuba el siguiente campo además es obligatorio:</p> <ul style="list-style-type: none"> • Identidad
	5. El sistema verifica que los datos introducidos sean correctos.
	6. El sistema verifica que el paciente no esté registrado en el sistema mediante los siguientes criterios de comparación: <ul style="list-style-type: none"> • Si es cubano por el Carnet de Identidad. • Si no es cubano por la Identidad.
	7. El sistema guarda los datos indicados por el Médico y genera el número de la Historia Clínica que será el número del carnet de identidad del paciente si es cubano o la identidad si no es cubano.
	8. El sistema da la posibilidad de registrar los datos asociados a los exámenes del paciente en cuestión.
9. El Médico indica los datos de los siguientes exámenes: <ul style="list-style-type: none"> • Antecedentes patológicos personales • Antecedente patológicos familiares • Examen físico • Signos asociados a las dislipoproteinemias • Determinación de las lipoproteinemias • Otros exámenes 	10. El sistema verifica que los datos necesarios tengan valor. <ul style="list-style-type: none"> • En caso de registrar el Examen Físico <ul style="list-style-type: none"> - Talla - Peso - Presión arterial sistólica - Presión arterial diastólica • En caso de registrar el examen Determinación de las Lipoproteinemias <ul style="list-style-type: none"> - Colesterol - Triglicéridos

<ul style="list-style-type: none"> • Medicamentos • Hemoquímica • Ultrasonido doppler carotideo <p>El Médico indica que exámenes desea registrar.</p>	<ul style="list-style-type: none"> - cHDL - cLDL • En caso de registrar otros exámenes <ul style="list-style-type: none"> - Glucemia en ayunas
	11. El sistema verifica que los datos de los exámenes registrados estén correctos.
	12. El sistema registra los datos de los exámenes indicados por el Médico.
	<p>13. El sistema verifica si el Médico registró los exámenes necesarios para generar los cálculos auxiliares (peso ideal, índice de masa corporal, calorías de la dieta y riesgo cardiovascular).</p> <ul style="list-style-type: none"> • Examen físico • Determinación de las lipoproteinemias
	14. El sistema genera los cálculos auxiliares. Se ejecutan las acciones 5, 6 y 7 de la sección “Calcular calorías de la dieta, peso ideal e índice de masa corporal” y la acción 5 de la sección “Calcular valor del riesgo cardiovascular” del Caso de uso “Realizar Cálculos”
	15. El sistema guarda los cálculos auxiliares, el diagnóstico y el tratamiento.
	<p>16. El sistema verifica que los examen necesarios para generar el diagnóstico estén realizados:</p> <ul style="list-style-type: none"> • Antecedentes patológicos personales • Antecedentes patológicos familiares
	17. El sistema genera y guarda el diagnóstico y tratamiento asociado al paciente en cuestión.

	18. El sistema muestra los resultados de los cálculos auxiliares, el diagnóstico y tratamiento del paciente en cuestión y da la posibilidad de que el Médico indique sus observaciones o visualice la dieta correspondiente a las calorías calculadas.
19. El Médico realiza una de las siguiente operaciones: <ul style="list-style-type: none"> • Indicar sus observaciones • Visualizar la dieta correspondiente a las calorías. 	20. El sistema en dependencia de la operación seleccionada por el Médico realiza lo siguiente: <ul style="list-style-type: none"> • Si seleccionó indicar sus observaciones entonces el sistema registra las observaciones indicadas. • Si seleccionó visualizar la dieta correspondiente a las calorías entonces se ejecuta las acciones 10-12 de la sección “Calcular el peso ideal, índice de masa corporal y las calorías de la dieta” del caso de uso “Realizar Cálculos”.
Flujo Alterno	
Acciones del Actor	Respuesta del Sistema
	5.1. Si el Médico no registra todos los datos obligatorios del paciente entonces el sistema muestra un mensaje de alerta señalando los campos faltantes.
	6.1 Si el Médico indicó valores incorrectos entonces el sistema emite un mensaje de alerta indicando los campos incorrectos.
	7.1. Si el paciente ya está registrado el sistema emite un mensaje de alerta indicando que el paciente ya existe.
9.1 El Médico no desea registrar ninguno de los exámenes y sale de la sección.	11.1 Si los datos necesarios no tienen valor en caso de registrar el examen físico, el examen

	determinación de las lipoproteinemias u otros exámenes entonces el sistema emite un mensaje de alerta.
	12.1 Si algunos de los valores de los exámenes indicados para registrar no están correctos entonces el sistema muestra un mensaje de alerta.
	14.1 Si el Médico no registró los exámenes necesarios para generar los cálculos auxiliares entonces sale de la sección.
	17.1 Si el Médico no registró los exámenes necesarios para generar el diagnóstico y tratamiento entonces el sistema muestra los resultados de los cálculos auxiliares y da la posibilidad de que el visualice la dieta correspondiente a las calorías calculadas.
17.2 El Médico indica visualizar la dieta correspondiente a las calorías.	17.3 El sistema ejecuta las acciones 10-12 de la sección "Calcular el peso ideal, índice de masa corporal y las calorías de la dieta" del caso de uso "Realizar Cálculos".
19.1 Si el Médico no desea realizar ninguna de las operaciones sale de la sección.	
Prototipo	

Descripción del caso de uso: "Buscar Paciente."

Nombre del Caso de Uso	Buscar Paciente
Actores	Médico
Propósito	Se buscan los pacientes que coincidan con los criterios de búsqueda indicado por el médico y que pertenezcan al

	médico en cuestión.
Resumen	El CU se inicia cuando el Médico desea buscar un paciente. Para realizar la búsqueda del paciente el Médico indica los criterios de búsqueda (nombre, primer apellido, segundo apellido y carnet de identidad). Luego el sistema muestra un listado con los pacientes que coincidan con los criterios indicados por el Médico. En caso de que el Médico no indique criterios de búsqueda entonces el sistema muestra todos los pacientes existentes. En ambos casos sólo se muestran los pacientes pertenecientes al Médico en cuestión.
Referencias	R8
Precondiciones	Que el Médico esté autenticado en el sistema.
Poscondiciones	Se buscó un listado de pacientes.
Prioridad	Crítico
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Médico indica buscar paciente.	2. El sistema muestra la interfaz correspondiente a la búsqueda de los pacientes.
3. El Médico indica los criterios de búsqueda para buscar los pacientes. <ul style="list-style-type: none"> • Nombre • Primer Apellido • Segundo Apellido • Carnet de identidad Indica realizar la búsqueda.	4. El sistema verifica que el Médico haya indicado algún criterio de búsqueda.
	5. El sistema verifica que los valores indicados

	estén correctos.
	6. El sistema busca los pacientes que coincidan con los criterios indicados y que pertenezcan al Médico en cuestión.
	7. El sistema muestra en una tabla el listado de los pacientes encontrados y da la posibilidad de realizarle una consulta evolutiva, visualizar su HC y actualizar los datos que le faltan por registrar al paciente.
8. El Médico selecciona una de las siguientes operaciones: <ul style="list-style-type: none"> • Realizar Consulta Evolutiva. • Visualizar Historia Clínica. • Actualizar datos del paciente. 	9. En dependencia de la opción seleccionada por el Médico el sistema realiza una de las siguientes operaciones: <ul style="list-style-type: none"> • Si seleccionó "Realizar Consulta Evolutiva" se ejecuta el CU Realizar Consulta Evolutiva. • Si seleccionó "Visualizar Historia Clínica" se ejecuta el CU Realizar Visualizar HC. • Si seleccionó "Actualizar datos del paciente" se ejecuta el CU Actualizar datos del paciente.
Flujo Alterno	
Acciones del Actor	Respuesta del Sistema
	5.1 Si el Médico no indicó ningún criterio de búsqueda el sistema busca todos los pacientes correspondientes al Médico en cuestión.
	5.2 Ir a la acción 7 del Curso Normal de Eventos.
	6.1 Si los valores indicados no están correctos entonces el sistema muestra una

	alerta indicando el valor incorrecto.
	7.1 Si el sistema no encuentra ningún paciente entonces muestra un mensaje indicando que no existen pacientes.
8.1 Si el Médico no desea realizar ninguna de las operaciones entonces sale de la sección.	
Prototipo	

Descripción del caso de uso: "Realizar Consulta Evolutiva".

Nombre del Caso de Uso	Realizar Consulta Evolutiva
Actores	Médico
Propósito	Registrar los datos de una consulta evolutiva realizada al paciente.
Resumen	El CU inicia cuando el sistema muestra la interfaz correspondiente a la Consulta Evolutiva y luego indica los datos asociados a dicha consulta y a partir de los datos indicados se genera un nuevo diagnóstico y tratamiento del paciente. También el Médico puede visualizar los cálculos auxiliares asociados a los datos registrados y/o emitir sus observaciones. El sistema registra todos los datos generados e indicados.
Referencias	R9, R9.1, R9.2
Precondiciones	<ul style="list-style-type: none"> • Que el Médico esté autenticado en el sistema. • Que se haya realizado el examen físico y la determinación de las lipoproteinemias en la primera consulta.
Poscondiciones	Se registraron los datos de una Consulta Evolutiva.

Prioridad	Crítico
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
	1. El sistema muestra la interfaz correspondiente a la realización de la Consulta Evolutiva permitiéndole al Médico insertar los datos de la misma.
2. El Médico inserta los datos del paciente e indica registrarlos. <ul style="list-style-type: none"> • Peso • Colesterol • Triglicéridos • cHDL • cLDL • Presión arterial sistólica • Presión arterial diastólica • Otros 	3. El sistema verifica que los siguientes campos estén llenos: <ul style="list-style-type: none"> • Peso • Colesterol • Triglicéridos • cHDL • cLDL • Presión arterial sistólica • Presión arterial diastólica
	4. El sistema verifica que los datos introducidos sean correctos.
	5. El sistema registra los datos introducidos por el Médico.
	6. El sistema genera los cálculos auxiliares. Se ejecutan las acciones 5, 6 y 7 de la sección “Calcular calorías de la dieta, peso ideal e índice de masa corporal” y la acción 5 de la sección “Calcular valor del riesgo cardiovascular” del Caso de uso “Realizar cálculos”.
	7. El sistema genera el diagnóstico y tratamiento asociado al paciente en cuestión.
	8. El sistema guarda los cálculos auxiliares, el diagnóstico y el tratamiento.

	9. El sistema muestra los resultados de los cálculos auxiliares, el diagnóstico y tratamiento del paciente en cuestión y da la posibilidad de que el Médico indique sus observaciones o visualice la dieta correspondiente a las calorías calculadas.
10. El Médico realiza una de las siguiente operaciones: <ul style="list-style-type: none"> • Indicar sus observaciones • Visualizar la dieta correspondiente a las calorías. 	11. El sistema en dependencia de la operación seleccionada por el Médico realiza lo siguiente: <ul style="list-style-type: none"> • Si seleccionó indicar sus observaciones entonces el sistema registra las observaciones indicadas. • Si seleccionó visualizar la dieta correspondiente a las calorías entonces se ejecuta las acciones 10-12 de la sección “Calcular el peso ideal, índice de masa corporal y las calorías de la dieta” del caso de uso “Realizar Cálculos”.
Flujo Alterno	
Acciones del Actor	Respuesta del Sistema
	4.1. Si el Médico dejó campos vacíos, entonces el sistema emite un mensaje de alerta indicando los campos vacíos.
	5.1 Si el Médico indicó valores incorrectos, entonces el sistema emite un mensaje de alerta indicando los campos incorrectos.
10.1 Si el Médico no desea realizar ninguna de las operaciones sale de la sección.	
Prototipo	

Descripción del caso de uso: “Actualizar datos del paciente”.

Nombre del Caso de Uso	Actualizar Datos del Paciente
Actores	Médico
Propósito	Actualizar la primera consulta con los datos que aún no se hayan registrado.
Resumen	El CU se inicia cuando el sistema muestra la interfaz correspondiente para actualizar los datos del paciente que faltaban de la primera consulta. Luego del Médico indicar los datos a completar el sistema verifica que estén correctos y en caso positivo los guarda. En caso de que los datos no estén correctos entonces se muestra una alerta.
Referencias	R7
Precondiciones	Que el Médico esté autenticado en el sistema.
Poscondiciones	Se actualizaron los datos asociados a la primera consulta.
Prioridad	Secundario
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
	1.El sistema muestra la interfaz correspondiente para actualizar los datos del paciente y brinda la posibilidad de introducir los valores a los campos que faltaban.
2. EL Médico introduce los datos a actualizar. <ul style="list-style-type: none"> • Exámenes que no se hayan registrado aún. • Exámenes que estén incompletos. 	3. El sistema verifica que los campos siguientes estén llenos: En caso de actualizar el Examen físico <ul style="list-style-type: none"> - Talla - Peso - Presión arterial sistólica

<ul style="list-style-type: none"> - Antecedentes patológicos personales (Tiempo de padecimiento, Tiempo de evolución, Tiempo fumado, Hipertensión arterial (¿Desde cuándo?), Años transcurridos desde el primer infarto) - Examen físico (Todos los parámetros menos ¿Sigue algún tratamiento?) - Determinación de las dislipoproteinemias (Todos los parámetros) - Otros exámenes (Todos los parámetros) - Hemoquímica (Todos los parámetros) - Ultrasonido dopler carotideo (Engrosamiento del CIM Derecho, Engrosamiento del CIM Izquierdo, Extensión del CIM Derecho, Extensión del CIM Izquierdo, Por ciento de estenosis) - Nota (Se pueden modificar en todos los exámenes todos los parámetros que sean de listar y que no se haya interactuados con ellos) 	<ul style="list-style-type: none"> - Presión arterial diastólica <p>En caso de actualizar el examen</p> <p>Determinación de las lipoproteinemias</p> <ul style="list-style-type: none"> - Colesterol - Triglicéridos - cHDL - cLDL <p>En caso de actualizar Otros exámenes</p> <ul style="list-style-type: none"> - Glucemia en ayunas
	4. El sistema verifica que los datos introducidos sean correctos.
	5. El sistema guarda los datos actualizados.

	6. El sistema verifica si ya se calcularon los cálculos auxiliares y se generó el diagnóstico y tratamiento.
	7. Se ejecuta las acciones 13-20 del caso de uso “Registrar Datos del Paciente”.
Flujo Alterno	
Acciones del Actor	Respuesta del Sistema
	4.1. Si el Médico dejó campos vacíos entonces el sistema emite un mensaje de alerta indicando los campos vacíos.
	5.1 Si el Médico indicó valores incorrectos entonces el sistema emite un mensaje de alerta indicando los campos incorrectos.
Prototipo	

Descripción del caso de uso: “Autenticar usuario.”

Nombre del Caso de Uso	Autenticar Usuario
Actores	Médico, SAAA
Propósito	Permitir que sólo los Médicos registrados en el SAA puedan gestionar la información relacionada a la atención y seguimiento de los pacientes en el sistema.
Resumen	El Médico introduce su usuario y contraseña. Luego el SAAA verifica que es usuario esté autorizado y en caso positivo el sistema brinda acceso a las funcionalidades según los privilegios del usuario.
Referencias	R1

Precondiciones	Que un Médico solicite autenticarse.
Poscondiciones	Se le brinda acceso o no a las funcionalidades según los privilegios del usuario.
Prioridad	Crítico
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Médico indica datos para acceder al sistema: <ul style="list-style-type: none"> • Usuario • Contraseña 	2. El sistema solicita al registro SAAA que verifique si los datos introducidos son válidos
3. El SAAA indica que los datos son válidos.	4. El sistema verifica que el usuario tenga autorización para acceder.
	5. El sistema brinda acceso a las funcionalidades según los privilegios del Médico en cuestión.
Flujo Alterno	
Acciones del Actor	Respuesta del sistema.
3.1 El SAAA indica que los datos no son válidos.	3.2 El sistema indica un mensaje de error y no brinda acceso a las funcionalidades del sistema.
	5.1 Si el usuario no tiene acceso para acceder entonces el sistema muestra un mensaje de error.
Prototipo	

Descripción del caso de uso: “Solicitar traslado de paciente”.

Nombre del Caso de Uso	Solicitar Traslado de Paciente	
Actores	Médico	
Propósito	Solicitar el traslado de un paciente de una consulta para otra.	
Resumen	El CU se inicia cuando un Médico desea solicitar el traslado de un paciente para su consulta. El Médico indica buscar los pacientes existentes y luego indica cual es el paciente a solicitar su traslado. Luego el sistema registra la solicitud realizada por el Médico en cuestión.	
Referencias	R22	
Precondiciones	Que esté autenticado como Médico.	
Poscondiciones	Se realiza una solicitud de traslado de un paciente para otra consulta.	
Prioridad	Secundaria	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del sistema.	
1. El Médico indica realizar una solicitud de traslado de un paciente.	2. Se ejecuta el CU "Buscar pacientes" Nota: Los pacientes asociados al Médico que está realizando la petición de traslado no se muestran en el resultado de la búsqueda.	
3. El Médico indica el paciente a realizarle la solicitud de traslado.	4. El sistema pide confirmación de si se desea trasladar al paciente.	
5. El Médico indica que si desea realizar la solicitud de traslado.	6. El sistema registra la solicitud de traslado de ese paciente.	
	7. El sistema muestra un mensaje informando que la solicitud se ha registrado.	
Flujo Alterno		

Acciones del Actor	Respuesta del sistema.
5.1 Si el Médico no desea realizar la solicitud de traslado entonces sale de la sección.	
Prototipo	

Descripción del caso de uso: **“Trasladar pacientes.”**

Nombre del Caso de Uso	Trasladar Pacientes
Actores	Administrador_traslado
Propósito	Autorizar el traslado de un paciente para otra consulta.
Resumen	El CU se inicia cuando el Administrador de sistema indica realizar el traslado de los pacientes pendientes a trasladar para otra consulta.
Referencias	R23
Precondiciones	Que esté autenticado como Administrador de sistema.
Poscondiciones	Se autorizó o no el traslado de un paciente para otra consulta.
Prioridad	Secundaria
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Administrador de sistema indica trasladar paciente que están pendientes de traslado.	2. El sistema muestra un listado de los paciente a los que se les realizado una solicitud de traslado y todavía no se han revisado.

	<ul style="list-style-type: none"> • Paciente • Carnet de identidad • Médico al que pertenece • Médico que solicita el traslado <p>El sistema da la posibilidad de aceptar o rechazar el traslado de dicho paciente.</p>
3. El Administrador de sistema indica aceptar el traslado de un paciente determinado.	4. El sistema muestra un mensaje de confirmación de traslado de dicho paciente.
5. El Administrador de sistema indica que desea trasladar al paciente.	6. El sistema le asigna el paciente al médico que realizó la solicitud y le niega el acceso al médico al cual pertenecía. Dicho médico no puede ver los datos del antiguo paciente una vez realizada la operación.
	7 El sistema actualiza el listado de solicitudes pendientes.
Flujo Alterno 1	
Acciones del Actor	Respuesta del sistema.
3.1 El Administrador de sistema indica rechazar el traslado de un paciente determinado.	3.2. El sistema muestra un mensaje de confirmación de traslado de dicho paciente.
3.3 El Administrador de sistema indica que desea rechazar el traslado del paciente.	3.4. El sistema no le asigna el paciente al médico que realizó la solicitud y actualiza el listado de solicitudes pendientes.
5.1 El Administrador de sistema indica que no desea trasladar al paciente y entonces sale de la sección.	
Flujo Alterno 2	
Acciones del Actor	Respuesta del sistema.
3.4.1 El Administrador de sistema indica	

que no desea rechazar el traslado del paciente y entonces sale de la sección.	
Prototipo	

Descripción del caso de uso: “Buscar documentos.”

Nombre del Caso de Uso	Buscar Documentos
Actores	Usuario
Propósito	Buscar y visualizar documentos existentes.
Resumen	El CU inicia cuando el Usuario desea buscar los documentos existentes. Luego el sistema muestra una interfaz con el listado de los documentos y sus datos.
Referencias	R18, R19
Precondiciones	Que el usuario indique buscar los documentos existentes.
Poscondiciones	Se buscó y visualizó un listado con los datos de los medicamentos existentes.
Prioridad	Secundario
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Usuario indica buscar los documentos existentes.	2. El sistema busca los documentos existentes.
	3. El sistema muestra un listado de los documentos existentes en una tabla. La tabla contiene los siguientes campo: <ul style="list-style-type: none"> • Título • Autor

	<ul style="list-style-type: none"> • Año • Descripción <p>Además da la posibilidad de ver el contenido de un documento.</p>
4. El Usuario indica ver el contenido de un documento.	5. El sistema muestra el contenido del documento indicado.
Flujo Alterno	
Acciones del Actor	Respuesta del Sistema
4.1 El Usuario no indica ver el contenido de ningún documento y sale de la sección.	
Prototipo	

Descripción del caso de uso: "Gestionar documentos."

Nombre del Caso de Uso	Gestionar Documentos
Actores	Médico_administrador_general
Propósito	Insertar, actualizar o eliminar documentos
Resumen	El CU inicia cuando el Administrador general desea insertar, actualizar o eliminar un documento. Luego de que el administrador general indique los datos del documento a insertar, actualizar o eliminar entonces el sistema registra, actualiza o elimina el medicamento correspondientemente.
Referencias	R17, R20, R21
Precondiciones	Que el Administrador general esté autenticado en el sistema.

Poscondiciones	Se actualizó el listado de documentos
Prioridad	Secundario
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Administrador general indica gestionar documentos.	2. Se ejecuta la acción 2 del caso de uso "Buscar Documentos".
	3. El sistema muestra un listado de los documentos existentes en una tabla. La tabla contiene los siguientes campo: <ul style="list-style-type: none"> • Título • Autor • Año • Descripción Además da la posibilidad de ver el contenido de un documento, crear un nuevo documento, actualizar sus datos o eliminarlo.
4. El Administrador general realiza una de las siguientes operaciones: <ul style="list-style-type: none"> • Ver contenido de un documento. • Insertar un nuevo documento. • Eliminar un documento existente. • Actualizar un documento. 	5. El sistema en dependencia de la operación indicada por el administrador general realiza lo siguiente: <ul style="list-style-type: none"> • Si el Administrador general va a ver el contenido de un documento entonces el sistema muestra el contenido del documento indicado. • Si el administrador general va a insertar un nuevo documento entonces se ejecuta la sección "Insertar documentos". • Si el Administrador general va a eliminar un documento existente entonces se ejecuta la sección "Eliminar documento". • Si el Administrador general va a

	actualizar un documento entonces se ejecuta la sección "Actualizar documento".
Sección "Insertar Documentos".	
Acciones del Actor	Respuesta del Sistema
1. El Administrador general indica los datos del documento e indica insertarlo <ul style="list-style-type: none"> • Título del documento • Nombre del Autor • Año • Descripción • Archivo 	2. El sistema verifica que los siguientes campos estén llenos: <ul style="list-style-type: none"> • Título del documento • Archivo
	3. El sistema verifica que los datos introducidos sean correctos.
	4. El sistema verifica que el documento adjuntado no sobrepase los 5 MG.
	5. El sistema verifica que no exista un documento con el mismo nombre de archivo.
	6. El sistema registra el documento indicado por el Administrador general.
	7. El sistema muestra el listado de documentos actualizado.
Flujo Alternativo Sección "Insertar Documentos".	
Acciones del Actor	Respuesta del Sistema
	3.1. Si el Administrador general dejó campos vacíos entonces el sistema emite un mensaje de alerta indicando que los campos que están vacíos.
	4.1. Si el Administrador general introdujo un valor incorrecto entonces el sistema emite un

	mensaje de alerta indicando error.
	5.1 Si el tamaño del documento adjuntado excede los 5 MG entonces el sistema muestra un mensaje de alerta.
	6.1 Si el nombre del documento indicado coincide con el nombre de un documento existente entonces se muestra un mensaje de alerta indicando ese documento ya existe.
Prototipo Sección “Insertar Documentos”	
Sección “Eliminar Documentos”.	
Acciones del Actor	Respuesta del Sistema
1. El Administrador general indica del listado mostrado el documento que desea eliminar.	2. El sistema muestra un mensaje de alerta indicando si desea eliminar el documento.
3. El Administrador general indica que desea eliminar el documento.	4. El sistema elimina el documento indicado por el administrador general.
	5. El sistema muestra el listado de documentos actualizado.
Flujo Alternativo Sección “Eliminar Documentos”.	
Acciones del Actor	Respuesta del Sistema
3.1 Si el Administrador general indica que no desea eliminar el documento y se ejecuta la acción 1 del Curso Normal de Eventos.	
Prototipo Sección “Eliminar Documentos”	

Sección “Actualizar Documentos”	
Acciones del Actor	Respuesta del Sistema
1. El Administrador general indica el documento que desea actualizar.	2. El sistema da la posibilidad de actualizar los campos correspondientes al documento indicado. <ul style="list-style-type: none"> • Título del documento • Nombre del Autor • Año • Descripción • Archivo
3. El Administrador general registra los campos a modificar e indica modificarlos.	4. El sistema verifica que los campos necesarios estén llenos: <ul style="list-style-type: none"> • Título del documento • Archivo
	5. El sistema verifica que los datos introducidos sean correctos.
	6. El sistema verifica que el documento adjuntado no sobrepase los 5 MG.
	7. El sistema verifica que no exista un documento con el mismo nombre de archivo.
	8. El sistema actualiza los datos indicados del documento.
	9. El sistema muestra el listado de documentos actualizado.
Flujo Alternativo Sección “Actualizar Documentos”.	
Acciones del Actor	Respuesta del Sistema
	3.1. Si el Administrador general dejó campos vacíos entonces el sistema emite un mensaje de alerta indicando que los campos

	están vacíos.
	4.1. Si el Administrador general introdujo un valor incorrecto entonces el sistema emite un mensaje de alerta indicando error.
	5.1 Si el tamaño del documento adjuntado excede los 5 MG entonces el sistema muestra un mensaje de alerta.
	6.1 Si el nombre del documento indicado coincide con el nombre de un documento existente entonces se muestra un mensaje de alerta indicando ese documento ya existe.
Prototipo Sección “Actualizar Documentos”	

Descripción del caso de uso: “Administrar usuarios.”

Nombre del Caso de Uso	Administrar Usuarios
Actores	Administrador_usuario
Propósito	Buscar, visualizar, eliminar un usuario y asignar permisos según el nivel de acceso del mismo.
Resumen	El CU inicia cuando el Administrador del sistema desea visualizar, eliminar un usuario o asignar permisos según el nivel de acceso del mismo. Luego el sistema muestra un interfaz con todos los usuarios y da la posibilidad de eliminarlo o asignarle el nivel de acceso. Los usuarios que saldrán en el listado serán según el nivel de acceso del administrador de sistema.
Referencias	R24, R25, R26, R27
Precondiciones	Que el Administrador de sistema esté autenticado en el

	sistema.
Poscondiciones	Se eliminó o se asignó permisos a un usuario.
Prioridad	Crítico
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Administrador de sistema indica administrar usuarios.	<p>2. El sistema busca los usuarios existentes según el nivel de acceso del Administrador de sistema logueado:</p> <ul style="list-style-type: none"> • Administrador de sistema nacional: todos los usuarios. • Administrador de sistema provincial: todos los usuarios que su provincia se corresponda con la provincia del administrador de sistema provincial logueado y que tengan un nivel inferior o igual. • Administrador de sistema municipal: todos los usuarios que su municipio se corresponda con el municipio del administrador de sistema municipal logueado y que tengan un nivel inferior o igual.
	<p>3. El sistema muestra un listado de los usuarios encontrados en una tabla. La tabla contiene los siguientes campos:</p> <ul style="list-style-type: none"> • Usuario • Nombre • Registro • Miembro desde • Visitado <p>Además da la posibilidad de actualizar o eliminar usuario.</p>

	También permite visualizar todos los usuarios o solamente aquellos que no estén activos.
<p>4. El Administrador de sistema realiza una de las siguientes operaciones:</p> <ul style="list-style-type: none"> • Actualizar los datos del usuario • Eliminar un usuario • Visualizar todos los usuarios • Visualizar los usuarios no activos 	<p>5. El sistema en dependencia de la acción seleccionada por el Administrador de sistema realiza lo siguiente:</p> <ul style="list-style-type: none"> • Si seleccionó actualizar los datos del usuario se ejecuta la sección “Actualizar datos del usuario”. • Si seleccionó eliminar un usuario se ejecuta la sección “Eliminar Usuario” • Si seleccionó visualizar todos los usuarios entonces se muestra el listado de todos los usuarios existentes. • Si seleccionó visualizar los usuarios no activados entonces se muestran solamente los usuarios que no estén activados.
Sección “Actualizar datos del usuario”	
Acciones del Actor	Respuesta del sistema.
	<p>1. El sistema muestra la interfaz correspondiente a la actualización de los datos del usuario. Los siguientes campos contienen valores:</p> <ul style="list-style-type: none"> • Nombre • Registro • Es Médico • Provincia • Municipio • Miembro desde • Visitado
<p>2. El Administrador de sistema indica los datos a actualizar:</p> <ul style="list-style-type: none"> • Nivel (Doctor, Nacional, Provincial y 	<p>3. El sistema verifica los permisos asignados al usuario</p>

Municipal)	
<ul style="list-style-type: none"> • ¿Está activo? (Si o No) • Permiso (administradorgeneral, administradordesistema, doctor) 	
	3. El sistema registra los datos indicados a actualizar.
	4. Se ejecuta la acción 1 de la sección "Actualizar datos del usuario"
Flujo Alternativo "Actualizar datos del usuario"	
Acciones del Actor	Respuesta del sistema.
2.1 Si el Administrador general no desea actualizar ningún dato sale de la sección.	
Sección "Eliminar Usuario".	
Acciones del Actor	Respuesta del Sistema
	1. El sistema verifica si el usuario es médico.
	2. El sistema verifica que el usuario no tenga pacientes registrados hasta el momento.
	3. El sistema muestra un mensaje de confirmación.
4. El Administrador de sistema indica que desea eliminar al usuario.	5. El sistema elimina el usuario indicado por el Administrador de sistema.
	6. El sistema muestra el listado de usuarios actualizado.
Flujo Alternativo Sección "Eliminar usuario".	
Acciones del Actor	Respuesta del Sistema
	2.1 Si el usuario no es médico entonces el sistema verifica si está activo.

	2.2 Se ejecuta las acciones 3-6 de flujo normal de la sección “Eliminar Usuario”.
	3.1 Si el usuario tiene pacientes asociados entonces el sistema muestra una alerta de que este usuario no se puede eliminar porque tiene datos asociados.
Prototipo	

Descripción del caso de uso: “Registrar usuario”.

Nombre del Caso de Uso	Registrar Usuario
Actores	Usuario
Propósito	Registrar los datos de un usuario en el sistema.
Resumen	El CU inicia cuando el Usuario desea registrarse en el sistema e introduce los datos necesarios para el registro. Luego el sistema registra los datos indicados por el usuario y lo ubica pendiente de autorización para acceso
Referencias	R28
Precondiciones	Que el Usuario indique registrarse en el sistema.
Poscondiciones	Se registraron los datos de un usuario en el sistema.
Prioridad	Crítico
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del sistema.
1. El Usuario indica registrarse.	2. El sistema muestra la interfaz correspondiente al registro de un Usuario.

<p>3. El Usuario indica los datos para el registro:</p> <ul style="list-style-type: none"> • Usuario • Contraseña • Nombre y Apellidos • ¿Es médico? • Provincia • Municipio 	<p>4. El sistema verifica que se haya indicado valor para los parámetros necesarios:</p> <ul style="list-style-type: none"> • Todos los campos son obligatorios menos el campo “Registro profesional”, que sólo sería obligatorio también si Usuario indica que es médico.
	<p>5. El sistema verifica que no se haya registrado anteriormente en el sistema un Usuario con el mismo “Usuario”</p>
	<p>6. El sistema verifica si el Usuario indicó que era médico.</p>
	<p>7. El sistema genera automáticamente el registro del Usuario. El registro médico se genera de la siguiente forma:</p> <ul style="list-style-type: none"> • fecha de registro+1
	<p>8. El sistema registra los datos del Usuario y lo ubica en estado de espera de autorización de acceso.</p>
	<p>9. El sistema muestra un mensaje de confirmación del registro del usuario.</p>
Flujo Alterno	
Acciones del Actor	Respuesta del Sistema
	<p>5.1 Si no se indicó valor para alguno de los campos el sistema muestra un mensaje de error.</p>
	<p>6.1 Si existe un usuario registrado con el mismo nombre o registro profesional indicados el sistema muestra un mensaje de alerta.</p>

	7.1 Si el Usuario indicó que no era médico entonces el sistema ejecuta la acción 8 del Flujo Normal de los Eventos.
Prototipo	

Anexo 2. Diagramas de clase del diseño.

Diagrama de clase del CU: “Buscar Paciente”

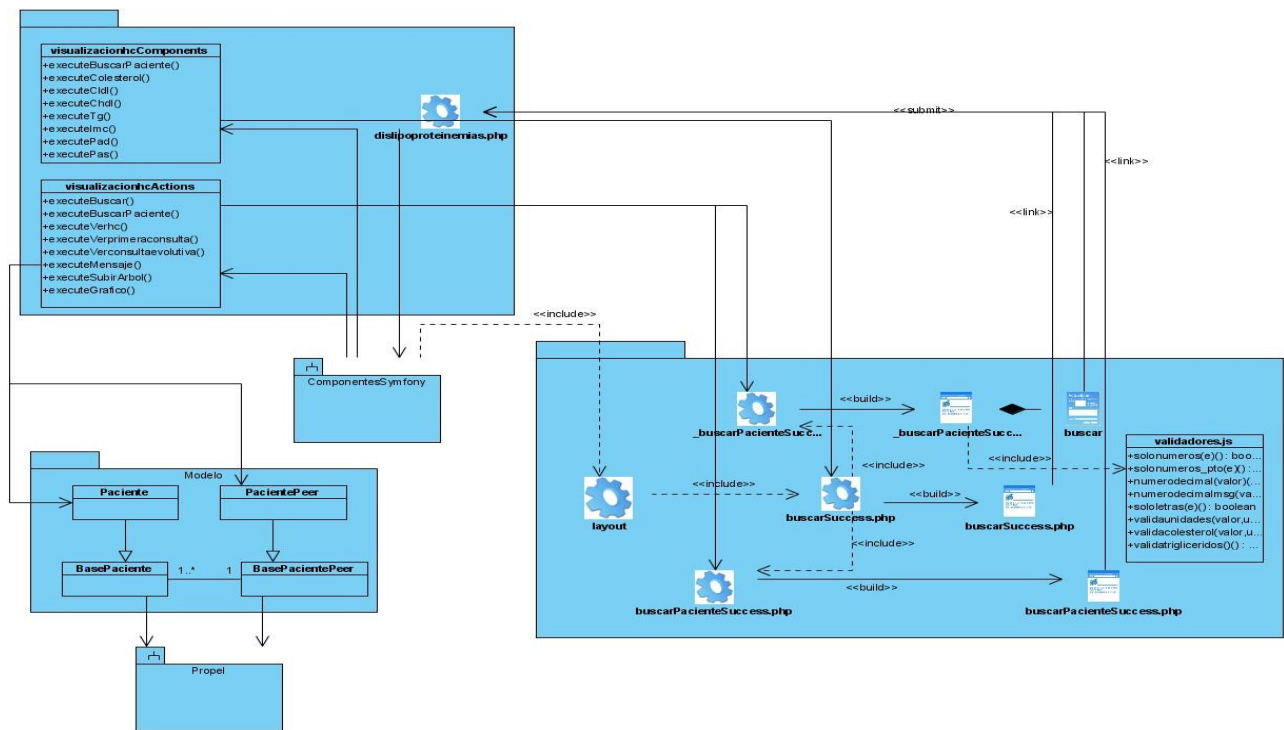


Diagrama de clase del CU: "Realizar calculos".

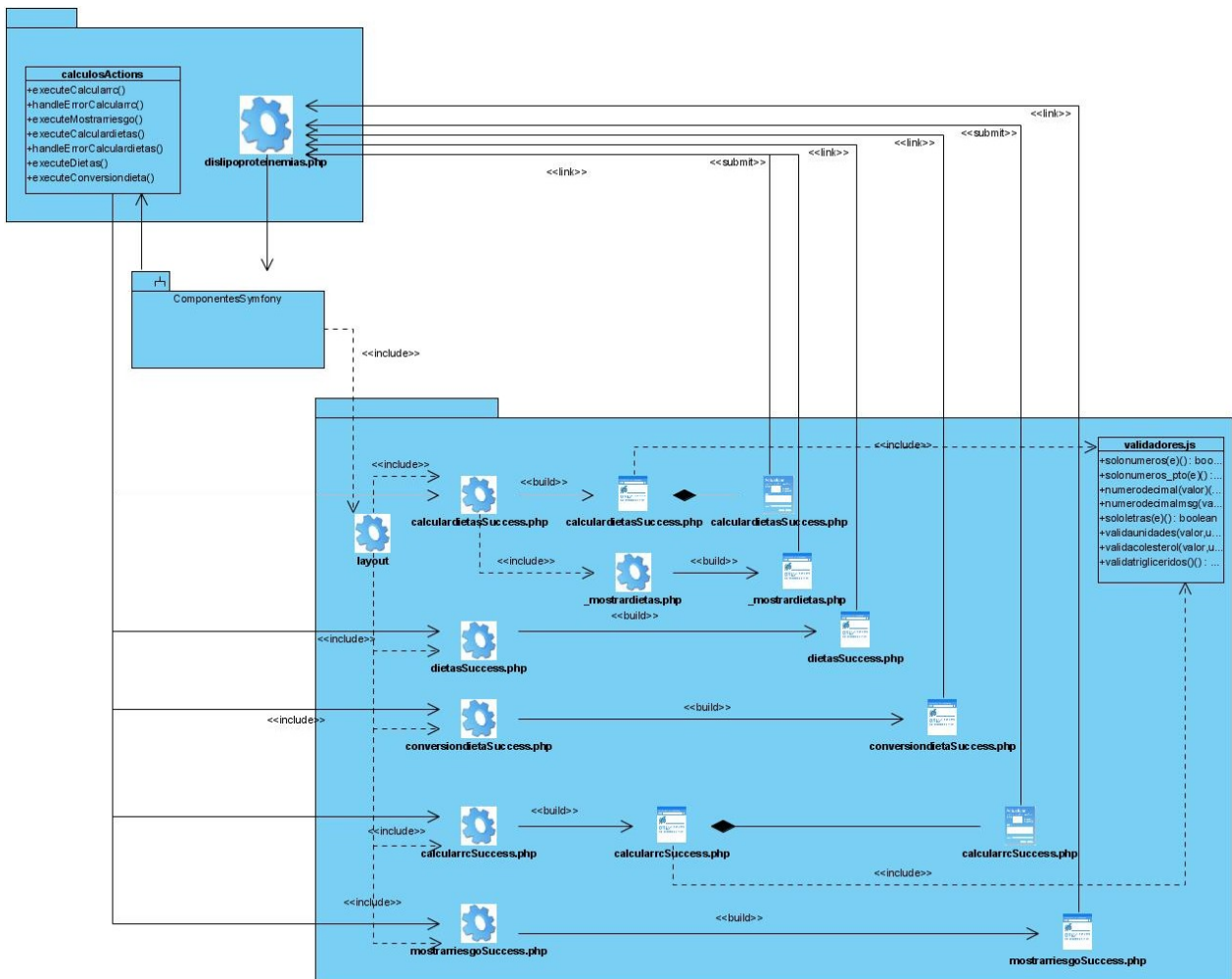
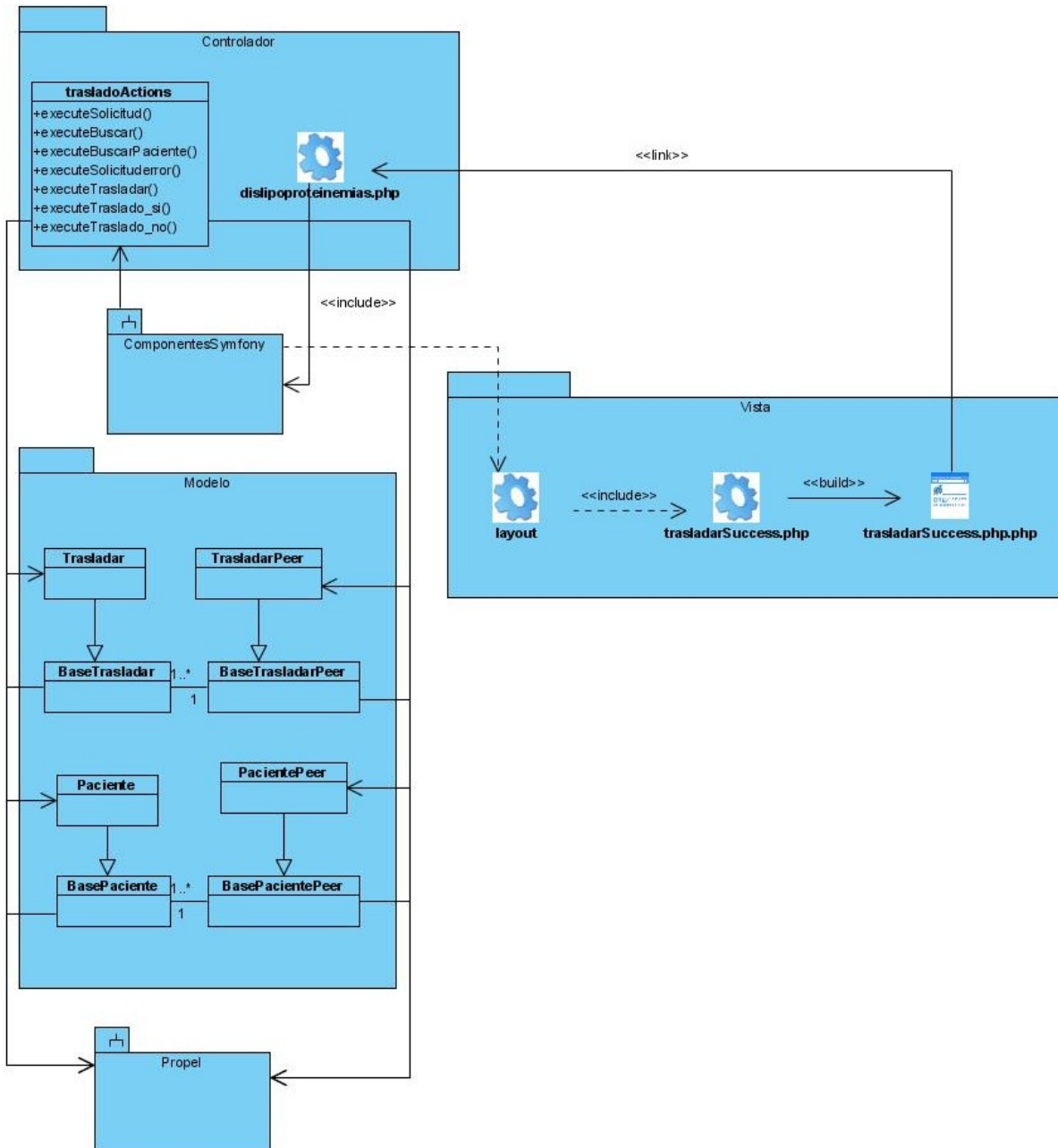


Diagrama de clase del CU: "Trasladar paciente".



Anexo 3 Descripción de las clases que contienen la lógica de negocio del sistema.

Descripción de la clase: “completarActions”

Nombre de la Clase: completarActions
Nombre del método: executeIndex()
Descripción: Este método se encarga de verificar en la base de datos el completamiento o no de todos los exámenes algo fundamental para el cumplimiento correcto del módulo completar, de los exámenes ya completados se encarga de visualizar sus datos y de los que no da la opción de realizarlos, además una vez completados los exámenes genera de forma dinámica los cálculos correspondientes y un diagnostico y tratamiento en dependencia de los mismos y los exámenes insertados.
Nombre del método: executeActualizarapp()
Descripción: Este método tiene dos variantes la primera es que nunca se haya realizado el examen de Antecedentes Patológicos Personales en este caso se insertaran los datos nuevos, la segunda variante es cuando ya este examen se realizo entonces dará la opción de insertar los campos que se quedaron vacios la vez de la inserción anterior. Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes.
Nombre del método: executeActualizarApf()
Descripción: Este método tiene dos variantes la primera es que nunca se haya realizado el examen de Antecedentes Patológicos familiares en este caso se insertaran los datos nuevos, la segunda variante es cuando ya este examen se realizo entonces dará la opción de insertar los campos que se quedaron vacios la vez de la inserción anterior. Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes.
Nombre del método: executeActualizarexamenfisico()
Descripción: Este método tiene dos variantes la primera es que nunca se haya realizado el examen físico en este caso se insertaran los datos nuevos, la segunda variante es cuando ya este examen se realizo entonces dará la opción de insertar los campos que se

quedaron vacios la vez de la inserción anterior.

Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes.

Nombre del método: executeActualizarsignos

Descripción: Este método tiene dos variantes la primera es que nunca se haya realizado el examen de Signos Asociados a las Dislipoproteinemias en este caso se insertaran los datos nuevos, la segunda variante es cuando ya este examen se realizo entonces dará la opción de insertar los campos que se quedaron vacios la vez de la inserción anterior.

Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes.

Nombre del método: executeActualizarlipo()

Descripción: Este método tiene dos variantes la primera es que nunca se haya realizado el examen de Determinación de las Lipoproteinemias en este caso se insertaran los datos nuevos, la segunda variante es cuando ya este examen se realizo entonces dará la opción de insertar los campos que se quedaron vacios la vez de la inserción anterior.

Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes.

Nombre del método: executeActualizarotros()

Descripción: Este método tiene dos variantes la primera es que nunca se haya realizado el examen Otros exámenes en este caso se insertaran los datos nuevos, la segunda variante es cuando ya este examen se realizo entonces dará la opción de insertar los campos que se quedaron vacios la vez de la inserción anterior.

Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes.

Nombre del método: executeActualizarhemoquimica()

Descripción: Este método tiene dos variantes la primera es que nunca se haya realizado el examen Hemoquímica en este caso se insertaran los datos nuevos, la segunda variante es cuando ya este examen se realizo entonces dará la opción de insertar los campos que se quedaron vacios la vez de la inserción anterior.

Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes.
Nombre del método: executeActualizarultrasonido()
Descripción: Este método tiene dos variantes la primera es que nunca se haya realizado el examen de Ultrasonido doppler carotideo en este caso se insertaran los datos nuevos, la segunda variante es cuando ya este examen se realizo entonces dará la opción de insertar los campos que se quedaron vacios la vez de la inserción anterior. Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes.
Nombre del método: executeActualizararmetos()
Descripción: Este método da la opción de insertar los medicamentos que está tomando el paciente al momento de realizarse el completar de la primera consulta. Nota: Por petición del cliente no se pueden actualizar los datos ya insertados solo completar los exámenes. Para el caso de los medicamentos tomados solo se insertaran una vez.
Nombre del método: executeCompletardiagnostico()
Descripción: Este método muestra el diagnostico y el tratamiento recomendado para este paciente calculado en el método executeindex() y da la opción al médico de insertar observaciones propias.

Descripción de la clase: “**calculosActions**”

Nombre de la clase: calculosActions
Nombre del método: executeCalcularrc()
Descripción: Este método permite realizar a través de la clase Riesgo cardiovascular los cálculos correspondientes al riesgo cardiovascular según la tabla de riesgos de Framingham y redirecciona para la página mostrarriesgo con todos los valores necesarios para mostrar los mismos.
Nombre del método: handleErrorCalcularrc()
Descripción: Se encarga del tratamiento de errores de la página calcularrc.

Nombre del método: executeMostrarriesgo()
Descripción: Este método recibe los datos del método executeCalcularrc() y como hace referencia su nombre muestra el resultado de los cálculos del riesgo cardiovascular.
Nombre del método: executeCalculardietas()
Descripción: Según el paso y la talla este método realiza a través de la clase Dietas el cálculo correspondiente al índice de masa corporal, al peso ideal máximo , al peso ideal mínimo y a las dietas calóricas , además según el valor de las dietas calóricas el mismo las incluye en un rango de mínimo y máximo para no ser tan explícito a la hora de dar posibles opciones de dietas.
Nombre del método: handleErrorCalculardietas()
Descripción: Se encarga del tratamiento de errores de la página Calculardietas.
Nombre del método: executeDietas()
Descripción: Se encarga de ejecutar la página dietas del módulo herramientas en dependencia de la dieta necesaria esta será la que se observe.
Nombre del método: executeConversiondieta()
Descripción: Se encarga de ejecutar la página conversiondieta del módulo herramientas en dependencia de la dieta necesaria esta será la que se observe.

Descripción de la clase: “**inserciónhcActions**”

Nombre de la clase: insercionhcActions
Nombre del método: executeIndex()
Descripción: Este método se encarga de la visualización de la página principal del módulo inserción, en el mismo se verifica si el examen ya se insertó, en caso positivo le imposibilita al usuario volver a insertarlo mostrándole los datos insertados, además de forma dinámica controla la realización de los cálculos los cuales no se ejecutan hasta no se completen todos los exámenes necesarios para los mismos.
Nombre del método: executeInsertarpaciente()
Descripción: Este método se encarga como dice su nombre de insertar un paciente en la base de datos para esto primero verifica que no se encuentre en la misma, una vez

verificado lo mismo en caso de encontrarse lanza un error y en caso negativo lo inserta, le crea una consulta y luego lo redirecciona a la pagina index donde estarán todos los exámenes del paciente.
Nombre del método: handleErrorInsertarpaciente()
Descripción: Se encarga del tratamiento de errores de la página insertar paciente.
Nombre del método: executeInsertarapp()
Descripción: Este método se encarga de insertar los datos del examen Antecedentes Patológicos Personales en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: executeInsertarapf()
Descripción: Este método se encarga de insertar los datos del examen Antecedentes Patológicos Familiares en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: executeInsertarexamenfisico
Descripción: Este método se encarga de insertar los datos del examen físico en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: handleErrorInsertarexamenfisico()
Descripción: Se encarga del tratamiento de errores de la página insertar examen físico.
Nombre del método: executeInsertarsignos()
Descripción: Este método se encarga de insertar los datos del examen Signos Asociados a las Dislipoproteinemias en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: executeInsertarlipo()
Descripción: Este método se encarga de insertar los datos del examen Determinación de las Lipoproteinemias en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: executeInsertarotros()
Descripción: Este método se encarga de insertar los datos del examen Otros Exámenes

en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: handleErrorInsertarotros()
Descripción: Se encarga del tratamiento de errores de la página insertar otros exámenes.
Nombre del método: executeInsertarmedicamentos
Descripción: Este método se encarga de insertar los datos del examen Medicamentos en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: executeInsertarhemoquimica
Descripción: Este método se encarga de insertar los datos del examen Hemoquímica en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: executeInsertarultrasonido()
Descripción: Este método se encarga de insertar los datos del examen Ultrasonido por doppler carotideo en la base de datos, además actualiza la tabla exámenes_realizados con la realización del mismo.
Nombre del método: executeDiagnostico()
Descripción: Este método se encarga de forma automática y dinámica de generar un diagnostico y un tratamiento de acuerdo a los exámenes insertados y los cálculos anteriormente realizados, los mismo se guardan en la base de datos para una consulta posterior. El método también le da la opción al médico de insertar observaciones propias
Nombre del método: executeConsultaevolutiva()
Descripción: Para la ejecución de la consulta evolutiva es imprescindible la inserción de los exámenes necesarios para poder completar la primera consulta, de forma dinámica este método verifica que los mismos estén insertados en caso negativo redirecciona para una página de error, en el caso positivo inserta la nueva consulta con los datos nuevos en la base de datos.
Nombre del método: executediagnosticoevolutiva()
Descripción: Este método realiza y muestra los cálculos realizados con los datos

insertados por el paciente en la consulta evolutiva y genera un diagnostico y tratamiento para este paciente, además de dar la opción de visualizar los mismo.
Nombre del método: executeDiagnosticoevolutiva()
Descripción: Este método muestra y genera un diagnostico y tratamiento para el paciente en cuestión además de dar la opción de insertar observaciones propias del doctor.
Nombre del método: handleErrorGestionarmedicamentos()
Descripción: Se encarga del tratamiento de errores de la página gestionar medicamentos.
Nombre del método: executeEliminarmedicamentos()
Descripción: Este método como plantea su nombre no hace más que según el id del medicamento lo elimina de la base de datos.
Nombre del método: executeGestionarmedicamentos()
Descripción: Este método inserta el medicamento en la base de datos.
Nombre del método: executeGestionarmed()
Descripción: Este método permite visualizar todos los medicamentos existentes en la base de datos, además verifica que no se repitan.

Anexo 4 “Descripción de las tablas de la base de datos”.

Descripción de la tabla: “antecedentes_patologicos_familiares”.

Nombre: antecedentes_patologicos_familiares		
Descripción: Tabla que almacena los datos asociados a los antecedentes patológicos familiares de un paciente.		
Atributo	Tipo	Descripción
id_apf	int	Identificador de la tabla “antecedentes_patologicos_familiares”.
dislipoproteinemias_primera	int	Dislipoproteinemias (Familiares 1ra línea).

dislipoproteinemias_segunda	int	Dislipoproteinemias (Familiares 2da línea).
diabetis_mellitus_primera	int	Diabetes Mellitus (Familiares 1ra línea).
diabetis_mellitus_segunda	int	Diabetes Mellitus (Familiares 2da línea).
infarto_padre_hermanos	int	Infarto del corazón o muerte súbita (Padre o hermanos menores 55 años).
infarto_madre_hermanas	int	Infarto del corazón o muerte súbita (Madre o hermanas menores 65 años).
hipertension_familiar	int	Hipertensión arterial (Familiares 1ra línea).
id_consulta	int	Identificador de la tabla "consulta" (llave foránea).

Descripción de la tabla: "antecedentes_patologicos_familiares".

Nombre: antecedentes_patologicos_personales		
Descripción: Tabla que almacena los datos asociados a los antecedentes patológicos personales de un paciente.		
Atributo	Tipo	Descripción
id_app	int	Identificador de la tabla "antecedentes_patologicos_personales".
dislipoproteinemias	int	Dislipoproteinemias (Si o No).
tiempo_padecimiento_dislipoproteinemias	int	Tiempo de padecimiento (años).
alcoholismo	int	Ingestión de bebidas alcohólicas (Nunca, Diaria o Semanal, Ocasional).
diabetis_mellitus	int	Diabetes Mellitus (No, Tipo 1, Tipo 2).
tiempo_evolucion	int	Tiempo de evolución (años).

hipertension_arterial	int	Hipertensión arterial (Desde cuando?) (años).
hipertension_arterial_tratada	int	Tratamiento actual (Si o No).
tabaquismo_clasificacion	int	Tabaquismo (Nunca fumó, Exfumador, Fumador actual).
tabaquismo_anno	int	Tiempo fumando (años).
amaurosis_fugaz	int	Otros (Amaurosis Fugaz).
pancreatitis	int	Otros (Pancreatitis).
cirrosis_biliar	int	Otros (Cirrosis biliar).
ataque_transitorio	int	Otros (Ataque transitorio de isquemia).
aneurisma_aorta_abdominal	int	Otros (Aneurisma de la aorta abdominal).
claudicacion_intermitente	int	Otros (Claudicación intermitente).
infarto_cerebral	int	Otros (Infarto cerebral isquémico).
infarto_miocardio	int	Cardiopatía isquémica (Infarto del miocardio).
infarto_miocardio_anno	int	Cardiopatía isquémica (Año(s) del primero).
angina_estable	int	Cardiopatía isquémica (Angina estable).
angina_inestable	int	Cardiopatía isquémica (Angina inestable).

angioplastia_coronaria	int	Cardiopatía isquémica (Angioplastia coronaria).
bypass_coronario	int	Cardiopatía isquémica (Bypass coronario).
enfermedades_renales_cronicas	int	Cardiopatía isquémica (Enfermedades renales crónicas).
otros_sindromes_coronarios	entero	Cardiopatía isquémica (Otros síndromes coronarios).
id_consulta	int	Identificador de la tabla “consulta” (llave foránea).

Descripción de la tabla: “consulta”.

Nombre: consulta		
Descripción: Tabla que almacena los datos coincidentes de la 1ra Consulta y la Consulta Evolutiva.		
Atributo	Tipo	Descripción
id_consulta	int	Identificador de la tabla “consulta”
tipo_consulta	int	Tipo de la Consulta (1ra Consulta o Consulta Evolutiva).
created_at	datetime	Fecha de registro de la consulta.
diagnostico	text	Diagnóstico asociado a un paciente para cada consulta.
diagnostico_clasificacion	text	Clasificación del diagnóstico asociado a un paciente para cada consulta (Riesgo Bajo, Riesgo Moderado, Riesgo Alto).
tratamiento_recomendaciones	text	Tratamiento asociado a un paciente para cada consulta.

observaciones	text	Observaciones asociado a un paciente para cada consulta.
historia_clinica	char	Identificador de la tabla "paciente" (Llave foránea).

Descripción de la tabla: "consulta_evolutiva".

Nombre: consulta_evolutiva		
Descripción: Tabla que almacena los datos asociados a la Consulta Evolutiva realizada al paciente.		
Atributo	Tipo	Descripción
id_consulta_evolutiva	int	Identificador de la tabla "consulta"
persion_arterial_sistolica	int	Presión arterial Sistólica
colesterol	float	Colesterol
triglicéridos	float	Triglicéridos
chdl	float	cHdl
cldl	float	cLdl
otros	varchar	Otros exámenes
id_consulta	int	Identificador de la tabla "consulta" (llave foránea).
presion_arterial_diasistolica	int	Presión arterial diasistólica

peso	float	Peso corporal del paciente.
------	-------	-----------------------------

Descripción de la tabla: “exámenes_realizados”.

Nombre: exámenes_realizados		
Descripción: Tabla que almacena exámenes que el paciente ya tiene realizado completamente.		
Atributo	Tipo	Descripción
id_consulta	int	Identificador de la tabla “paciente” (Llave foránea).
App	int	Antecedentes Pátológicos Personales
apf	int	Antecedentes Patológicos Familiares
calculos	int	Cálculos Auxiliares
hemoquimica	int	Hemoquímica
lipidograma	int	Lipidograma
ultrasonido	int	Ultrasonido
examen_fisico	int	Examen físico
otros_exámenes	int	Otros exámenes
sintomas	int	Síntomas

medicamentos_tomados	int	Medicamentos
----------------------	-----	--------------

Descripción de la tabla: "lipidograma".

Nombre: lipidograma		
Descripción: Tabla que almacena los datos asociados al examen de Lipidograma pertenecientes al paciente.		
Atributo	Tipo	Descripción
id_lipidograma	int	Identificador de la tabla "lipidograma".
apo_a1	float	ApoA1.
apo_b	float	ApoA1
lp	float	Lp(a)
electroforesis	varchar	Electroforesis.
isomorfismo_apo_e	int	Isomorfo de ApoE.
id_consulta	int	Identificador de la tabla "consulta" (Llave foránea).

Descripción de la tabla: "paciente".

Nombre: paciente		
Descripción: Tabla que almacena los datos asociados a los pacientes registrados en el sistema.		
Atributo	Tipo	Descripción

historia_clinica	int	Identificador de la tabla "paciente".
nombre	varchar	Nombre.
primer_apellido	varchar	Primer apellido.
segundo_apellido	varchar	Segundo apellido.
sexo	int	Sexo.
edad	int	Edad.
direccion_particular	varchar	Dirección particular.
telefono	int	Teléfono.
pais	varchar	País.
id_municipio	int	Identificador de la tabla "municipio" (Llave foránea).
id_provincia	int	Identificador de la tabla "provincia" (Llave foránea).
codigo_medico	varchar	Identificador de la tabla "medico" (Llave foránea).
archivo	varchar	Archivo del árbol genealógico

Descripción de la tabla: "síntomas".

Nombre: síntomas

Descripción: Tabla que almacena los datos asociados a los síntomas.		
Atributo	Tipo	Descripción
id_sintomas	int	Identificador de la tabla "síntomas".
dolor_abdominal_angina_mesenterica	int	Dolor abdominal angina mesentérica.
xantomas_tuberosos	int	Xantomas tuberosos.
xantomas_tuberosos_localizacion	varchar	Xantomas tuberosos localización.
xantomas_cutaneos_eruptivos	int	Xantomas cutáneos eruptivos.
xantomas_cutaneos_eruptivos_localizacion	varchar	Xantomas cutáneos eruptivos localización.
xantomas_tuberosos_tendones_aquiles	int	Xantomas tuberosos tendones Aquiles.
xantomas_tuberosos_extensores_mano_derecha	int	Xantomas tuberosos extensores mano derecha.
xantomas_tuberosos_extensores_mano_izquierda	int	Xantomas tuberosos extensores mano izquierda.
xantomas_tuberosos_extensores_pie_derecho	int	Xantomas tuberosos extensores pie derecho.
xantomas_tuberosos_extensores_pies_izquierdo	int	Xantomas tuberosos extensores pies izquierdo.
xantelasma_parpado_superior_derecho	int	Xantelasma parpado superior derecho.
xantelasma_parpado_superior_izquierdo	int	Xantelasma parpado superior izquierdo.
xantelasma_parpado_inferior_derecho	int	Xantelasma parpado inferior derecho.

xantelasma_parpado_inferior_izquierdo	int	Xantelasma parpado inferior izquierdo.
xantomas_pliegues_palmares	int	Xantomas pliegues palmares.
arco_lipoideo_corneal	varchar	Arco lipoideo corneal.
lipemia_retiniana	int	Lipemia retiniana.
hepatomegalia	int	Hepatomegalia.
esplenomegalia	int	Esplenomegalia.
id_consulta	int	Identificador de la tabla "consulta" (Llave foránea).

Anexo 5 “Diagramas de componentes”.

Diagrama de componentes del caso de uso: “Gestionar documentos”.

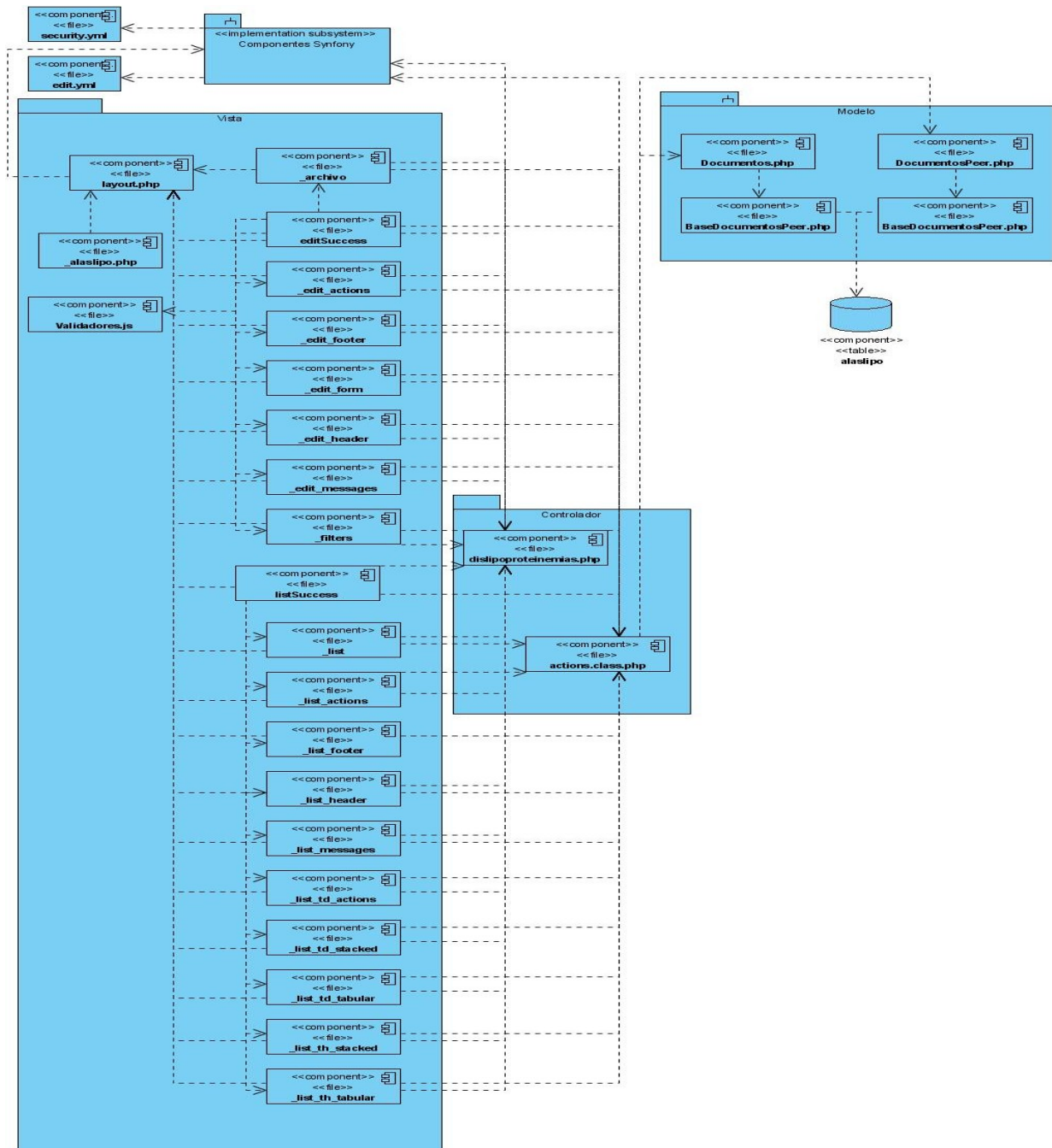


Diagrama de componentes del caso de uso: “Gestionar medicamentos”.

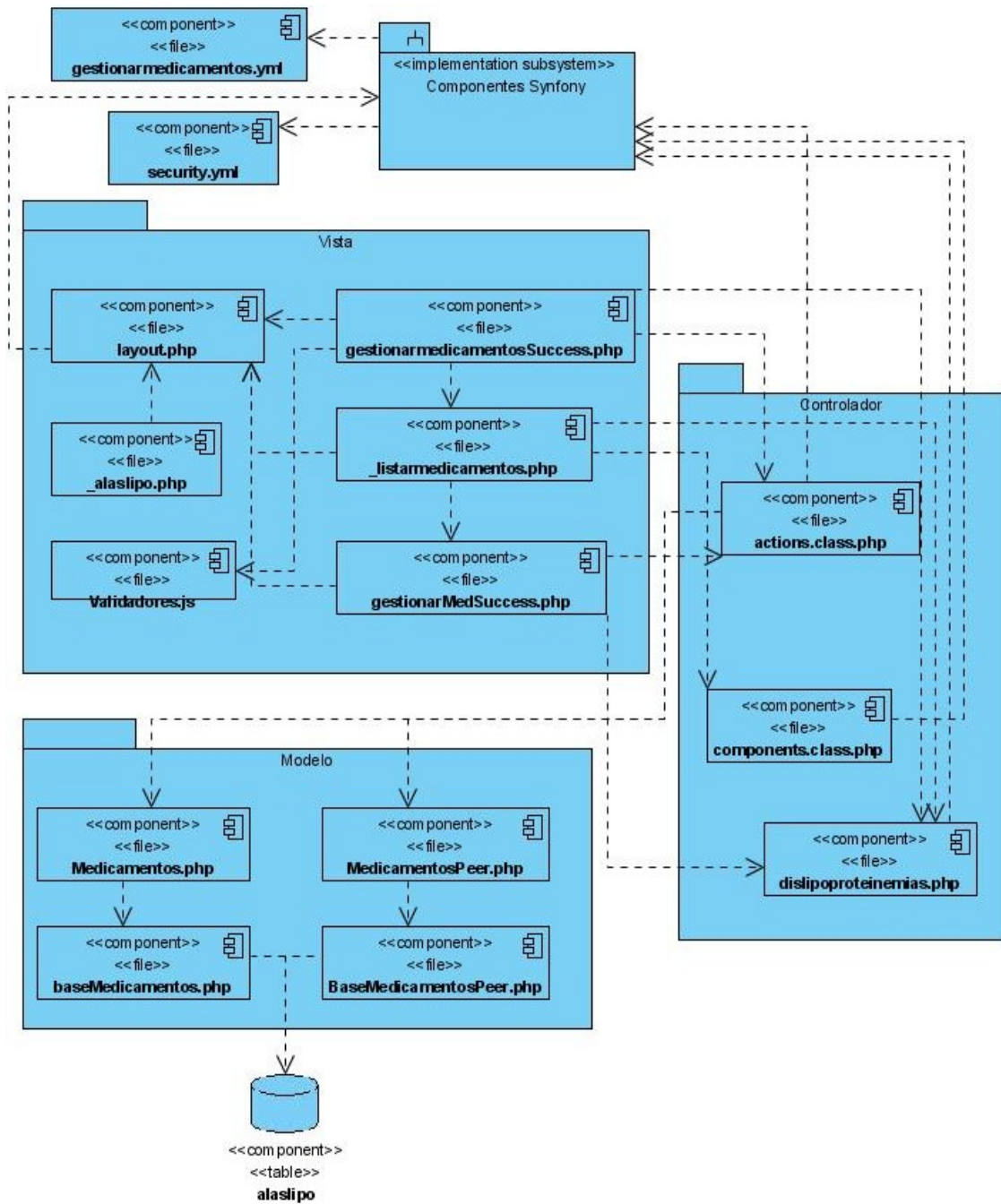


Diagrama de componentes del caso de uso: "Realizar Cálculos".

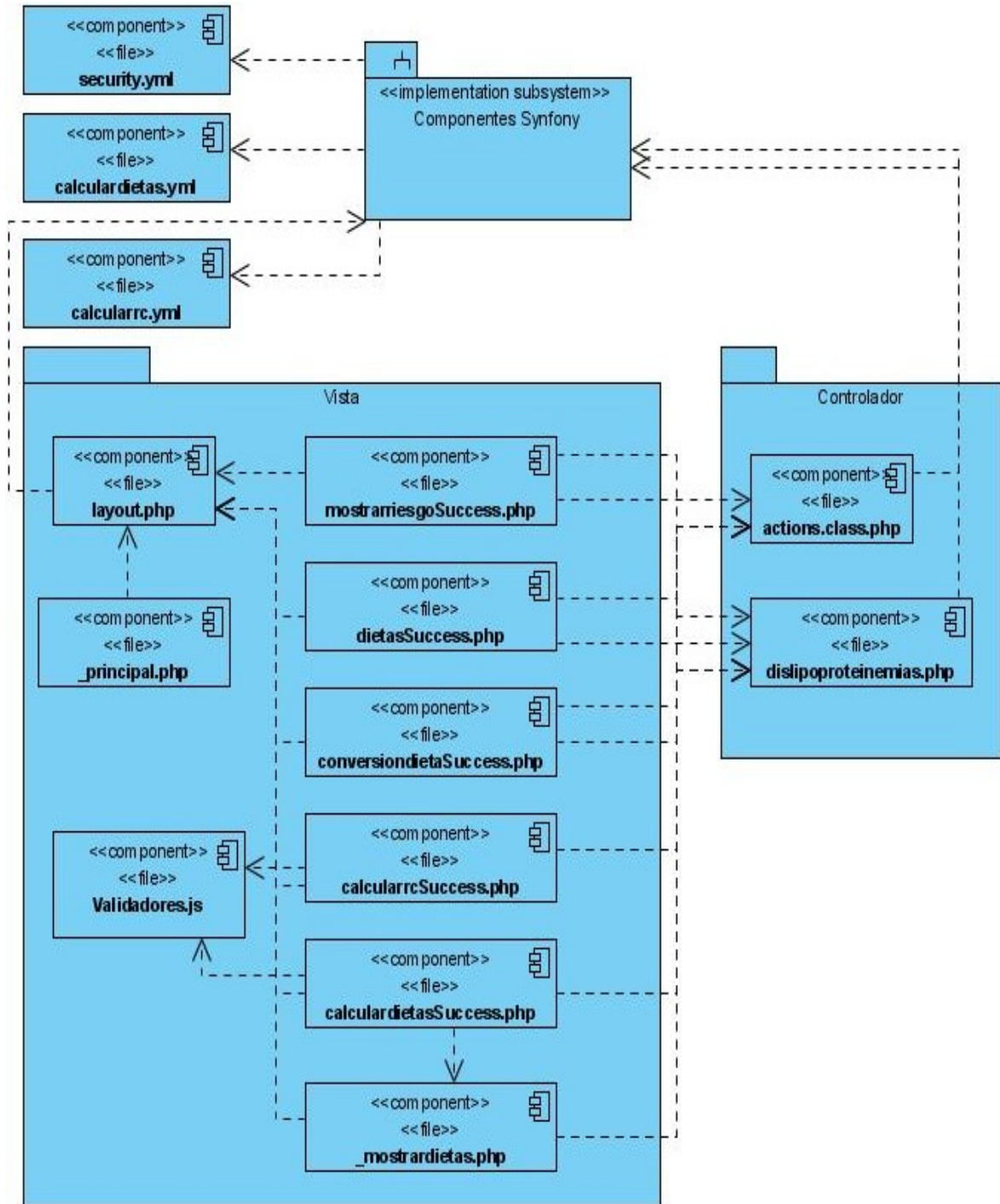


Diagrama de componentes del caso de uso: "Realizar consulta evolutiva".

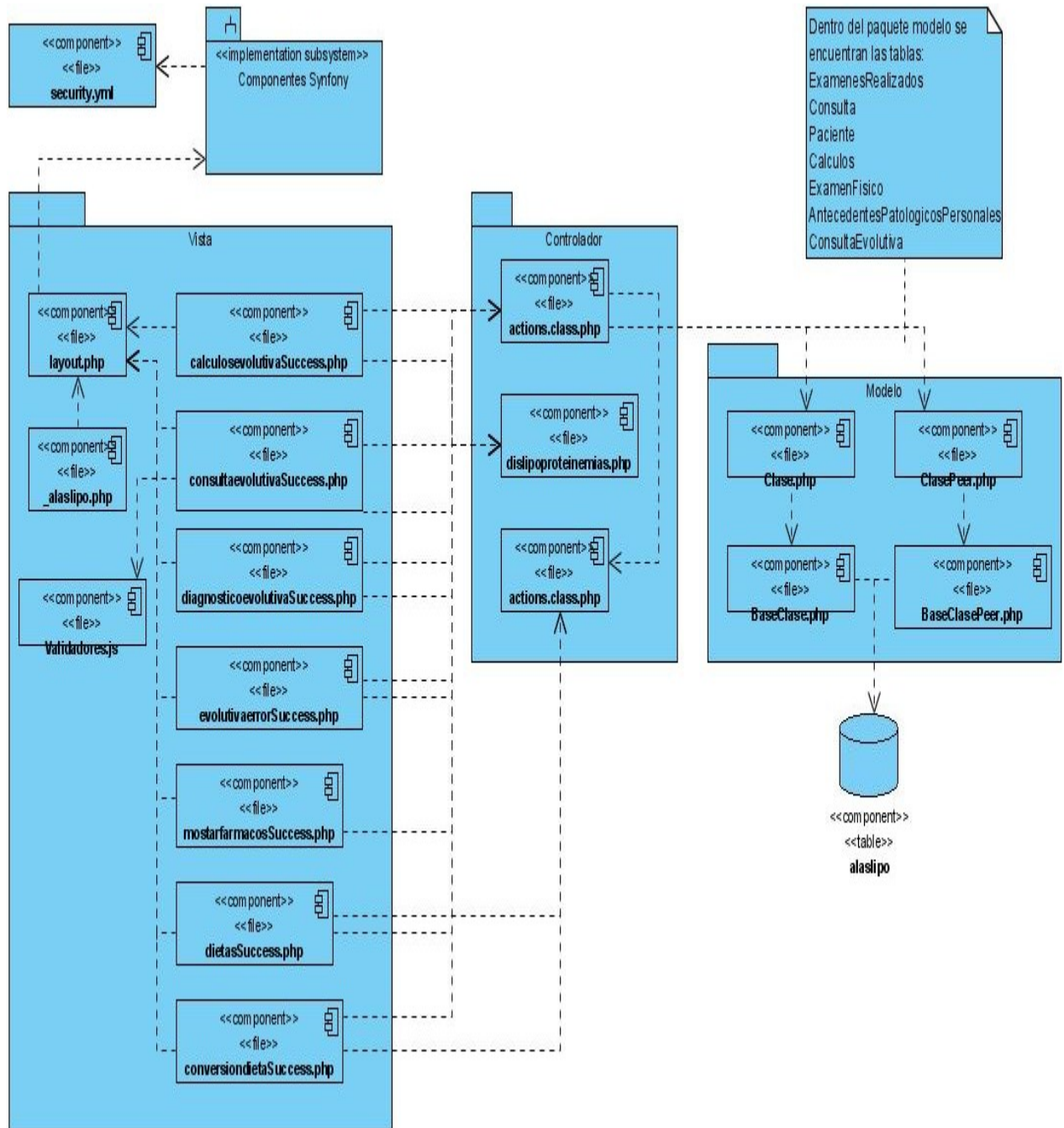


Diagrama de componentes del caso de uso: “Registrar Usuario”.

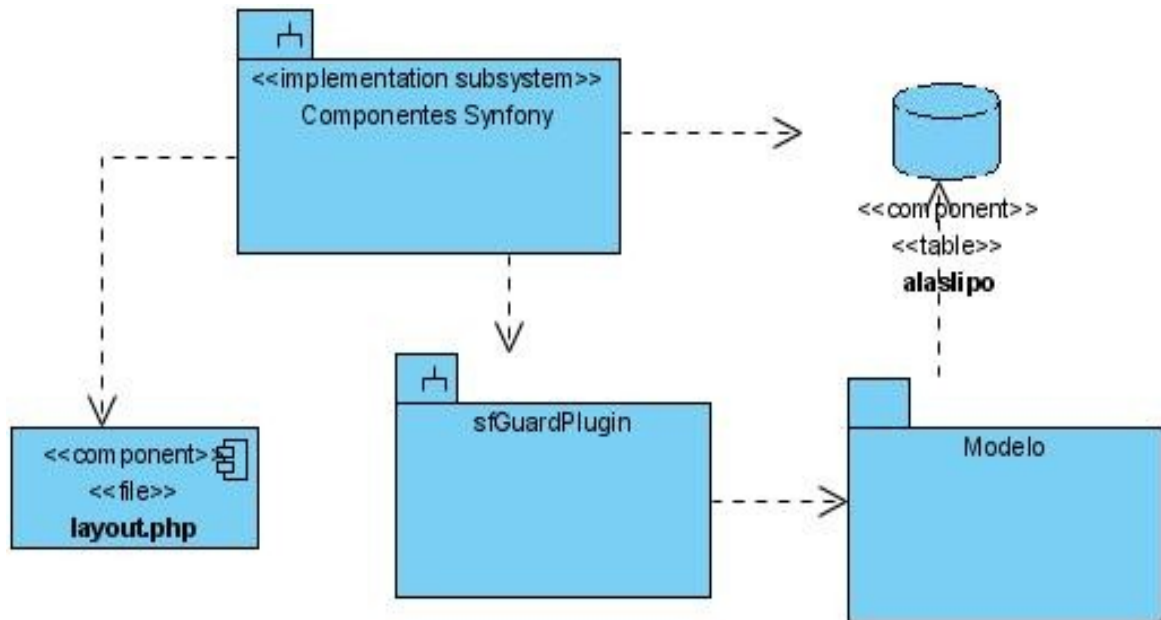


Diagrama de componentes del caso de uso: “Administrar Usuario”.

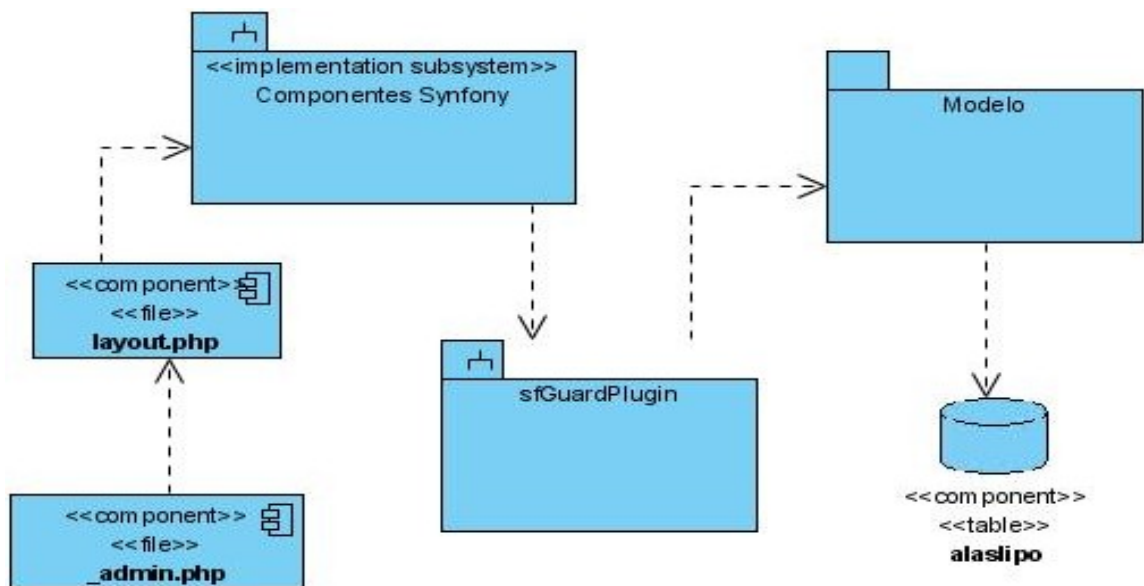


Diagrama de componentes del caso de uso: "Buscar documentos".

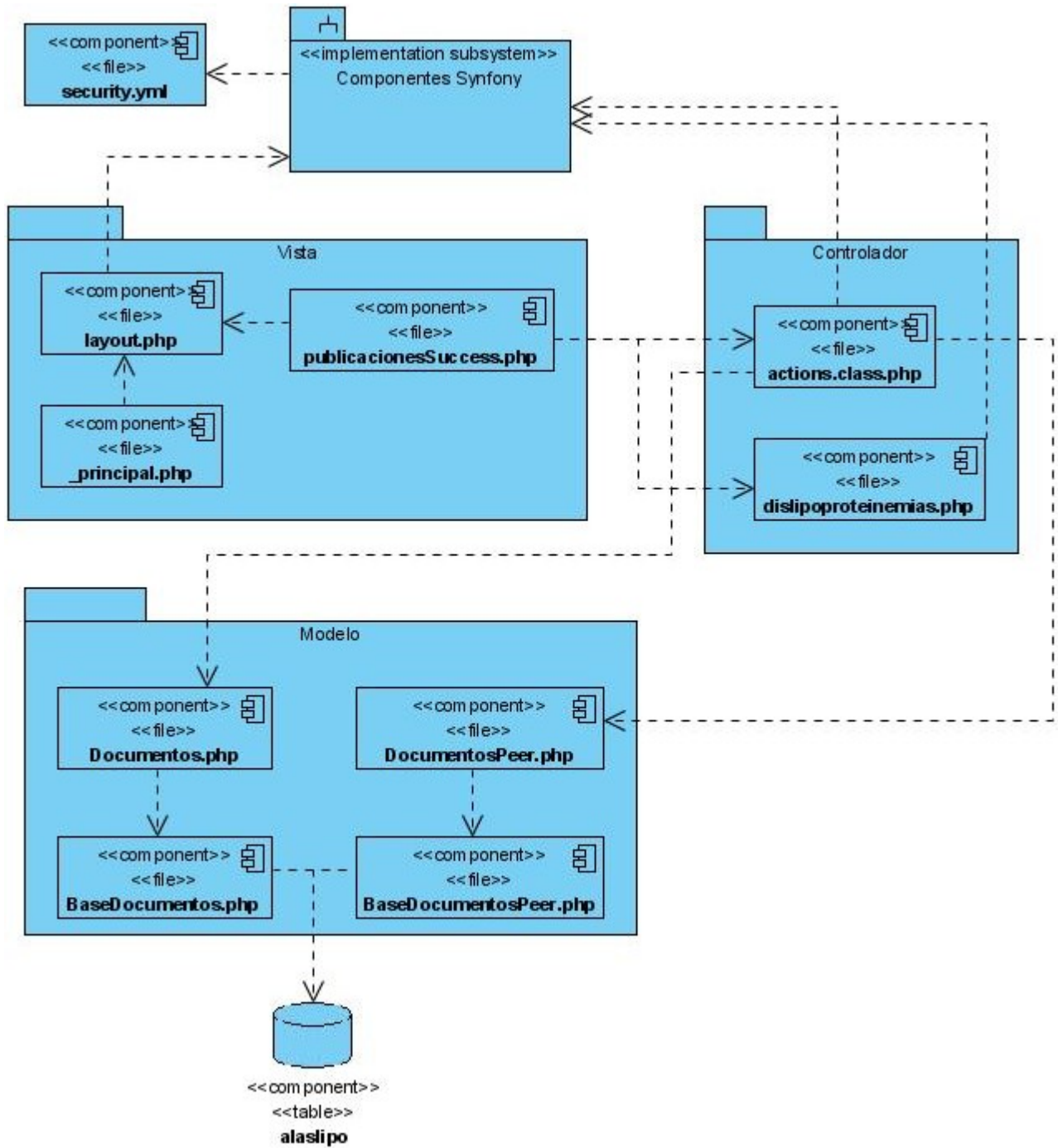


Diagrama de componentes del caso de uso: "Completar datos del paciente".

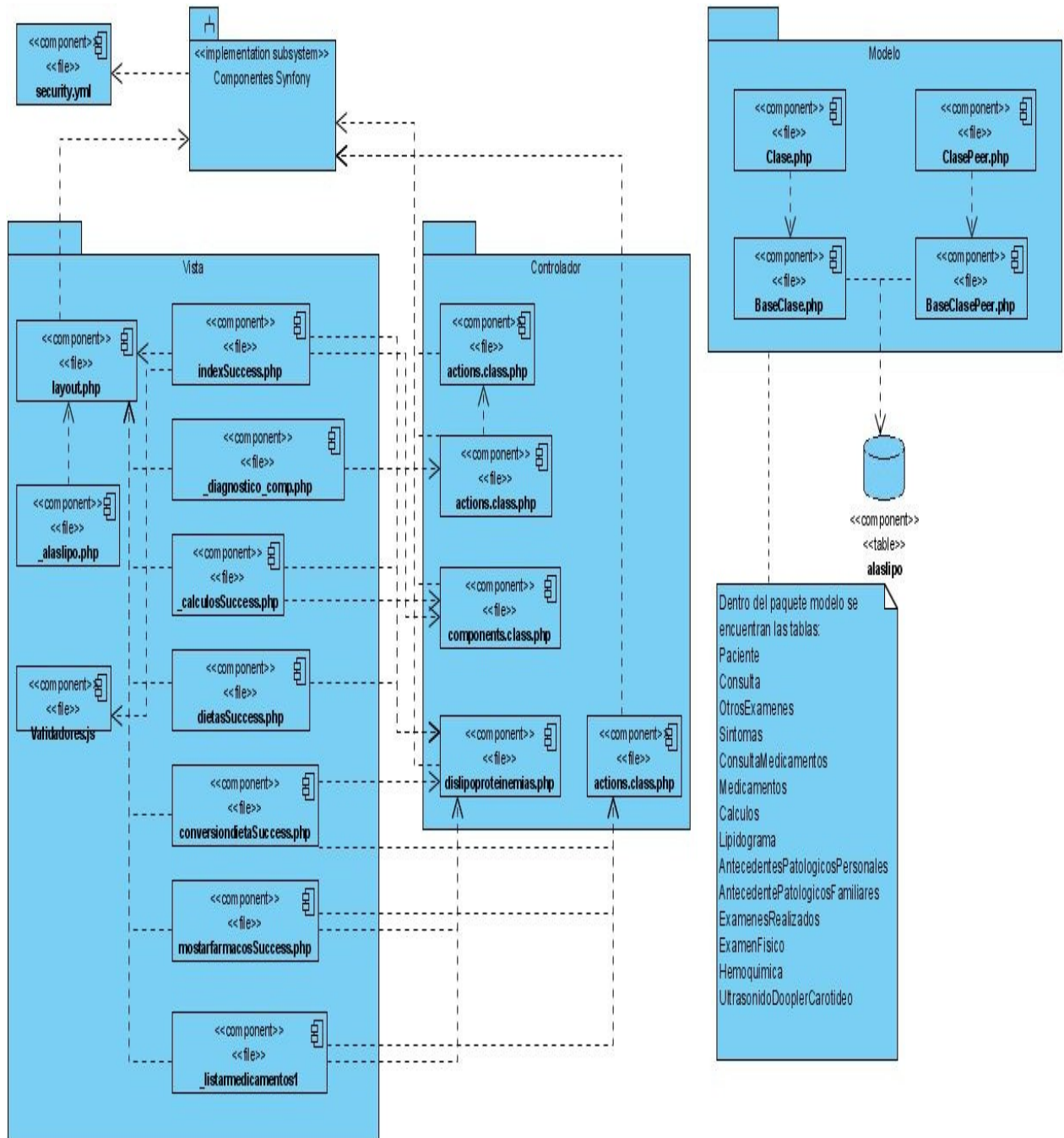


Diagrama de componentes del caso de uso: "Solicitar traslado paciente".

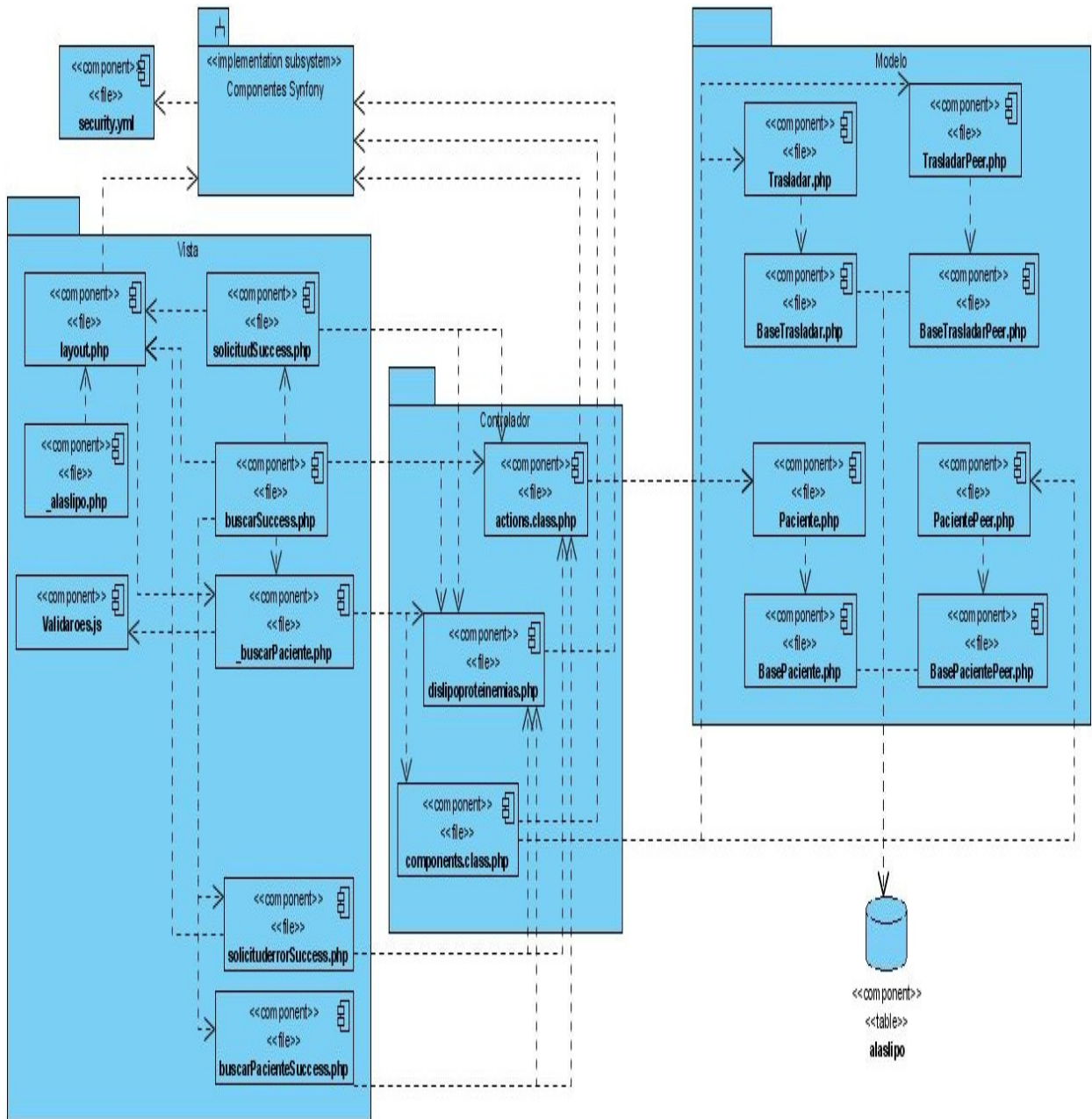


Diagrama de componentes del caso de uso: "Trasladar paciente".

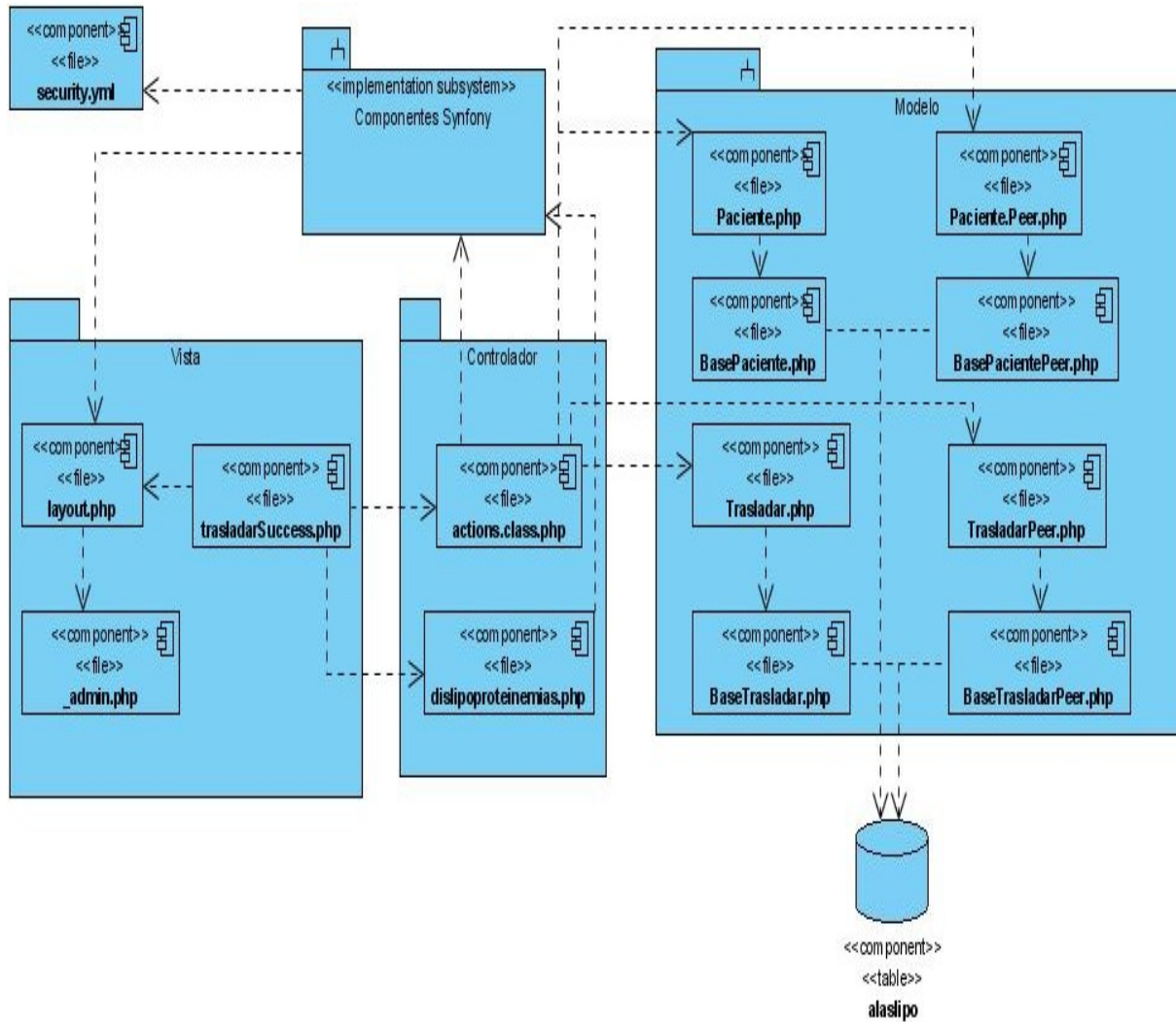
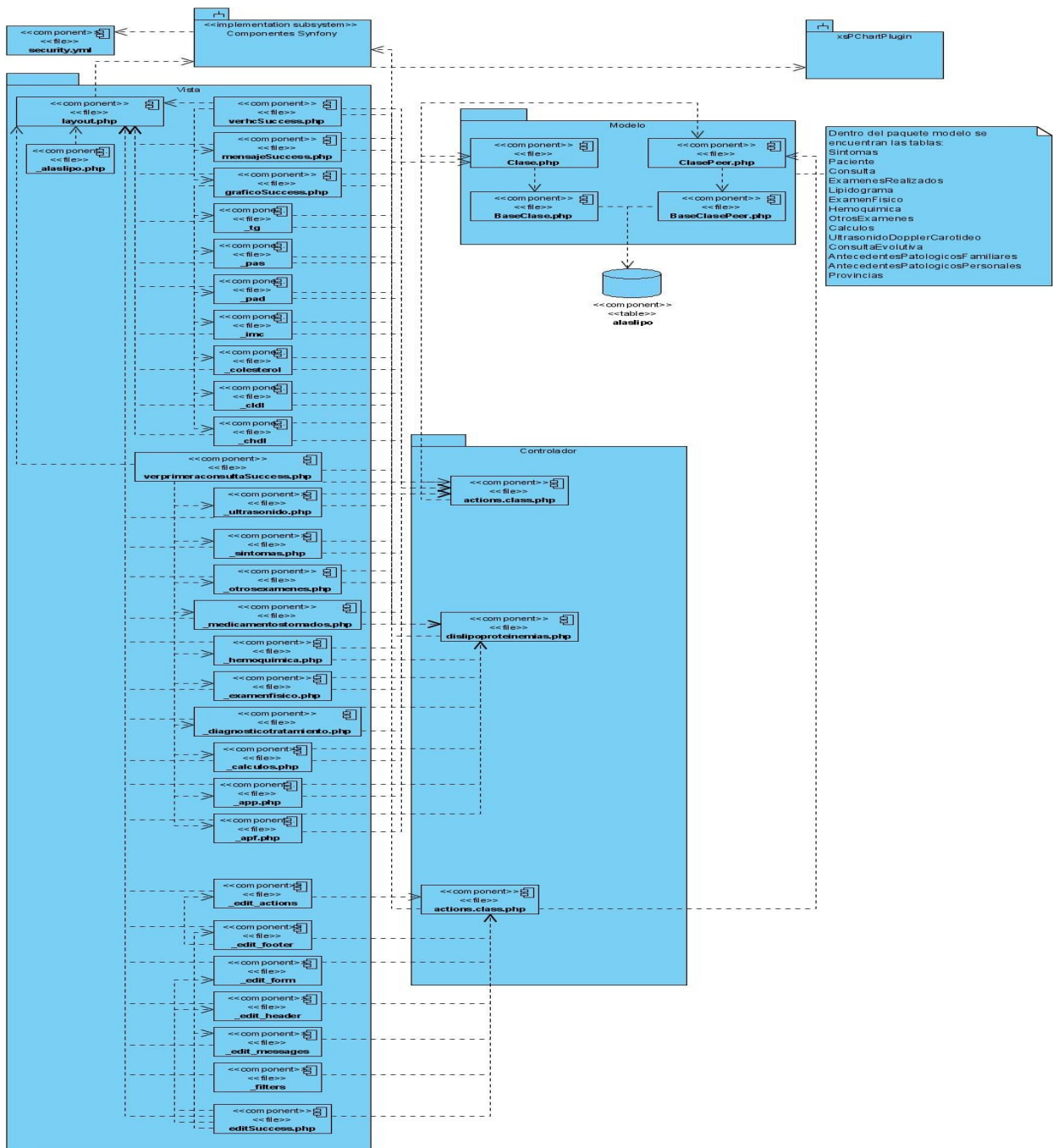


Diagrama de componentes del caso de uso: "Visualizar historia clínica".



Glosario de términos.

Antecedentes patológicos familiares: Lista de enfermedades padecidas por los padres o hermanos del paciente durante un período o toda su vida.

Antecedentes patológicos personales: Lista de enfermedades padecidas por el paciente durante un período o toda su vida.

Dislipoproteinemias: Se agrupan dentro de este término las enfermedades producidas por trastornos del metabolismo de las lipoproteínas. Pueden expresarse por aumento, disminución o presencia de las lipoproteínas anormales. Pueden ser secundarias a algunas enfermedades como la diabetes, obesidad, alcoholismo, hipotiroidismo, enfermedad renal crónica y otras o pueden producirse como consecuencia de un trastorno genético.

Diagnóstico: Identificación de la enfermedad, afección o lesión que sufre un paciente, de su localización y su naturaleza, llegando a la identificación por los diversos síntomas y signos presentes en el enfermo, siguiendo un razonamiento analógico.

Índice de masa corporal (IMC): Es una medida de asociación entre el peso y la talla del individuo, que se utiliza universalmente para estimar el grado de obesidad. Se conoce como índice de Quetelet, en honor al estadístico belga L. A. J. Quetelet, quien lo propuso. Se calcula dividiendo el peso en kilogramos por la talla en metros al cuadrado.

INFOMED: Red de Salud de Cuba.

Lípidos: Son un conjunto de moléculas orgánicas, la mayoría biomoléculas, compuestas principalmente por carbono e hidrógeno y en menor medida oxígeno, aunque también pueden contener fósforo, azufre y nitrógeno, que tienen como característica principal el ser hidrofóbicas o insolubles en agua y sí en disolventes orgánicos como la bencina, el alcohol, el benceno y el cloroformo.

Lipoproteínas: Son complejos de macromoleculares esféricos formados por un núcleo que contiene lípidos apolares (colesterol esterificado y triglicéridos) y una capa externa polar formada por

fosfolípidos, colesterol libre y apolipoproteínas. Su función principal es el transporte de triglicéridos, colesterol y otros lípidos al hígado, tejido adiposo, músculo y otros tejidos.

Morbimortalidad: Mortalidad por causa de una enfermedad.

SAAA: Servicio que brinda el sistema de salud basado en el modelo de Autenticación, Autorización y Auditoria (Authentication, Authorization and Accounting, AAA).

SOFTEL: Empresa que ofrece soluciones informáticas para el Sistema de Salud.