

Universidad de las Ciencias Informáticas

Facultad 6



Título: “Sistema Informático para la Red Nacional de Genética Médica (SIGM): Registro Cubano de Retraso Mental versión 2.0”

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

Autores: Yosúan Crespo García
Otto Leosdán Sterling Velázquez

Tutor: Ing. Yadira Barroso Rodríguez

Consultante: Dr. Roberto Lardoeyt Ferrer

Ciudad de la Habana, Cuba

Junio 2008

“Año 50 de la Revolución”

“Nunca consideres el estudio como un deber, sino como una oportunidad para penetrar en el maravilloso mundo del saber.”

Albert Einstein

DECLARACIÓN DE AUTORÍA

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

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

Yosuán Crespo García

Otto Leosdán Sterling Velázquez

Ing. Yadira Barroso Rodríguez

Firma del autor

Firma del autor

Firma del tutor

DATOS DE CONTACTO

Tutor:

Ing. Yadira Barroso Rodríguez

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: ybarroso@uci.cu

Consultante:

Dr. Roberto Lardoezt Ferrer

Centro Nacional de Genética Médica, Habana, Cuba

Email: lardgen@infomed.sld.cu

AGRADECIMIENTOS

Agradecemos a los que de una forma u otra ayudaron a nuestra formación, a nuestros familiares por su apoyo y confianza. Al Dr. Roberto y los profesores Alfonso, Yusdenis, Yadira y Dayana por brindarnos su ayuda. A nuestros compañeros de producción Reinier y Leinys, por ser pilares fundamentales en la realización de éste trabajo. Agradecer además a Joan, Franklís, Marta, Odannis, Jommy, Aleli, Angel, Leydi, Elvismary, Eylen, Asdrubal, Polivio, Aparicio, Omar y Sandra por su apoyo en momentos oportunos.

A nuestros amigos de manera general por su preocupación y por brindarnos todo su apoyo incondicional aunque no se encontraran cerca.

Agradecimiento inmenso a nuestro Comandante Fidel Castro Ruz y a la Revolución Cubana por darnos la oportunidad de estudiar en esta maravillosa Universidad y de ser parte de un gran sueño realizado.

DEDICATORIA

A mi mamá donde quiera que esté por haber confiado en mí y haberme brindado su entrega, perseverancia, dedicación y amor durante toda una vida. A ti te debo todo lo que soy.

A mi abuela Susana, a quien quiero con el alma, por haber sido siempre otra madre para mí y por guiar mis pasos por el camino correcto en todo momento.

A mi papá por haberme apoyado cuando lo he necesitado y cuando no también.

A mi abuelo Arístides por sus consejos y por haber sido un ejemplo a seguir.

A mi compañera y amiga Marta por su apoyo y comprensión y por estar junto a mí en los momentos difíciles.

A mis familiares y amigos por estar junto a mí en los buenos y malos momentos.

Yosuán Crespo García

A mi madre, por tener la fuerza y la dedicación para guiarme. Por brindarme todo su amor, enseñanza moral e intelectual. Por ser un ejemplo a seguir por su perseverancia. Lo que he logrado ha sido por ti y para ti.

A mi padre, amigo incondicional, por no perder en ningún momento su confianza en mí, por sus consejos acertados, por el apoyo y preocupación que no faltaron en momentos cuando más lo necesitaba.

A mi abuela querida Vidalina, por brindarme todo su cariño y optimismo.

A mi novia Mabel, por su apoyo y ayuda en los buenos y malos momentos y por guiar mis pasos por el camino correcto siempre. Gracias mi amor.

Otto Leosdán Sterling Velázquez

RESUMEN

En Cuba el Centro Nacional de Genética Médica (CNGM) es el organismo encargado de realizar estudios para mejorar los niveles de salud del pueblo y disminuir el impacto de las enfermedades con implicación genética como es el caso del retraso mental.

En la actualidad no se cuenta con un sistema informático que se encuentre integrado a la Red Nacional de Salud que sea accesible desde cualquier provincia del país, por lo que no existe la coordinación nacional de la actividad de asistencia médica en la red de centros y servicios de genética médica para llevar a cabo eficazmente el programa de diagnóstico y prevención del retraso mental en el país.

La presente investigación es el primer intento de integrar un sistema que permita gestionar las informaciones de las personas que presentan algún tipo de retraso mental a la Red Nacional de Salud. El Centro Nacional de Genética Médica conjuntamente con la Universidad de las Ciencias Informáticas decidió que se desarrollase el Sistema Informático de Genética Médica (SIGM) el cual incluye varios registros entre los que se encuentra, el Registro Cubano de Retraso Mental versión 2.0, siendo el diseño e implementación de dicho registro el tema a desarrollar en la presente investigación. Dicho registro aportará eficiencia y rapidez a la gestión de las informaciones de los pacientes que presentan retraso mental por parte de los genetistas cubanos.

Palabras claves:

Retraso mental, genética

TABLA DE CONTENIDOS

AGRADECIMIENTOS I

DEDICATORIA II

RESUMEN III

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 4

1.1 Antecedentes y sistemas existentes para la gestión de la información de retraso mental. 4

1.2 Roles y artefactos 6

 1.2.1 Rol Diseñador..... 6

 1.2.2 Rol Implementador 7

1.3 Herramientas y tecnologías..... 7

 1.3.1 Tecnologías del lado del cliente 7

 1.3.2 Tecnologías del lado del servidor 8

 1.3.3 Sistema de control de versiones..... 9

 1.3.4 Herramientas de desarrollo 10

 1.3.5 Framework para el desarrollo 11

 1.3.6 Sistema Gestor de Base de Datos..... 11

 1.3.7 Servidor Web Apache..... 12

 1.3.8 Herramienta CASE 13

 1.3.9 Metodología 13

1.4 Patrones de diseño y de arquitectura 16

 1.4.1 Patrones GRASP 16

 1.4.2 Patrones GoF 16

 1.4.3 Patrones de arquitectura 17

1.5 Conclusiones del capítulo 18

CAPÍTULO 2: DISEÑO DEL SISTEMA 19

2.1 Características del sistema 19

2.1.1	Requerimientos funcionales	19
2.1.2	Requerimientos no funcionales	19
2.1.3	Diagrama de casos de uso del sistema	21
2.1.4	Breve descripción de los casos de uso del sistema	23
2.1.5	Descripción detallada de los casos de uso del sistema	24
2.2	Pautas del Diseño	25
2.3	Patrones aplicados.....	26
2.3.1	Patrones GRASP aplicados.....	26
2.3.2	Patrones GoF aplicados	27
2.3.3	Patrón de arquitectura aplicado	27
2.4	Modelo de diseño.....	28
2.4.1	Descripción de las clases del diseño	29
2.4.2	Diagramas de clases web del diseño.	30
2.4.3	Diagramas de interacción.....	34
2.5	Modelo de despliegue	42
2.6	Tratamiento de errores.....	42
2.7	Seguridad	42
2.8	Conclusiones del capítulo.	43
CAPÍTULO 3: IMPLEMENTACIÓN DEL SISTEMA		44
3.1	Modelo de implementación.	44
3.1.1	Diagramas de componentes.....	44
3.2	Estilo de codificación.....	47
3.3	Código fuente de los principales métodos y su descripción.....	49
3.4	Interfaz del registro.	52
3.5	Validación a nivel de desarrollador.....	56
3.6	Conclusiones del capítulo.	58
CONCLUSIONES		59
RECOMENDACIONES		60
REFERENCIAS BIBLIOGRÁFICAS		61

BIBLIOGRAFÍA.....	63
ANEXOS.....	65
GLOSARIO DE TÉRMINOS.....	76

INTRODUCCIÓN

En la historia de la humanidad el retraso mental hace referencia a limitaciones sustanciales en el desenvolvimiento corriente de una persona, por lo que, es de vital importancia para el desarrollo de la población, que el sistema de salud pueda conocer las características de su comportamiento, su distribución geográfica, los factores etiológicos, y la tendencia de su incidencia, para sus posteriores investigaciones y soluciones a esta discapacidad.

La necesidad de conocer, cuantificar y calificar los casos registrados por padecer una determinada patología no es algo nuevo en el área de la medicina preventiva y la salud pública. Resulta evidente el papel primordial que han desempeñado los registros de enfermedades, como herramienta fundamental en la mencionada área. Los registros de casos han posibilitado el conocimiento de la extensión de la población afectada, contribuyendo a una mejor tipificación de sus necesidades en planificación sanitaria, medidas sociosanitarias, políticas de prevención, nuevos tratamientos y en definitiva, conocer la magnitud del problema. Una de las principales aplicaciones de los registros es la posibilidad de diseñar y realizar estudios de investigación epidemiológica, clínica y genética.

A nivel mundial se le ha prestado especial importancia al estudio de las causas que lo originan y no escasean las intenciones de hacerles la vida un poco más fácil a estas personas que presentan dicho defecto genético, por lo que existen revistas internacionales, programas de apoyo, libros y hasta se han creado líneas de investigación enmarcadas en este tema.

España es uno de los países que más se interesa por este tipo de discapacidad. Se han creado diferentes grupos encargados de realizar estudios referentes al Retraso Mental de Origen Genético (RMOG) como es el caso de la Fundación de Investigación del Hospital de la Fe.

En Cuba el Centro Nacional de Genética Médica es el organismo encargado de realizar estudios para mejorar los niveles de salud del pueblo y disminuir el impacto de las enfermedades con implicación genética como es el caso del retraso mental.

En el año 2007, se realizó una versión inicial del Registro Cubano de Retraso Mental (RECURM), la cual permitió que la gestión de las informaciones de los pacientes que presentan esta discapacidad se

hiciera de forma dinámica utilizando herramientas informáticas evitándose así la posibilidad de pérdida de información que existía anteriormente. (2)

En la actualidad no se cuenta con un sistema informático que se encuentre integrado a la Red Nacional de Salud que sea accesible desde cualquier provincia del país, por lo que no existe la coordinación nacional de la actividad de asistencia médica en la red de centros y servicios de genética médica para llevar a cabo eficazmente el programa de diagnóstico y prevención del retraso mental en el país.

Se identificó el siguiente **problema**: ¿Cómo obtener un producto funcional a partir de los requerimientos identificados para el Registro Cubano de Retraso Mental versión 2.0 integrado a la Red Nacional de Salud?

Este problema se enmarca en el **objeto de estudio**: Proceso de gestión de información de la salud.

El **campo de acción** abarcado es: Proceso de desarrollo del software para la gestión de la información de retraso mental del Sistema Informático para la Red Nacional de Genética Médica.

Objetivo general de la investigación: Desarrollar el diseño e implementación del Registro Cubano de Retraso Mental versión 2.0 del Sistema Informático para la Red Nacional de Genética Médica.

Objetivos específicos de la investigación:

- Diseñar el Registro Cubano de Retraso Mental versión 2.0.
- Implementar el Registro Cubano de Retraso Mental versión 2.0.

Tareas a desarrollar para cumplir los objetivos:

- Investigación sobre sistemas existentes.
- Análisis de la versión existente de RECURM.
- Estudio de la Arquitectura y las tecnologías a utilizar.
- Diseño de las clases del Registro Cubano de Retraso Mental versión 2.0.
- Elaboración del Modelo de Diseño del Registro Cubano de Retraso Mental versión 2.0.
- Implementación de las secciones insertar, modificar y generar reportes del Registro Cubano de Retraso Mental versión 2.0.

- Validación del registro a nivel de desarrollador.

Capítulo 1. Fundamentación teórica: Incluye el estado del arte del tema tratado en la investigación a nivel internacional y nacional, las tecnologías, metodologías y software usados en la actualidad para darle solución al problema. Se exponen los roles que se desempeñan en el proceso de desarrollo así como se definen los patrones de diseño y de arquitectura que se utilizan.

Capítulo 2. Diseño del sistema: Contiene los requerimientos funcionales y no funcionales del registro que darán solución al problema planteado. Además se identifican los casos de uso y los actores del sistema y se brinda una descripción textual de los casos de uso. Se definen las clases del diseño con las que va a contar el registro. Se exponen los diagramas de clases del diseño y los diagramas de interacción realizados en el diseño. Se comentan los patrones de diseño y de arquitectura aplicados. Además se trata el tratamiento de errores y la seguridad del registro.

Capítulo 3. Implementación del sistema: Contiene la representación de los diagramas de componentes del registro. Además se brinda una descripción de los principales métodos implementados así como se muestran imágenes de la interfaz del registro.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se analizan las principales herramientas y tecnologías que se utilizarán para el desarrollo del registro. Se realiza un estudio de las tendencias y tecnologías actuales existentes para la realización del registro, definiéndose los más adecuados, además se brinda un enfoque general de los sistemas informáticos existentes semejantes al de esta investigación. Igualmente se expone el desempeño de los roles envueltos en el proceso de desarrollo y se definen los patrones de diseño y de arquitectura que se utilizan.

1.1 Antecedentes y sistemas existentes para la gestión de la información de retraso mental.

Desde el punto de vista estadístico, gran parte de las dificultades con que cuenta la investigación en cuanto al Retraso Mental de Origen Genético (RMOG) se debe a que existe el problema de dispersión originado por la poca frecuencia de la enfermedad; hay pocos casos, muy alejados unos de otros, en diferentes localidades y poblaciones de difícil acceso. A esta dispersión de los casos de la enfermedad hay que añadir la dispersión de la información de cada uno de los casos, debida a la gran cantidad de especialistas y profesionales de la salud que entran en contacto con el paciente en la misma o diferentes instituciones: pediatras, psicólogos, psiquiatras, neurólogos, genetistas clínicos y de laboratorio, médicos de cabecera, etc. Un mismo caso puede llegar a pasar por varios centros o instituciones, donde se diagnostican y se tratan de formas diversas. Por lo tanto, la dispersión es el motivo principal que justifica la necesidad de gestionar las informaciones de los casos.

Uno de los países más destacados en el tema es España, el cual ha realizado varios estudios con el objetivo de determinar los factores causales del retraso mental de origen genético. Tal es el caso de la Fundación de Investigación del Hospital de la Fe, que está llevando a cabo un proyecto de investigación para detectar reordenamientos genómicos responsables del retraso mental con una novedosa técnica de biochips. Esta investigación tiene como objetivo final averiguar la causa de este tipo de enfermedades raras para asesorar genéticamente a las familias y con el tiempo, facilitar un posible pronóstico y tratamiento de este tipo de discapacidad. (1)

En el año 2003 surge en el estado español GIRMOGEN, el Grupo de Investigación en Retraso Mental de Origen Genético, que desde su constitución, ha perseguido la creación del primer registro de pacientes diagnosticados de Retraso Mental de Origen Genético (RMOG) del estado español, con un enfoque básicamente genético y clínico. (3)

El principal objetivo de GIRMOGEN es consolidar una estructura de grupos cooperativos interdisciplinarios para la investigación del Retraso Mental de Origen Genético en toda España para

integrar la investigación clínica, molecular y epidemiológica. En septiembre de 2007 comenzaron la realización de un registro de familias en España y de un censo del número de sus familiares, portadores y sanos. El registro de los casos se fundamenta en una base de datos actualizable en línea a través de Internet. (4)

Durante los últimos diez años, el interés de los investigadores europeos por los temas relacionados con la salud mental en la población con retraso mental ha aumentado espectacularmente. A la vez, se ha evidenciado la existencia de discrepancias a la hora de abordar las necesidades de salud mental de este grupo de población. Uno de los propósitos del proyecto europeo Standardising criteria for mental health assessment in adults with mental retardation: an European perspective, financiado por la Iniciativa Biomed-2, en el que participan investigadores de Gran Bretaña, Austria, Grecia, Irlanda y España, es alcanzar un consenso y elaborar unas directrices de “buenas prácticas” en relación a la atención psiquiátrica de las personas con retraso mental. (5)

Cuba no se ha quedado atrás y entre sus principales retos se encuentra la atención a personas que presentan algún tipo de retraso mental, por lo que uno de los más ambiciosos proyectos científicos y sociales del que se ha tenido noticia lo constituye, el censo de las personas que presentan algún tipo de retraso mental en la isla. En el 2003 en el país se realizó un estudio a nivel nacional del universo de personas que padecen de discapacidad intelectual. Cada una de ellas fue visitada en su propio domicilio por especialistas de alto nivel, hecho sin precedentes en el mundo.

Para la puesta en práctica del proyecto, se le encomendó la tarea al Centro Nacional de Genética Médica (CNGM), institución científica que tiene como principal objetivo llevar a cabo acciones asistenciales, docentes y de investigación en el campo de los problemas de salud de carácter genético, encaminados a elevar la calidad de vida y el bienestar del pueblo. Dentro de sus metas se propone lograr la condición de centro de excelencia por el desarrollo de valores humanos, científicos y patrióticos de sus integrantes, al servicio del pueblo y el reconocimiento nacional e internacional que derive de sus contribuciones. (6)

En el 2007 se realizó en la facultad 6 de la Universidad de la Ciencias Informáticas un Sistema Automatizado de Registro Cubano de Retraso Mental el cual aportó a los especialistas del CNGM la posibilidad de gestionar la información de las personas que presentan retraso mental en Cuba de una forma rápida y eficiente. Dicho sistema fue probado en la provincia Villa Clara pero no se puso en práctica en todo el país debido a no estar integrado a la Red de Salud de Cuba (Infomed).

La presente investigación es el primer intento de integrar un sistema que permita gestionar las informaciones de las personas que presentan algún tipo de retraso mental a la Red Nacional de Salud. Uno de los principales objetivos del Centro Nacional de Genética Médica es contar con una

herramienta sencilla y fácil de manejar por parte de los genetistas municipales y provinciales que sea capaz de gestionar de forma rápida y eficiente la información de las personas que presenten algún tipo de retraso mental desde cualquier parte del país y de esta forma tomar medidas para elevar la calidad de vida y el bienestar de las personas que presenten este padecimiento genético.

El Centro Nacional de Genética Médica conjuntamente con la Universidad de las Ciencias Informáticas decidió que se desarrollase el Sistema Informático de Genética Médica (SIGM) el cual incluye varios registros, entre los que se encuentran, el Registro Cubano de Enfermedades Genéticas (RECUEGEN), el Registro Cubano de Malformaciones Congénitas (RECUMAC), el Registro Cubano de Discapacitados (RECUDIS), el Registro Cubano de Gemelos (RECUGEM), el Registro Cubano de Historias Clínicas (RECUHCL), el Módulo Teleconsulta y el Registro Cubano de Retraso Mental (RECURM) versión 2.0, siendo este último el tema a desarrollar en la presente investigación. Todos los registros poseen una base de datos común y se rigen por la misma arquitectura.

1.2 Roles y artefactos

A continuación se exponen los roles y los artefactos que se generan en la presente investigación según RUP.

1.2.1 Rol Diseñador

En la metodología RUP el diseñador es el responsable de diseñar una parte del sistema cumpliendo con las restricciones de los requerimientos, arquitectura y proceso de desarrollo del proyecto, identifica y define las responsabilidades, operaciones, atributos y relaciones de los elementos de diseño.

Los artefactos que realiza el diseñador que se obtendrán en la presente investigación son:

Realización de casos de uso del diseño: Es una colaboración en el modelo de diseño que describe como se realiza un caso de uso específico, y como se ejecuta en términos de casos de uso del diseño. Una realización de caso de uso del diseño proporciona una traza directa a una realización de caso de uso del análisis en el modelo de análisis.

- **Clases del diseño:** Una clase es una descripción de un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos, y la semántica.

Diagrama de clases: Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases también son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Los diagramas de clases son importantes

no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Paquetes de diseño: Son una colección de clases, relaciones, realizaciones de casos de uso, diagramas y otros paquetes que están de alguna forma relacionados. Es usado para estructurar el modelo de diseño dividiéndolo en partes más pequeñas.

Subsistema de diseño: Es una parte del sistema que encapsula el comportamiento, incluye interfaces y paquetes. (18)

1.2.2 Rol Implementador

El rol Implementador es responsable de desarrollar y de probar componentes de acuerdo con los estándares adoptados del proyecto para la integración en subsistemas más grandes.

El artefacto que realiza el implementador que se obtendrá en la presente investigación es:

Elementos de implementación: los elementos de implementación son la parte física de la implementación, incluyen los archivos y directorios. Incluyen ficheros de código (fuentes, binarios o ejecutables), ficheros de datos y de documentación como ficheros de ayuda online. (18)

1.3 Herramientas y tecnologías

Para el desarrollo del registro se realizó un estudio sobre las posibles herramientas y tecnologías a utilizar en su construcción teniendo en cuenta las herramientas establecidas para el desarrollo de software de Infomed y las tecnologías soportadas por dicha institución.

1.3.1 Tecnologías del lado del cliente

Los lenguajes del lado del cliente (entre los cuales no sólo se encuentra el HTML sino también el Javascript el cual es simplemente incluido en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. El código Javascript es embebido directamente en el código HTML, haciendo fácil la creación de páginas Web con contenido dinámico. Está diseñado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador Web y es soportado por la gran mayoría de los navegadores lo que lo posiona en el lugar del lenguaje de programación web del lado del cliente más utilizado.

Con Javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. Javascript es el siguiente paso, después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para trabajar con rapidez, a veces con ligereza. Entre las acciones típicas que se pueden realizar en Javascript se tienen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, Javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, con las que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. Es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc.

(2)

1.3.2 Tecnologías del lado del servidor

Los lenguajes de lado servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Entre los lenguajes que trabajan del lado del servidor se pueden citar algunos, que se destacan por ser los más utilizados por los programadores como son PERL, ASP, PHP, Java, JSP, entre otros. Estos lenguajes desarrollan la lógica de negocio dentro del servidor, además se encargan de los accesos a los distintos Sistemas Gestores de Bases de Datos.

En el presente trabajo se utilizará el Personal Home Page (PHP) 5.2.0 como lenguaje de programación. El mismo es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor, el cual brinda las siguientes ventajas:

- PHP corre en casi cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al sistema operativo.
- La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C

se incluye al Java y Javascript, de hecho mucha de la funcionalidad del PHP se la debe al C por lo que le es familiar a muchos programadores.

- PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código.
- Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como módulo CGI.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre se podrá disponer de ODBC para situaciones que lo requieran.
- Una gran variedad de módulos. Cuando un programador PHP necesite una interfaz para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.
- PHP es código abierto, lo cual significa que el usuario no depende de una compañía específica para arreglar lo que no funcione, no es necesario pagar actualizaciones anuales para tener una versión que funcione. (7)

1.3.3 Sistema de control de versiones

Se utilizará el sistema de control de versiones Subversion (SVN) 1.4. El sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que se cambie con frecuencia, como una novela, o el código fuente de un programa.

Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local lo que permite compartir los cambios sobre un mismo conjunto de ficheros. Además, el repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad.

Es necesario el uso de este sistema de control de versiones dada las ventajas que brinda el mismo:

- Actualización de ficheros modificados.
- Copias de seguridad centralizadas.
- Historial de cambios.
- Brinda acceso remoto.
- Provee seguridad al sistema. (8)

1.3.4 Herramientas de desarrollo

El Eclipse PDT 3.3.1.1 es el entorno de desarrollo integrado IDE (Integrated Development Environment) seleccionado para implementar el registro. También existe el Zend Studio pero no es gratuito y por eso no se utiliza.

El Eclipse es un IDE para todo tipo de aplicaciones, inicialmente desarrollado por IBM, y actualmente gestionado por la Fundación Eclipse.

La característica clave de Eclipse es la extensibilidad. Eclipse es una gran estructura formada por un núcleo y muchos plugins que van conformando la funcionalidad final. La forma en que los plugins interactúan es mediante interfaces o puntos de extensión; así, los nuevos aportes se integran sin dificultad ni conflictos.

Se seleccionó el Eclipse como entorno de desarrollo porque:

- Permite el desarrollo de aplicaciones utilizando Subversion como sistema de control de versiones.
- Presenta un potente editor de código.
- Brinda la facilidad de agregarles las librerías de los frameworks a utilizar y una vez agregadas realizar completamiento de código del framework. (8)

Además del Eclipse también se utiliza el Quanta que es una herramienta de desarrollo web para el entorno de escritorio KDE. Está diseñado para el desarrollo web y rápidamente se está convirtiendo en un editor maduro y con gran variedad de características, entre las que se encuentran:

- Usa KIO para FTP, SSH (con FISH) y soporta otros protocolos.
- Asistentes para creación de tablas, enlaces y páginas en blanco.
- Resaltado de sintaxis de HTML, Javascript, CSS y varios más.
- Contiene un analizador que informa acerca de la correcta creación de nuestras páginas.
- Soporta plugins como son: Konsole, KImageMapEditor, KLinkStatus, Cervisia (CVS) y KFileReplace.
- Previsualización: Los documentos pueden ser previsualizados dentro de la aplicación usando el motor KHTML. Es posible preprocesar los documentos a través de un servidor web antes de previsualizar (20).

1.3.5 Framework para el desarrollo

Un framework es un conjunto de herramientas que pueden ser reutilizadas, de tal forma que el desarrollo de grandes proyectos se realiza de una manera más simple y rápida. Para PHP existe un gran número de frameworks, dentro de los cuales se encuentran:

- 1) CodeIgniter
- 2) Cake
- 3) Kumbia
- 4) Zend
- 5) Prado
- 6) Symfony

El framework seleccionado es Symfony 1.0.16, un completo framework diseñado para optimizar el desarrollo de las aplicaciones web gracias a sus características. 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.

Razones por la que fue seleccionado:

- 1) Gran documentación en varios idiomas (Libro gratuito + Documentación de la API + Wiki).
- 2) Desarrollo constante.
- 3) Gran comunidad de usuarios
- 4) Flexibilidad (Diseño + configuración + plugins)
- 5) Reutilización de componentes probados en vez de “inventar de nuevo la rueda”. (8)

1.3.6 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Entre los SGBD comúnmente utilizados en el mundo se conocen Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Interbase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

El SGBD seleccionado es el MySQL 5. MySQL es un Sistema Gestor de Base de Datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Las principales características de este SGBD son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
 - Soporta gran cantidad de tipos de datos para las columnas.
 - Dispone de API en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
 - Gran portabilidad entre sistemas.
 - Soporta hasta 32 índices por tabla.
 - Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- (9)

1.3.7 Servidor Web Apache

El servidor web utilizado es el Apache 2.0. Hoy en día es el servidor web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Por supuesto, se distribuye prácticamente con todas las implementaciones de Linux.

Tiene capacidad para servir páginas tanto de contenido estático, para lo que serviría sencillamente un viejo ordenador 486, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información. Es un software de libre distribución que publica su código fuente, lo que permite que cualquiera pueda modificarlo y colaborar así a su desarrollo.

Apache está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. (2)

1.3.8 Herramienta CASE

Las herramientas Computer Aided Software Engineering (CASE), en español: Ingeniería de Software Asistida por Computadoras, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software. Algunas herramientas CASE conocidas son el ArgoUML, Rational Rose, Visual Paradigm, Easy CASE, Xcase, CASE Studio 2, CASEWise, entre otras.

La herramienta CASE seleccionada es Visual Paradigm Community Edition 6.0. Es una herramienta basada en el uso de UML (Unified Modeling Language) 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 costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Una de las grandes ventajas de Visual Paradigm es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

Esta herramienta tiene licencia libre para la comunidad de desarrolladores, lo cual constituye otra gran ventaja ya que se pueden obtener actualizaciones anuales sin tener que pagar a compañías propietarias. (8)

1.3.9 Metodología

Existen algunos aspectos que se deben tener en cuenta para desarrollar exitosamente un proyecto. Entre ellos está el hecho de que se debe tener definida claramente la metodología de trabajo de cada fase del proceso del desarrollo de software, ya que todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. En la actualidad existen diferentes metodologías que se utilizan para el desarrollo de software, como son RUP, XP y MSF todas ellas con diferentes peculiaridades. Para este proyecto, por las características con que cuenta, se decidió tomar la metodología Rational Unified Process (RUP).

La metodología RUP, se define como un conjunto de actividades, métodos y prácticas utilizadas por un grupo de humanos para desarrollar y mantener el software y sus productos asociados. RUP utiliza UML para definir los modelos de software y puede definirse como un modelo que es dirigido por los

casos de uso, centrado en la arquitectura, iterativo e incremental. Se divide en 4 fases: Inicio, Elaboración, Construcción y Transición.

- **Inicio:** en esta fase el objetivo es determinar la visión del proyecto.
- **Elaboración:** en esta fase el objetivo es determinar la arquitectura óptima.
- **Construcción:** en esta fase el objetivo es llegar a obtener la capacidad operacional inicial.
- **Transición:** en esta fase el objetivo es llegar a obtener el release del proyecto.

Cada una de estas fases es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala, dando así la posibilidad de que el usuario no tenga que esperar hasta el final para darse cuenta que la funcionalidad desarrollada no era la esperada.

El ciclo de vida que se produce por cada iteración, es desarrollado bajo dos disciplinas: la Disciplina de Desarrollo que es la que se encarga de la Ingeniería de Negocios, Requerimientos, Análisis y Diseño, Implementación, y Pruebas y la Disciplina de Soporte que se encarga de la Configuración y administración del cambio, Administrando el proyecto, Ambiente y Distribución. Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración. RUP tiene como elementos: Actividades, Trabajadores y Artefactos.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. 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 de apoyo.

Flujos de trabajo:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce el release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto, así como el procedimiento para implementar el proceso en una organización. (2)

La metodología RUP utiliza el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) el cual es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar los procesos de negocio y las funciones de sistema, además permite escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. Es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software). Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque orientado a objetos, utilizándose también en el diseño Web. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño orientado a objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios. Su principal característica es que se utiliza para la modelación de sistemas, con tecnología orientada a objetos, en donde el cliente participa en todas las fases del proyecto, con corrección de errores viables en todas las fases, aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

1.4 Patrones de diseño y de arquitectura

Un patrón es un modelo que se puede seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. (11)

Los patrones de diseño constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan. (12)

1.4.1 Patrones GRASP

Los patrones GRASP (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y aplicable. (14)

Los patrones básicos GRASP se refieren a cuestiones y aspectos fundamentales del diseño, ellos son:

- **Experto:** Este patrón se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- **Creador:** Este patrón se encarga de asignarle a una clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A.
- **Alta cohesión:** Este patrón se encarga de asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. 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. El patrón alta cohesión presenta semejanzas con el mundo real, si alguna persona se le asignan demasiadas responsabilidades no será eficiente en ninguna de ellas.
- **Bajo acoplamiento:** Este patrón se encarga de asignar una responsabilidad para mantener bajo acoplamiento. Las clases deben comunicarse con un número pequeño de clases siempre que sea posible.
- **Controlador:** Este patrón se encarga de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.

1.4.2 Patrones GoF

Los patrones GoF (Gang of Four) se dividen en 3 categorías las cuales se mencionan a continuación:

- Patrones de creación: Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos un objeto delegará responsabilidades.
- Patrones estructurales: Describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.
- Patrones de comportamiento: Se utilizan para organizar, manejar y combinar comportamientos.

En el desarrollo de este trabajo de diploma se utiliza el framework Symfony el cual utiliza una serie de patrones GoF.

Uno de los tipos de patrones de creación que utiliza Symfony es el Singleton (Instancia única), el cual garantiza que una clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma.

En la categoría de patrones estructurales, Symfony utiliza el patrón Decorator (Envoltorio). Dicho patrón responde a la necesidad de añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades.

Otro patrón utilizado por Symfony perteneciente a la categoría de comportamiento es el Command (Acción) el cual permite que diferentes objetos puedan ejecutar la misma acción sin necesidad de repetir su declaración e implementación.

1.4.3 Patrones de arquitectura

Los patrones de arquitectura ayudan a especificar la estructura fundamental de una aplicación. Expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Cada patrón de arquitectura ayuda a conseguir una propiedad específica en el sistema global, por ejemplo, la adaptabilidad de la interfaz de usuario. Dentro de los patrones de arquitectura se puede encontrar el patrón Modelo Vista Controlador (MVC) y el patrón modelo de tres capas.

En el desarrollo de este trabajo de diploma se utiliza el framework Symfony, el cual está basado en el patrón MVC, que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio, que esta 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.

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, por ejemplo, del tipo de gestor de bases de datos utilizado por la aplicación. (15)

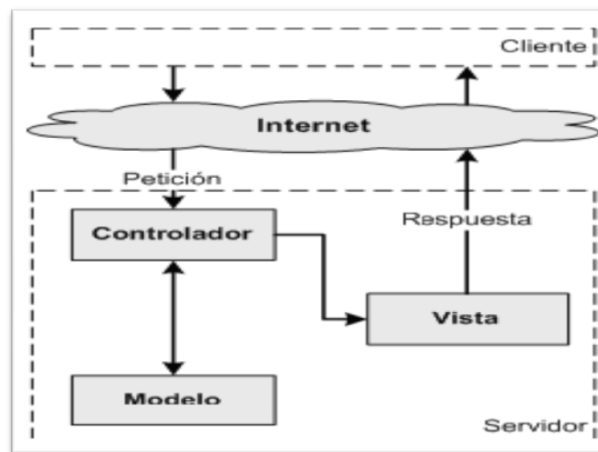


Figura 1. Modelo Vista Controlador

1.5 Conclusiones del capítulo

De acuerdo a lo analizado anteriormente se llegó a la conclusión de que la versión existente del Registro Cubano de Retraso Mental no se podía integrar a la Red de Salud de Cuba, provocando que el mismo no pueda ser utilizado por todos los genetistas del país. Se realizó un análisis de los antecedentes y los sistemas existentes para la gestión de la información de las personas que padecen de retraso mental, lo que permitió conocer las funcionalidades de esos sistemas desarrollados en Cuba y en el mundo. Para obtener un buen desarrollo de esta nueva versión del Registro Cubano de Retraso Mental se realizó una fundamentación de la metodología, tecnologías y herramientas a utilizar, todo esto con el objetivo de garantizar una adecuada prestación de los servicios a los genetistas que utilicen dicho registro.

CAPÍTULO 2: DISEÑO DEL SISTEMA

En este capítulo se definen los requisitos funcionales y no funcionales que tiene el registro, se identifican los casos de uso y los actores del sistema a desarrollar y se brinda una breve descripción de estos casos de usos. También se definen las clases del diseño del registro, se realizarán los diagramas de clases del diseño y los diagramas de interacción del diseño. Se comentan los patrones de diseño y de arquitectura aplicados. Además se trata el tratamiento de errores y la seguridad del registro.

2.1 Características del sistema

2.1.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Estos no alteran la funcionalidad del producto, es decir que se mantienen invariables sin importarle con que propiedades o cualidades se relacionen. (17)

Con ellos, se pretende determinar de manera clara y concisa lo que debe hacer el sistema siguiendo un enfoque funcional. Los requerimientos funcionales del registro son:

1. Insertar datos complementarios del paciente.
2. Modificar datos complementarios del paciente.
3. Generar reportes.
 - 3.1. Obtener reportes de los datos generales del Instrumento Clínico Genético
 - 3.1.1. Determinar la cantidad de personas con retraso mental que existen por provincias, municipios y consejo popular según el diagnóstico de su causa de retraso mental.
 - 3.1.2. Determinar la cantidad de personas con retraso mental que existen por provincias, municipios y consejo popular según la atención en algún servicio de genética.
 - 3.1.3. Mostrar las visitas a los centros según el nombre de la institución y el total de visitas realizadas al mismo.

2.1.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Constituyen las características que hacen al producto atractivo, usable, rápido o confiable al usuario. (17)

El registro debe cumplir ciertas características que se resumen en la siguiente lista de requisitos no funcionales:

1. Apariencia o interfaz externa:

Se deben utilizar imágenes y colores identificados con el negocio del sistema. La interfaz externa debe estar diseñada para verse en cualquier resolución igual o superior a 1024x768.

2. Usabilidad:

La aplicación informática debe garantizar un acceso fácil y rápido, contando con un menú que satisfaga las necesidades de los usuarios. Este podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora y del ambiente web.

3. Rendimiento:

Los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información.

4. Soporte:

Se debe asegurar el soporte para los usuarios de manera que se puedan satisfacer sus necesidades a partir de mejoras, una vez puesta en marcha la aplicación. Para ello se crearán una serie de manuales de usuarios y videos tutoriales, y se mantendrá la asistencia a los usuarios.

5. Seguridad:

El sistema debe tener un mecanismo propio para gestionar la seguridad a través de niveles de acceso a la información. Los permisos al ejecutar cualquier acción deben estar de acuerdo con el nivel jerárquico de acceso que presente el usuario en cada módulo, el cual es definido por los administradores del sistema.

6. Software:

Se requiere para el funcionamiento del sistema disponer de un servidor que cuente con Sistema Operativo Linux, Apache 2.0 y MySQL 5.0 o versiones superiores. Los usuarios del sistema deberán contar con un navegador Internet Explorer 5.5 o Mozilla Firefox 2.0 o superior, para poder acceder a las opciones que brinda el sistema.

7. Hardware

Para el desarrollo y ejecución de la aplicación se necesitará:

Para el servidor de aplicación:

- Microprocesador Pentium IV a 3.0 GHz o superior
- 1 GB de memoria RAM o superior

Para el cliente:

- Microprocesador Pentium a 233 MHz o superior
- 64 MB de memoria RAM o superior
- 52 MB de espacio libre del disco duro

Conexión al servidor a través de un MODEM o tarjeta de red.

8. Disponibilidad: El sistema debe ser capaz de funcionar por si solo en caso de que los servicios de los diferentes componentes de SISalud no estén disponibles. Se debe garantizar el funcionamiento de la aplicación durante las 24 horas del día y los siete días de la semana, con el menor tiempo posible de recuperación de fallos. Se deben crear copias de respaldo periódicas que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.

9. Requisitos Legales: Las herramientas y las tecnologías en que estará basada la aplicación informática deberán cumplir con las licencias de software libre.

10. Persistencia: La información debe almacenarse en bases de datos con carácter permanente con el objetivo de poder realizar análisis de la misma con el transcurso de los años.

2.1.3 Diagrama de casos de uso del sistema

Los diagramas de casos de uso se crean para visualizar las relaciones entre los actores y los casos de uso.

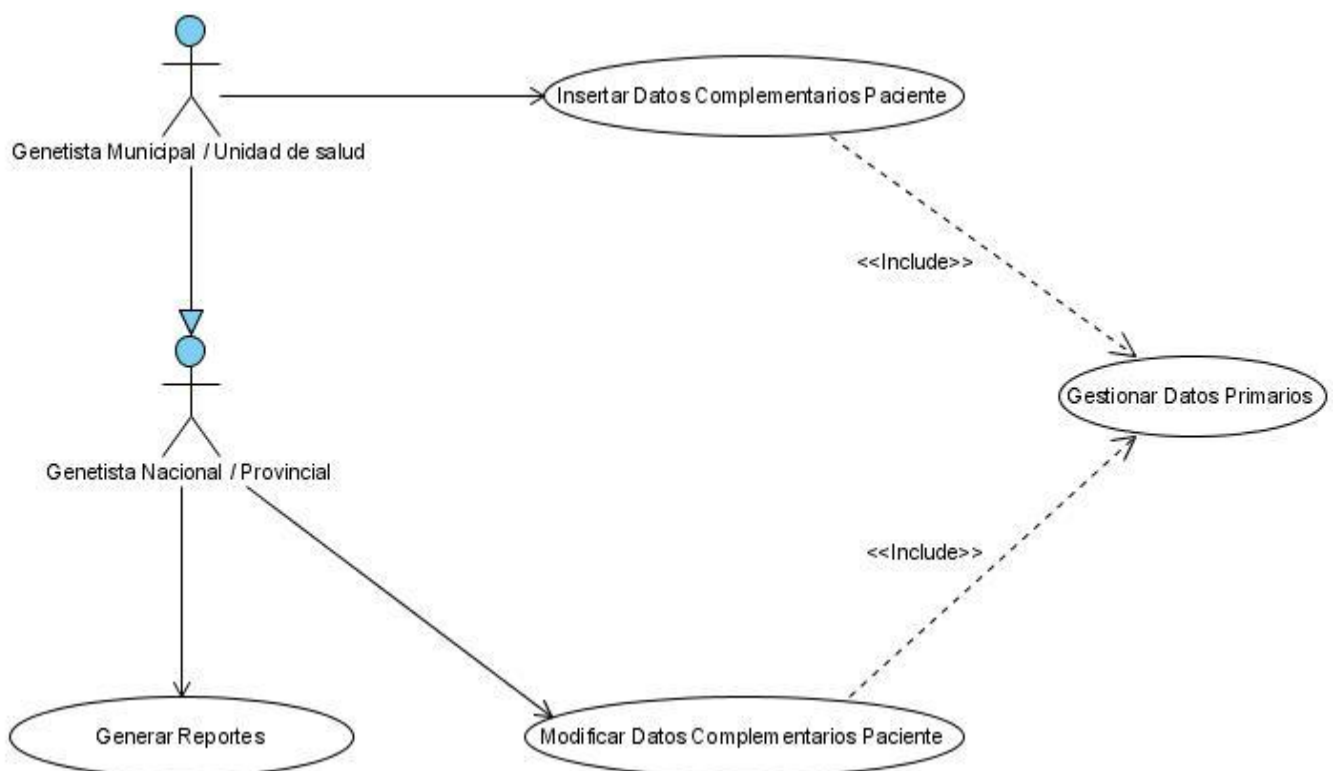


Figura 2. Diagrama de casos de uso del sistema

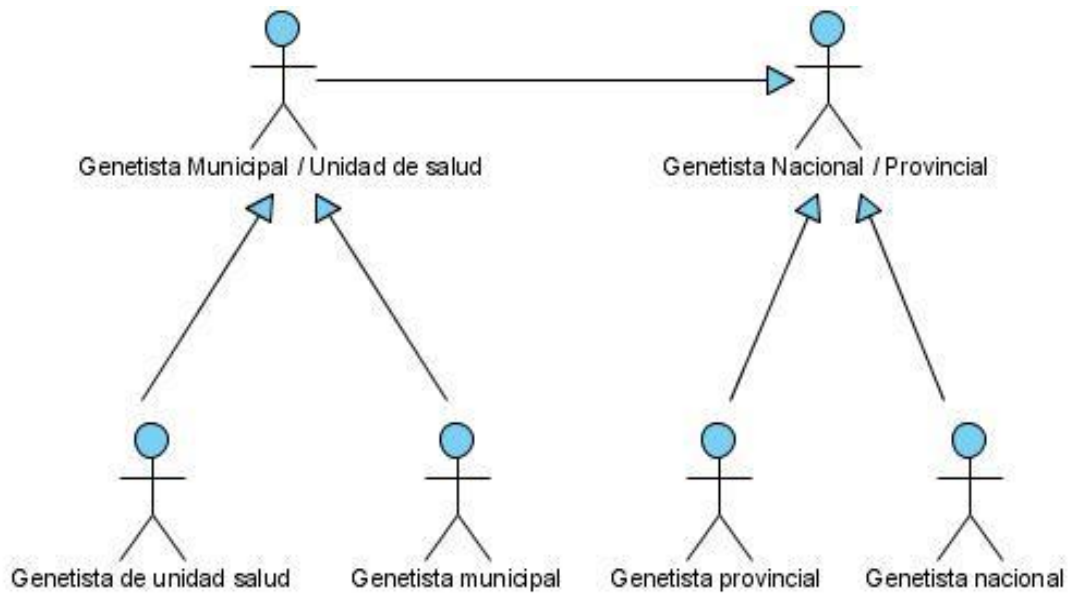


Figura 3. Diagrama de actores del sistema

Actores del sistema y su justificación.

Actor	Descripción
Genetista de Unidad de Salud	Es el encargado de insertar los datos complementarios, modificar los datos complementarios y obtener los reportes de los datos de los pacientes pertenecientes a una unidad de salud.
Genetista Municipal	Es el encargado de insertar los datos complementarios, modificar los datos complementarios y obtener los reportes de los datos de los pacientes pertenecientes a un municipio.
Genetista Provincial	Es el encargado de obtener los reportes y modificar los datos complementarios de los pacientes a nivel provincial.

Genetista Nacional	Es el encargado de obtener los reportes y modificar los datos complementarios de los pacientes a nivel nacional.
Genetista Nacional / Provincial	Es una generalización del Genetista de Unidad de Salud, del Genetista Municipal, del Genetista Provincial y del Genetista Nacional.
Genetista Municipal / Unidad de salud	Es una generalización del Genetista de Unidad de Salud y del Genetista Municipal.

Tabla 1: Justificación de los actores del sistema

2.1.4 Breve descripción de los casos de uso del sistema

CU-1	Insertar datos complementarios del paciente
Actores	Genetista Municipal / Unidad de salud (Inicia)
Resumen	El caso de uso se inicia cuando el Genetista Municipal o de Unidad de salud va a insertar los datos complementarios recogidos de un paciente. El registro muestra la interfaz correspondiente para la búsqueda de un paciente y permite seleccionarlo, mostrando una nueva interfaz para insertar los datos complementarios del paciente. El caso de uso finaliza cuando se emite un mensaje informando que los datos fueron insertados correctamente.
Referencia	RF1, Gestionar Datos Primarios (include).

Tabla 2: Breve descripción del caso de uso del sistema: **Insertar datos complementarios del paciente**

CU-2	Modificar datos complementarios del paciente
Actor	Genetista Nacional / Provincial (Inicia)

Resumen	El caso de uso se inicia cuando el Genetista Nacional o Provincial va a modificar alguno de los datos complementarios existentes en el registro del paciente. El registro muestra la interfaz correspondiente para la búsqueda de un paciente y permite seleccionarlo, mostrando una nueva interfaz para modificar los datos complementarios del paciente. El caso de uso finaliza cuando se emite un mensaje informando que los datos fueron modificados correctamente.
Referencia	RF2, Gestionar Datos Primarios (include).

Tabla 3: Breve descripción del caso de uso del sistema: **Modificar datos complementarios del paciente**

CU-3	Generar reportes
Actor	Genetista Nacional / Provincial (Inicia)
Resumen	El caso de uso se inicia cuando el Genetista Nacional o Provincial va a solicitar algún reporte. El registro le muestra la interfaz correspondiente para escoger un criterio de búsqueda. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
Referencia	RF3.1, 3.1.1, 3.1.2, 3.1.3

Tabla 4: Breve descripción del caso de uso del sistema: **Generar reportes**

El caso de uso Gestionar datos primarios pertenece al Registro Cubano de Historias Clínicas (RECUHCL). La descripción completa de este caso de uso se puede encontrar en el Expediente de proyecto de dicho registro.

CU-4	Gestionar datos primarios
Actor	Especialista Genético (Inicia), Registro de Ciudadano (RC)
Resumen	Registrar los datos personales de los pacientes del Centro Nacional de Genética Médica.
Referencia	2.1.1, 2.1.2, 2.1.3

Tabla 5: Breve descripción del caso de uso del sistema: **Gestionar datos primarios**

2.1.5 Descripción detallada de los casos de uso del sistema.

(Ver Anexo 2)

2.2 Pautas del Diseño.

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

1. Los formularios que sean creados deben estar centrados, con un ancho de un 90 % y con bordes de valor 0.
2. La primera fila del formulario debe contener el nombre de la operación que se pretende realizar con el formulario.
3. Cuando se le solicita datos al usuario se hará a través de una tabla de dos columnas. 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.
4. Los botones para efectuar operaciones sobre el formulario se ubicarán en la parte inferior derecha del formulario. Se posicionarán de izquierda a derecha teniendo en cuenta el peso de la operación que representan.
5. 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.
6. 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.

Componentes	Estilos
TextBox	Entradaplana
ComboBox	Entradaplana
Botones	Sbbtn
ListBox	Entradaplana
NombredeCampos	class="nombrecampo"

Tabla 5: Estilos a usar en los componentes web.

En este epígrafe se han plasmado algunas de las pautas de diseño, para una mayor información de las mismas se puede encontrar una descripción más detallada en el documento de las pautas del diseño definido por el grupo de arquitectura del Sistema Informático de Genética Médica. (16)

2.3 Patrones aplicados

2.3.1 Patrones GRASP aplicados

➤ Experto

Este es uno de los patrones que se utilizan al trabajar con el framework Symfony, un ejemplo de esto es la inclusión de Propel para mapear la base de datos. Este genera las clases para la gestión de las entidades con las responsabilidades debidamente asignadas según el patrón Experto, pues cada una de estas clases cuenta con un conjunto de funcionalidades relacionadas directamente con la entidad que representan.

➤ Creador

La clase `rmActions` es la que contiene las acciones del módulo retraso mental y es la encargada de ejecutar las mismas. En dichas acciones se crean los objetos de las clases que representan las entidades, por lo cual la clase `rmActions` es “creador” de dichas entidades, por ejemplo, en la acción `executeInsertar` se evidencia el uso de este patrón mediante la creación de las instancias de las clases entidades que contienen los datos de la persona con retraso mental a insertar (`rm_parentesco`, `rm_calor`, etc).

➤ Alta cohesión

Una de las características principales del framework Symfony es la organización del trabajo en el mismo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo, la clase `rmActions` contiene varias funcionalidades en las que cada una posee un propósito único, no desempeñado por el resto de los elementos, siendo estas funcionalidades las encargadas de controlar las acciones de las plantillas. Esto hace posible que el software sea flexible a cambios sustanciales con efecto mínimo.

➤ Bajo acoplamiento

Este patrón se evidencia dentro del framework Symfony en la capa modelo ya que las clases de acceso a los datos tienen bastante independencia de las clases de abstracción de datos. Hay poca dependencia entre esas clases lo que permite una mayor reutilización.

➤ Controlador:

Symfony es un framework basado en el patrón arquitectónico Modelo Vista Controlador, que define una capa específica para los controladores, que son el núcleo del mismo. Este patrón se puede observar en las clases `sfFrontController`, `sfWebFrontController`, `sfContext` propias de Symfony. Symfony también aplica el patrón “Front Controller” (Controlador frontal) y tiene una estructura bien organizada de sus controladores que parte desde la página “`indexRMSuccess.php`” y se cumplimenta en la clase `rmActions`. Cada clase de esta capa tiene

su responsabilidad y es única, por ejemplo, hay controladores que se encargan de la seguridad del sistema trabajando con ficheros .yml.

2.3.2 Patrones GoF aplicados

Patrón Singleton

Este patrón se aplica en el método getInstance de la clase sfRouting, el cual garantiza que esa clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma. La clase sfRouting es muy utilizada porque es la encargada de enrutar todas las peticiones que se hagan al registro.

Patrón Decorador

Este patrón se observa en el archivo denominado layout.php que contiene el Layout (plantilla global) de todas las páginas del registro, el cual permite añadir funcionalidades a las vistas dinámicamente. El mismo almacena el código HTML que es común a todas las páginas del registro, para no tener que repetirlo en cada página, por lo que el Layout decora la plantilla.

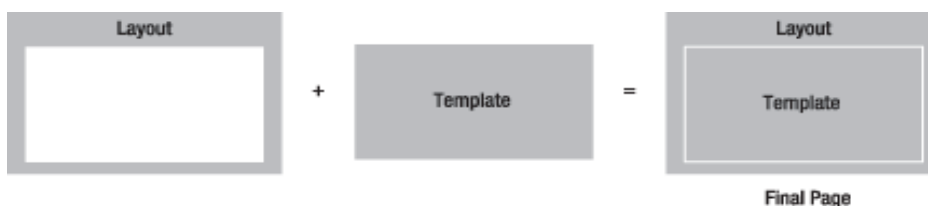


Figura 4: Planilla decorada con un Layout.

Patrón Command

Este patrón se aplica en el método dispatch() de la clase sfWebFrontController que es la encargada de determinar cual módulo y acción usar en dependencia de la petición del usuario, de esta forma se pone de manifiesto que diferentes objetos pueden ejecutar la misma acción sin necesidad de repetir su declaración e implementación.

2.3.3 Patrón de arquitectura aplicado

El framework Symfony utiliza el patrón arquitectónico MVC por lo que obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por este patrón. 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. Aplicar el patrón MVC a una aplicación resulta bastante útil además de restrictivo.

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, etc.)
- 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. (15)

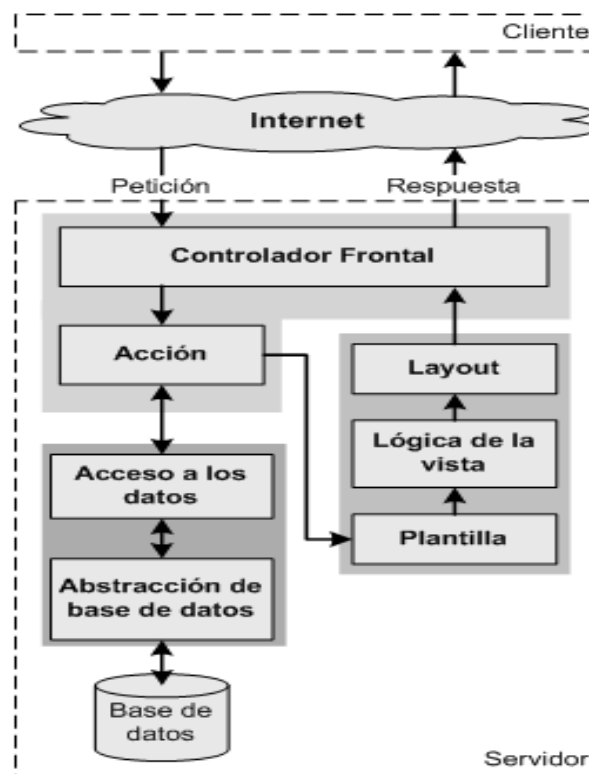


Figura 5. Flujo de trabajo de Symfony

2.4 Modelo de diseño.

El propósito del diseño es modelar el sistema y encontrar su forma para que soporte todos los requisitos. El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software. (18)

2.4.1 Descripción de las clases del diseño.

Nombre de la Clase: rmActions
Nombre del método: MostrarDatosPrimarios()
Descripción: Se encarga de mostrar los datos primarios de un paciente.
Nombre del método: CargarDatosPrimarios()
Descripción: Se encarga de mostrar los datos complementarios de un paciente.
Nombre del método: executeIndexRM()
Descripción: Es la página principal del registro.
Nombre del método: executeInsertar()
Descripción: Se encarga de insertar los datos complementarios de un paciente.
Nombre del método: executeNoAntesInsertado()
Descripción: Se encarga de mostrar un mensaje cuando el paciente seleccionado no ha sido insertado como un retrasado mental.
Nombre del método: executeCorrecto()
Descripción: Se encarga de mostrar un mensaje cuando se han insertado correctamente los datos complementarios del paciente.
Nombre del método: executePersonalInsertada()
Descripción: Se encarga de mostrar un mensaje cuando se selecciona un paciente que ya ha sido insertado con anterioridad como retrasado mental.
Nombre del método: executeBuscarInsertar()
Descripción: Se encarga de buscar un paciente que se desea insertar como retrasado mental.
Nombre del método: executeDatosModificados()
Descripción: Se encarga de mostrar un mensaje cuando los datos se han modificado correctamente.
Nombre del método: executeBuscarModificar()
Descripción: Se encarga de buscar un paciente que se desea modificar.
Nombre del método: handleErrorInsertar()
Descripción: Se encarga de mostrar mensajes de error si existen errores en la validación.
Nombre del método: handleErrorModificar()
Descripción: Se encarga de mostrar mensajes de error si existen errores en la validación.

Nombre del método: handleErrorReporteDiagnostico()
Descripción: Se encarga de mostrar mensajes de error si existen errores en la validación.
Nombre del método: handleErrorReporteAtencion()
Descripción: Se encarga de mostrar mensajes de error si existen errores en la validación.
Nombre del método: handleErrorReporteVisitas()
Descripción: Se encarga de mostrar mensajes de error si existen errores en la validación.
Nombre del método: executeModificar()
Descripción: Se encarga de modificar los datos complementarios de un retrasado mental.
Nombre del método: executeReporteDiagnostico()
Descripción: Se encarga de mostrar un reporte de la cantidad de personas con retraso mental que existen por provincias, municipios y consejo popular según el diagnóstico de su causa de retraso mental.
Nombre del método: executeReporteAtencion()
Descripción: Se encarga de mostrar un reporte de la cantidad de personas con retraso mental que existen por provincias, municipios y consejo popular según la atención en algún servicio de genética.
Nombre del método: executeReporteVisitas()
Descripción: Se encarga de mostrar un reporte con las visitas a los centros según el nombre de la institución y el total de visitas realizadas a él.

Tabla 6: Descripción de los métodos de la clase rmActions.

2.4.2 Diagramas de clases web del diseño.

Un diagrama de clases del diseño representa las clases del diseño y sus objetos, así como los subsistemas del diseño. (13)

Al ser muchas las clases que componen la capa modelo en los diagramas de clases web del diseño de los casos de uso Insertar datos complementarios del paciente y Modificar datos complementarios del paciente mostrados a continuación solo se puso la representación de la clase rm_datos_clinico conjuntamente con las otras tres clases que genera Symfony de la misma, con el objetivo de que se pudiese entender las relaciones entre estas clases, con la clase rmActions y con el subsistema Propel. Para ver el resto de las relaciones entre las clases de la capa modelo remítase al Expediente de proyecto.

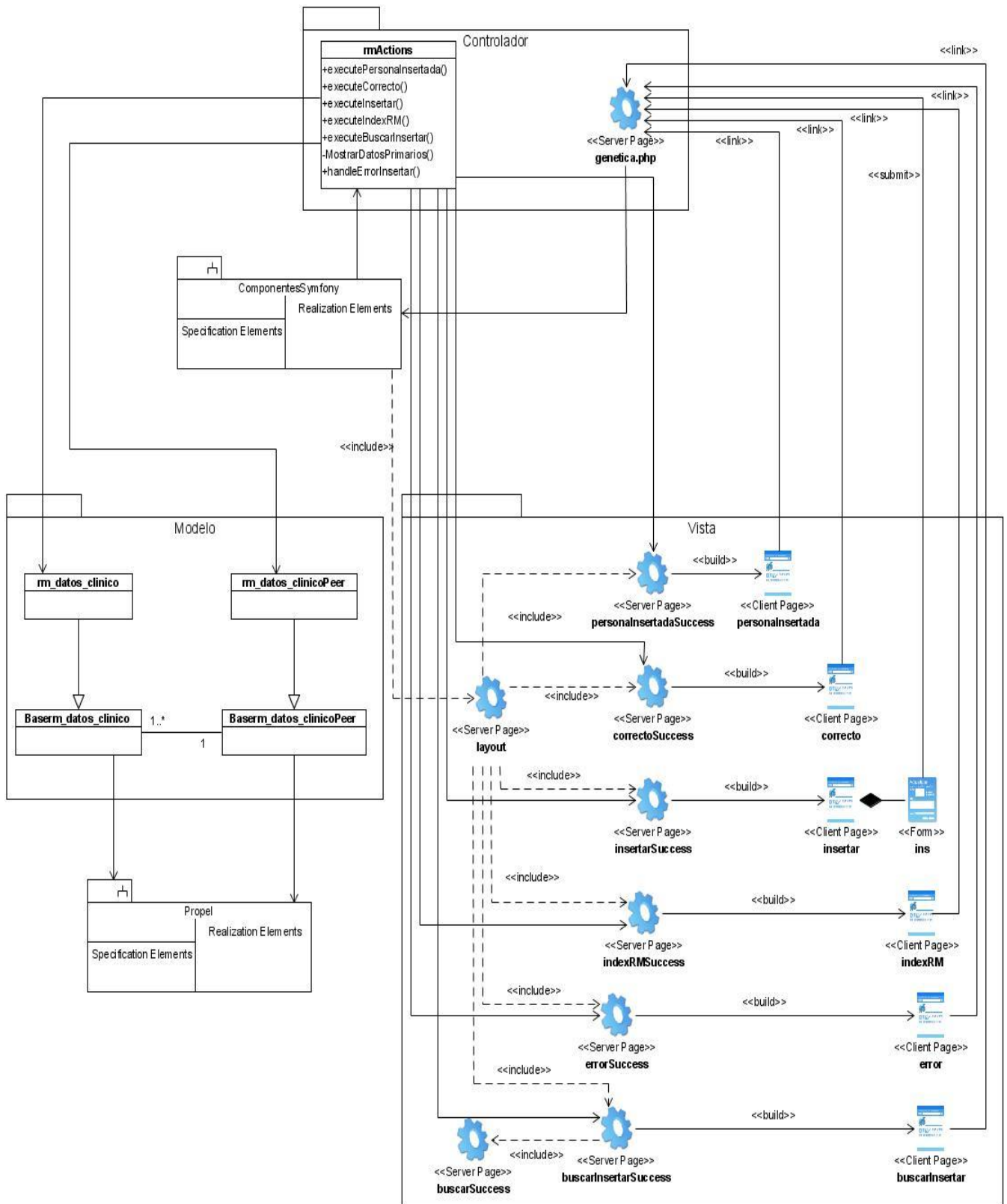


Figura 6. Diagrama de clases caso de uso Insertar datos complementarios del paciente

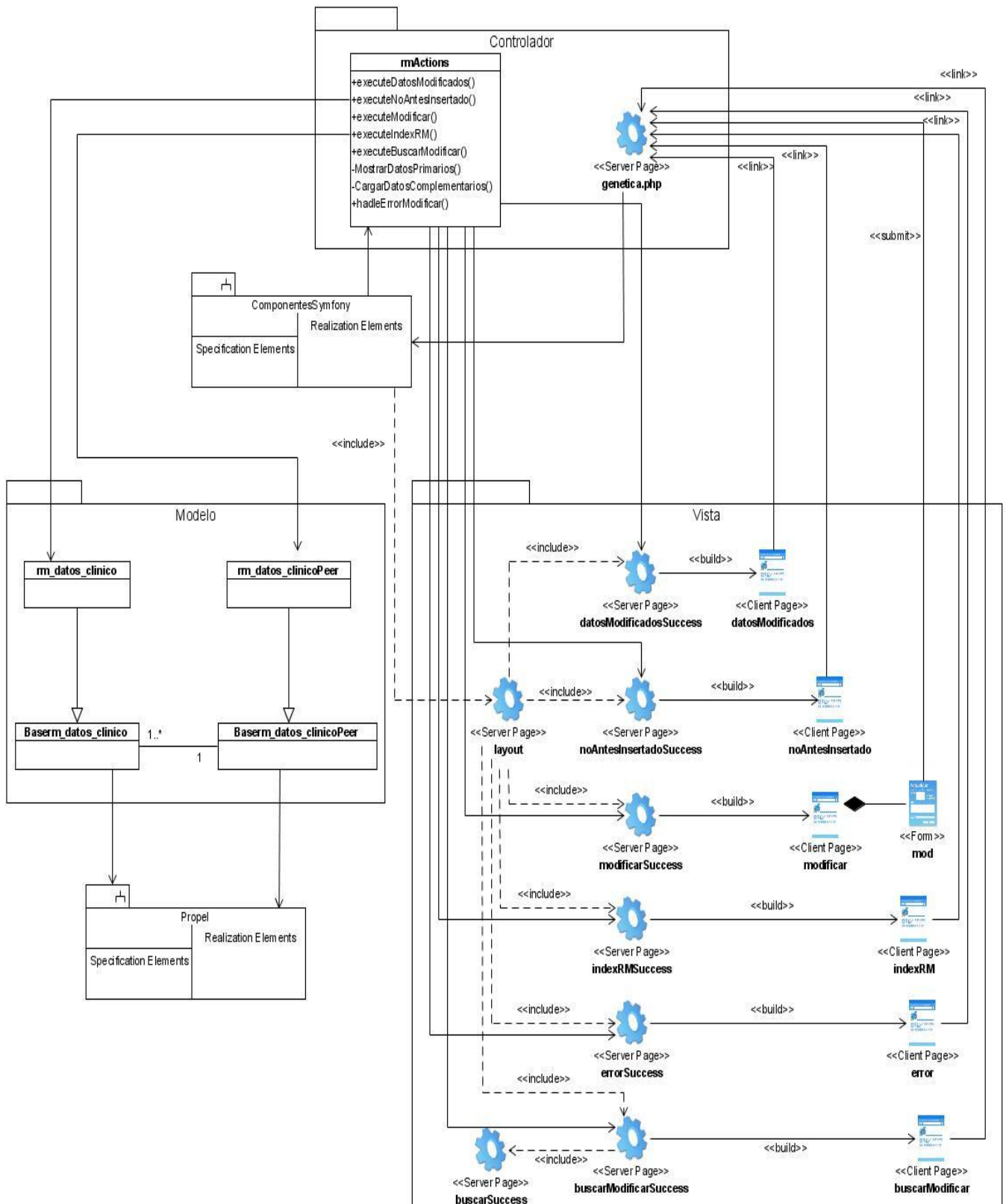


Figura 7. Diagrama de clases caso de uso Modificar datos complementarios del paciente

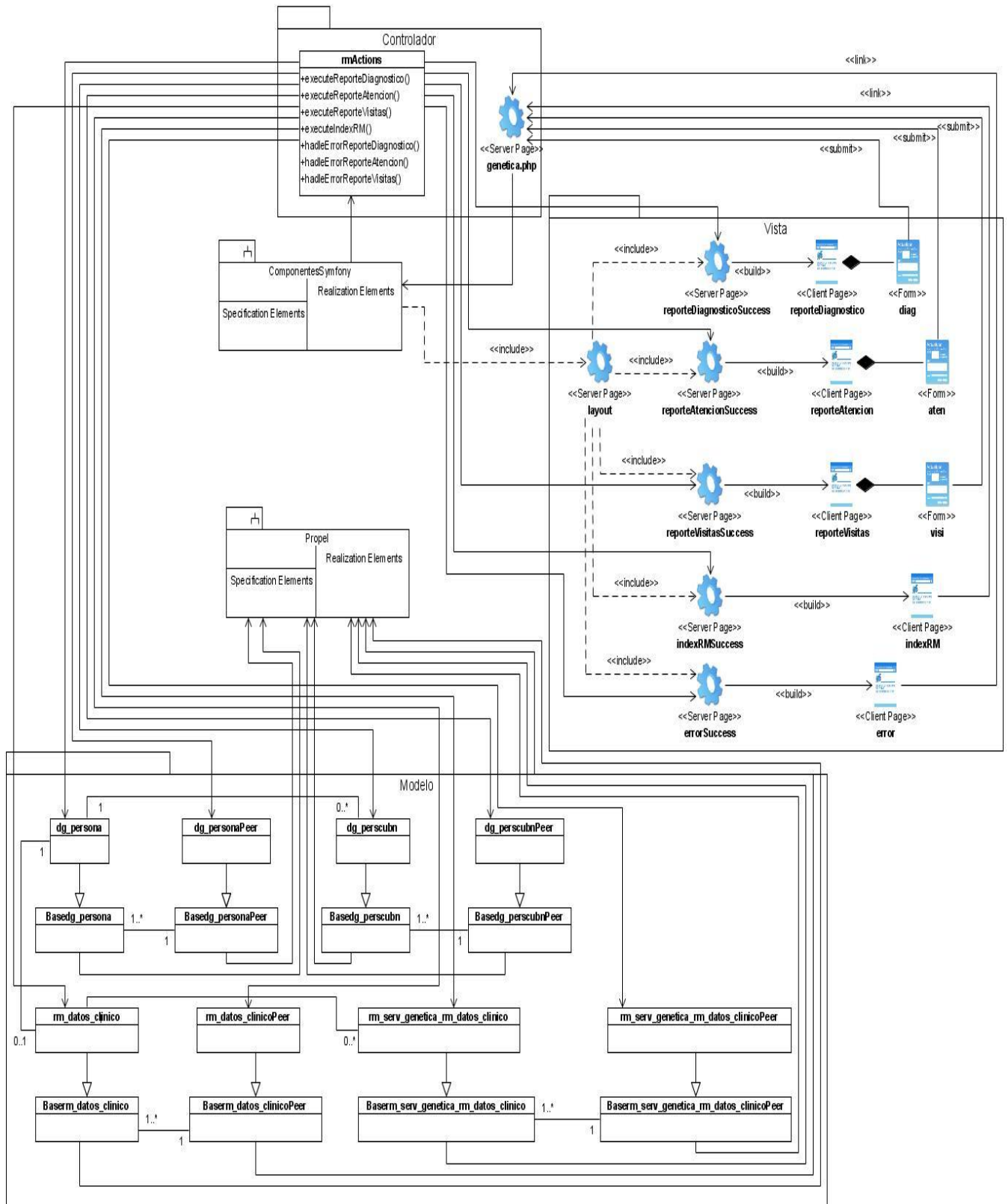


Figura 8. Diagrama de clases caso de uso Generar reportes

2.4.3 Diagramas de interacción.

Los diagramas de interacción muestran la relación que se establece mediante el paso de mensajes entre los distintos objetos que participan en un escenario. Modelan también el comportamiento dinámico del sistema, es decir, el flujo de control en una operación. Describen secuencia de acciones o colaboración entre objetos y son representados por diagramas de secuencia o de colaboración.

Diagramas de secuencia: Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar. (10)

Diagramas de colaboración: Muestra la interacción entre varios objetos y los enlaces que existen entre ellos. Representa las interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones. (10)

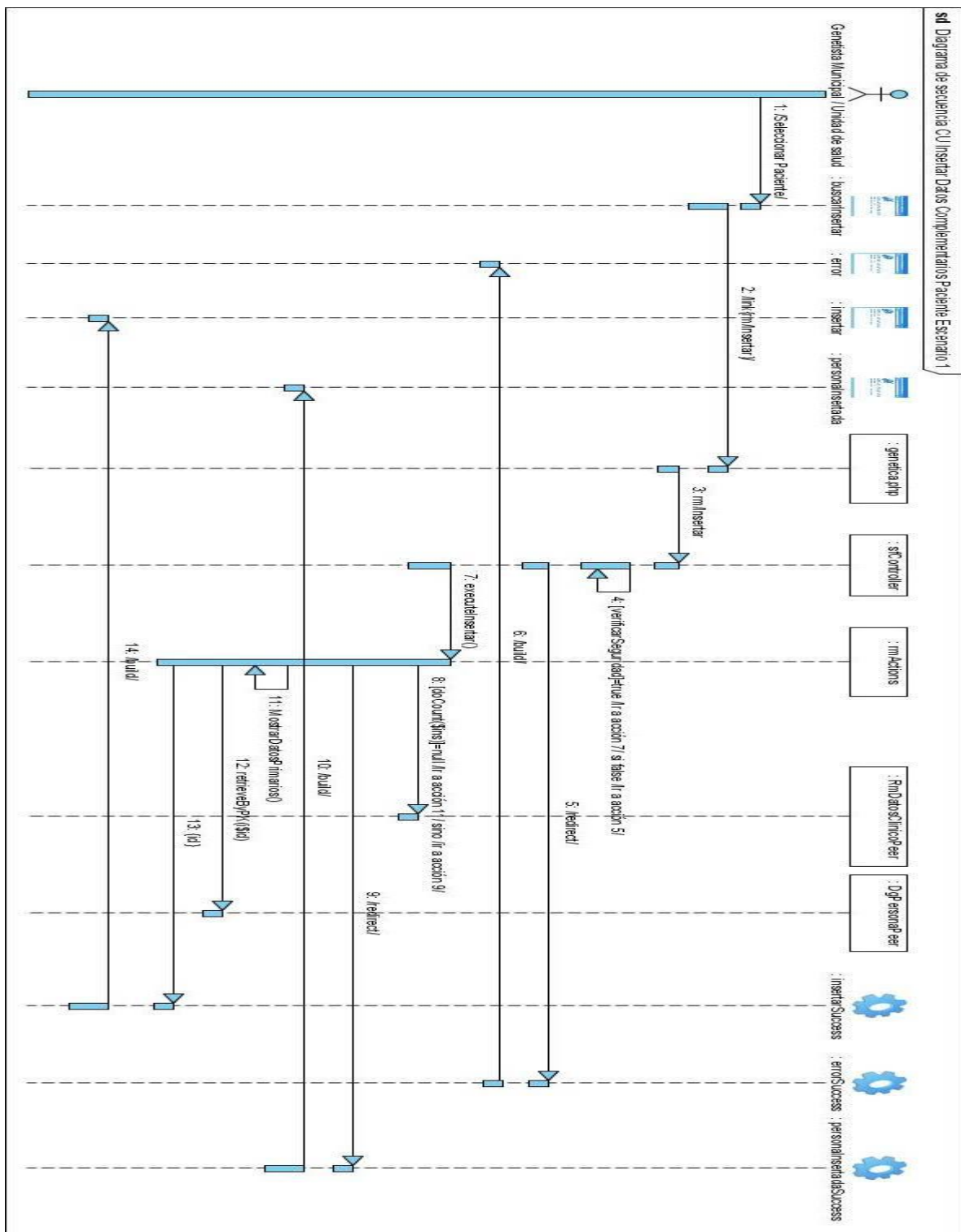


Figura 9. Diagrama de secuencia caso de uso Insertar datos complementarios del paciente, escenario

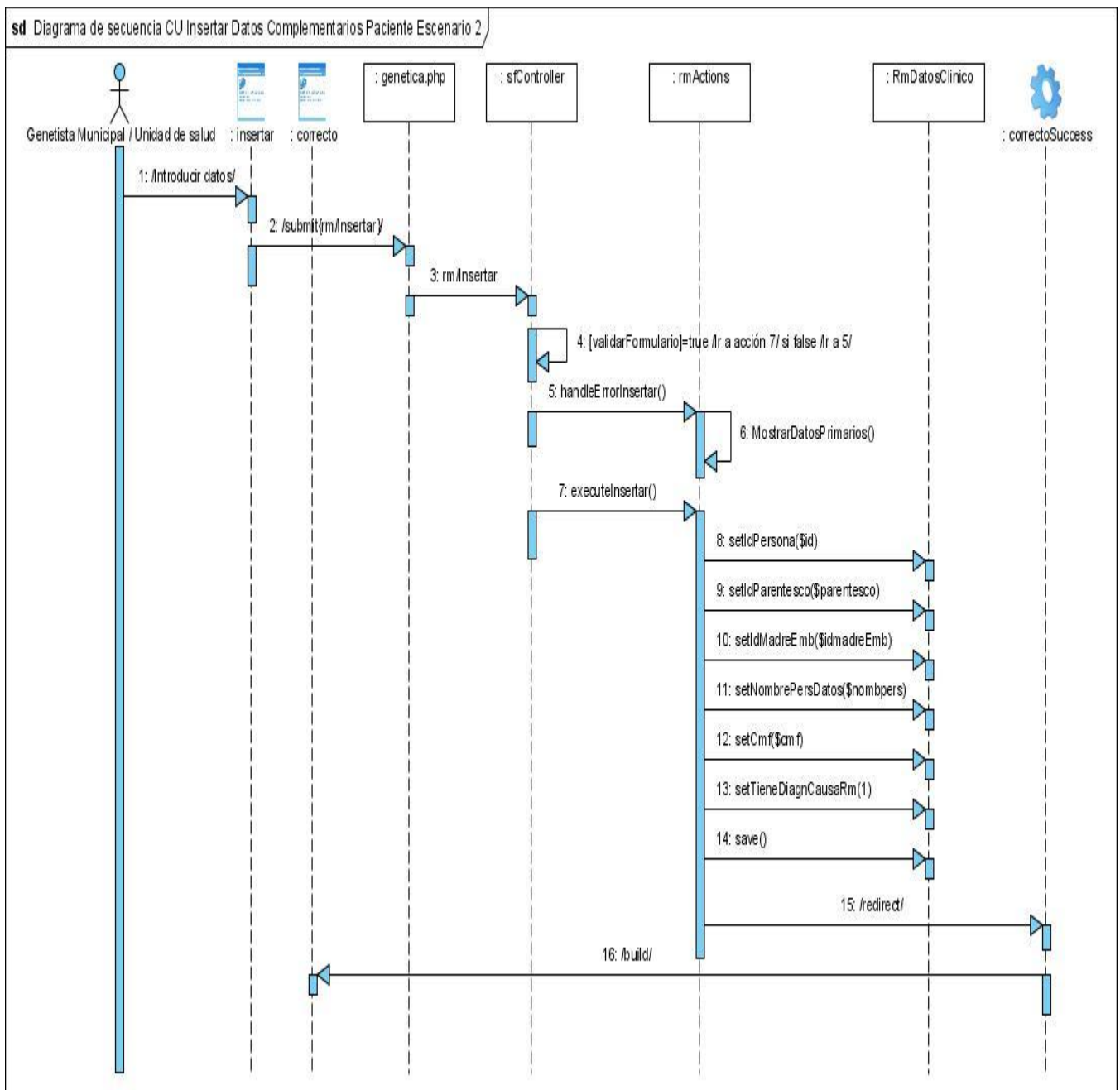


Figura 10. Diagrama de secuencia caso de uso Insertar datos complementarios del paciente, escenario 2

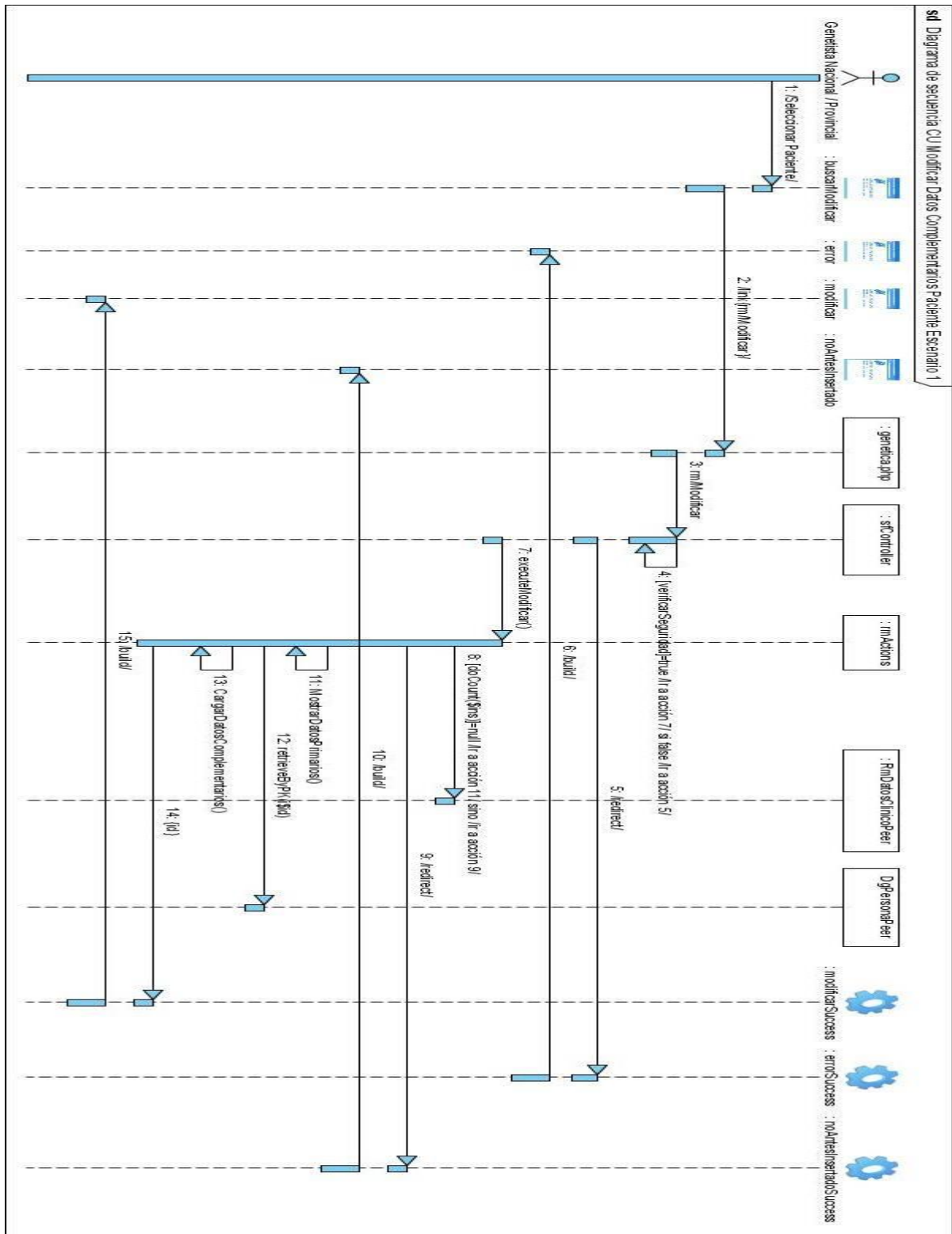


Figura 11. Diagrama de secuencia caso de uso Modificar datos complementarios del paciente, escenario 1

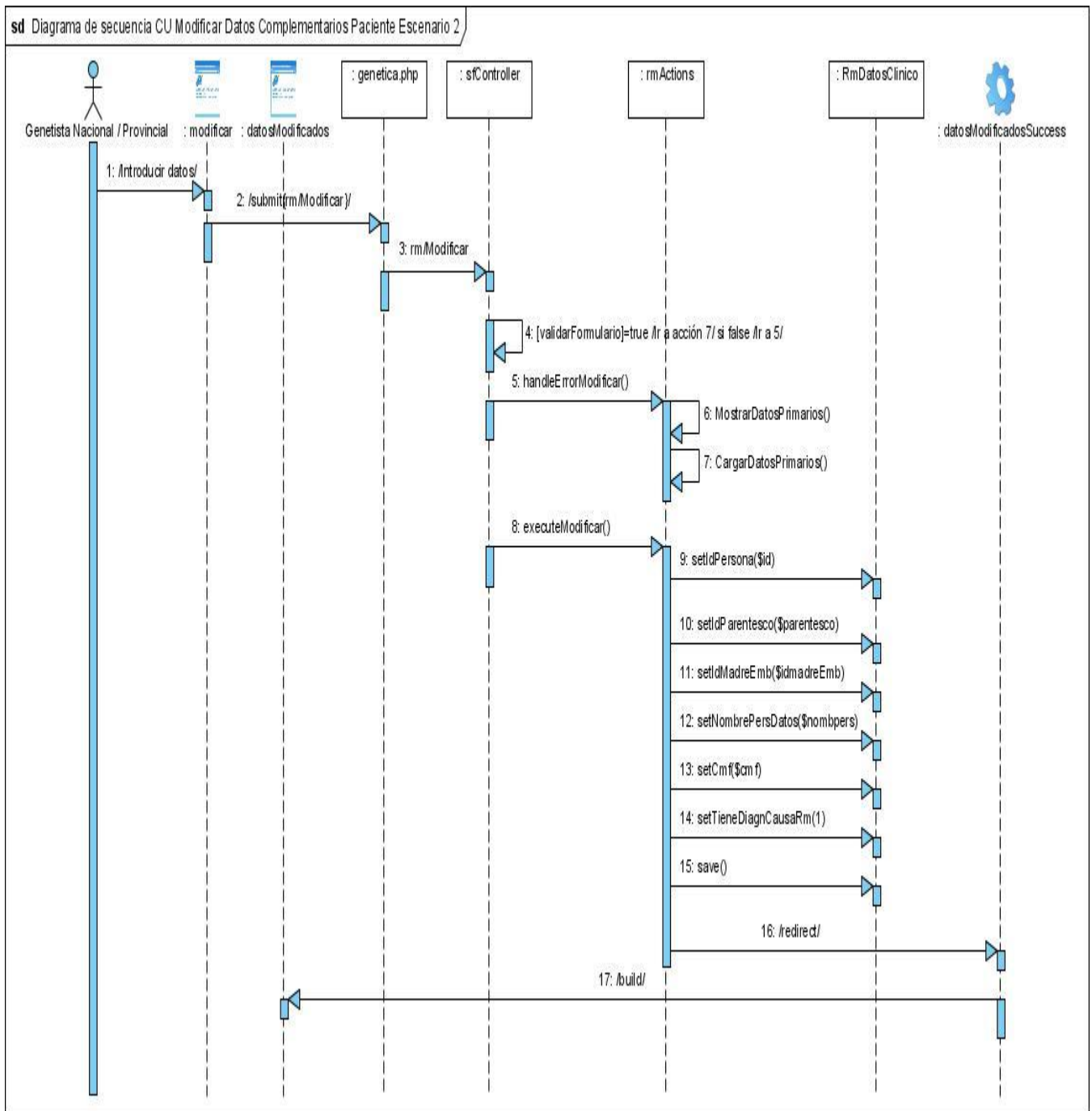


Figura 12. Diagrama de secuencia caso de uso Modificar datos complementarios del paciente, escenario 2

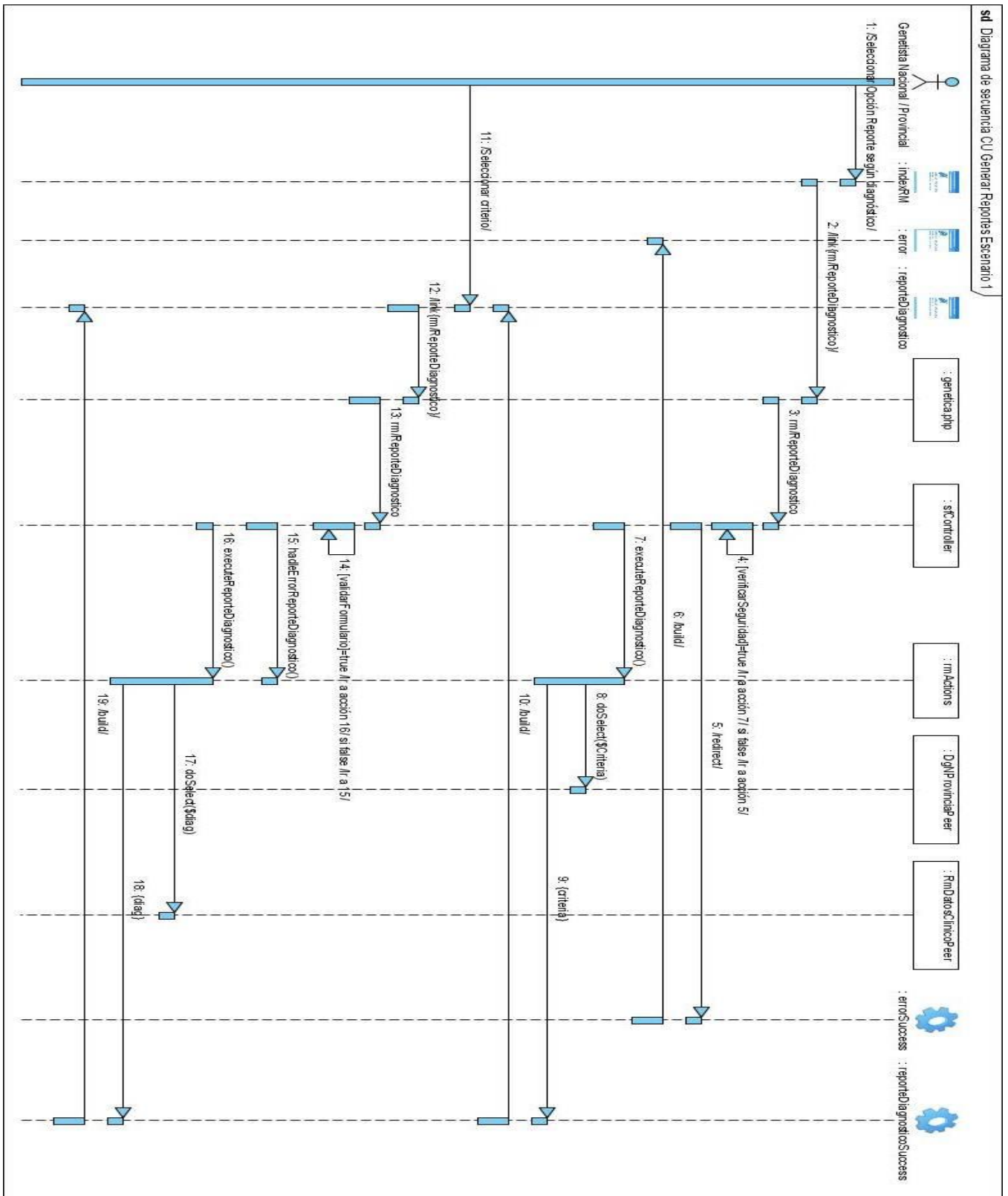


Figura 13. Diagrama de secuencia caso de uso Generar reportes, escenario 1

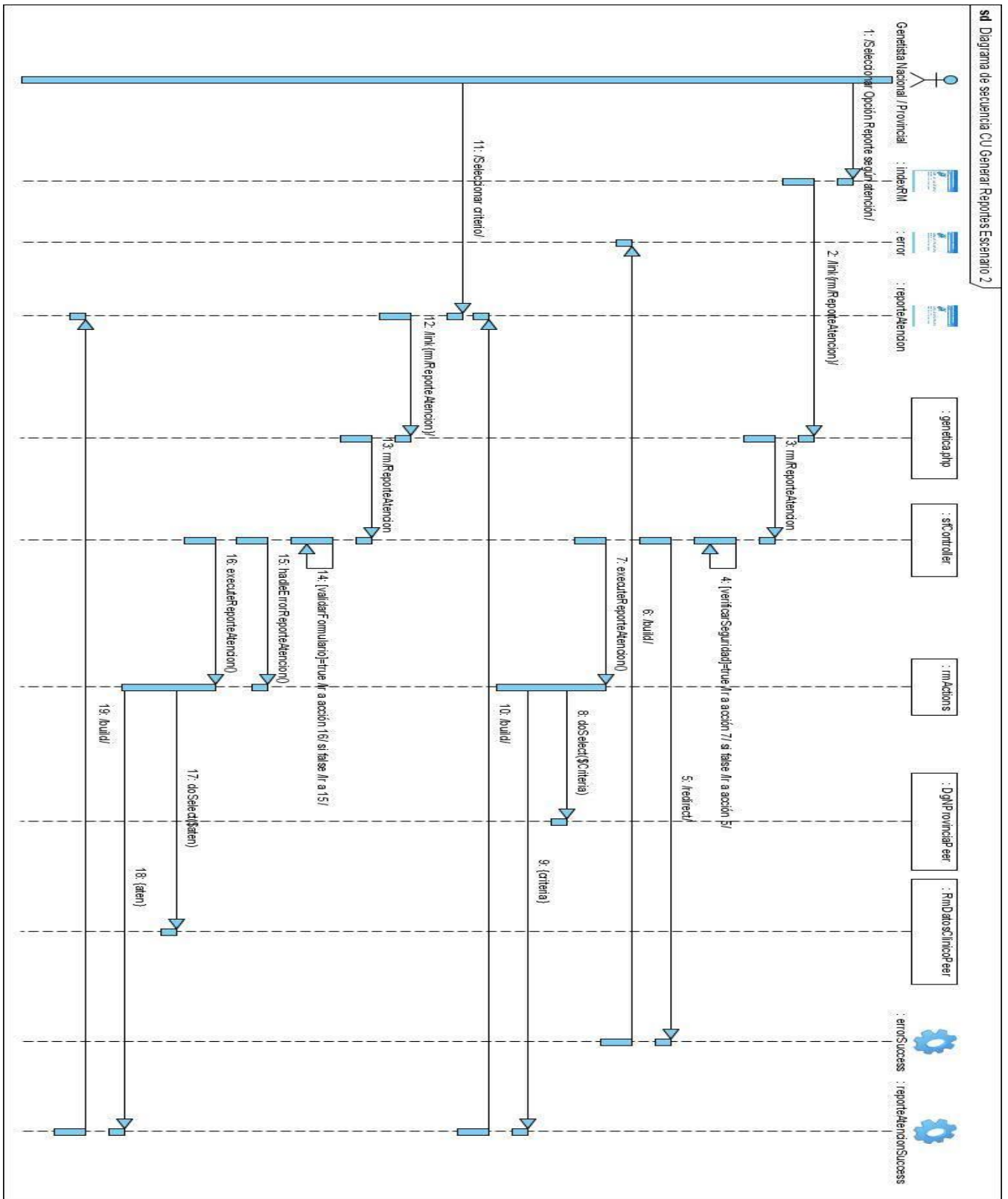


Figura 14. Diagrama de secuencia caso de uso Generar reportes, escenario 2

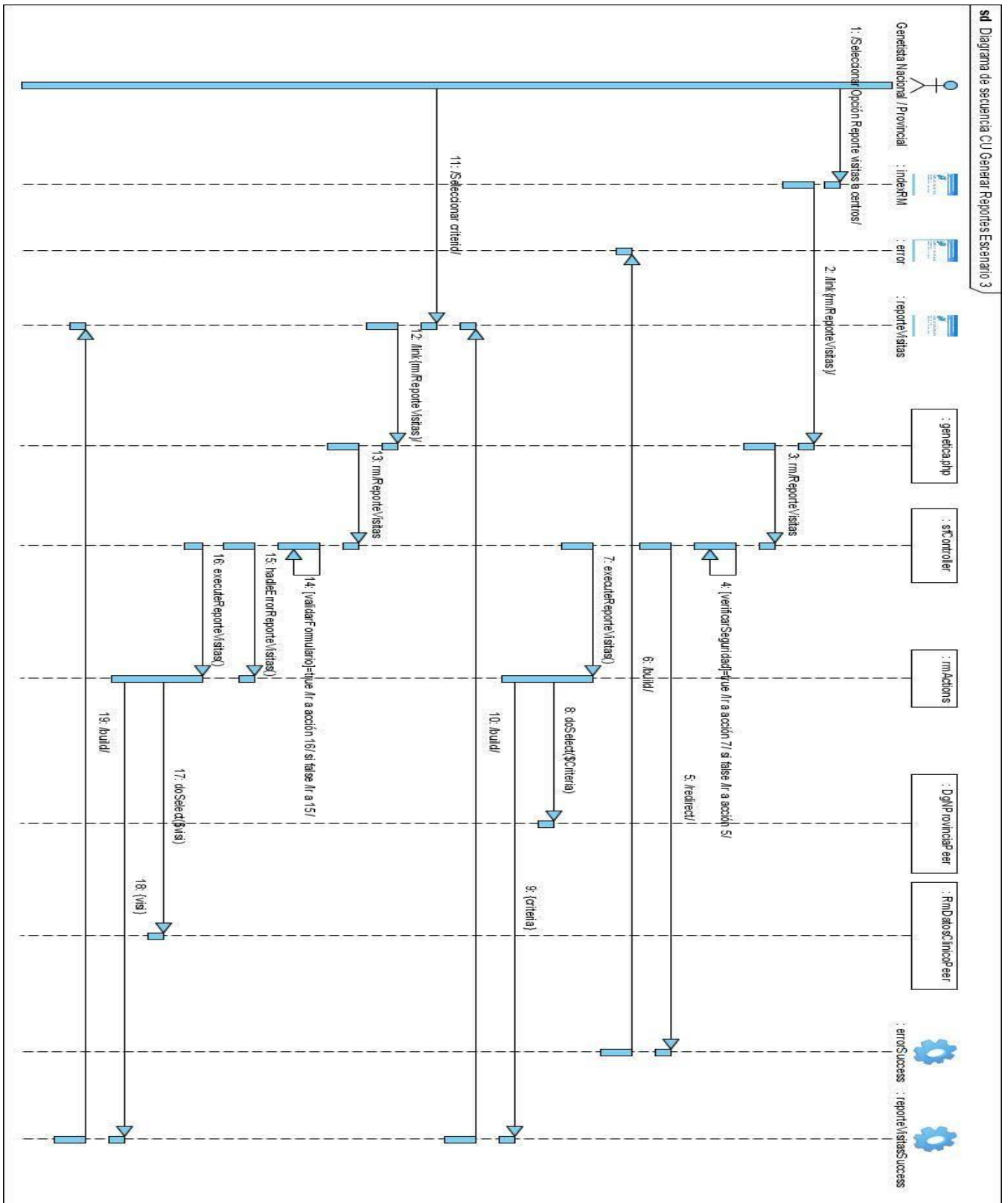


Figura 15. Diagrama de secuencia caso de uso Generar reportes, escenario 3

2.5 Modelo de despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo, se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos. (22)

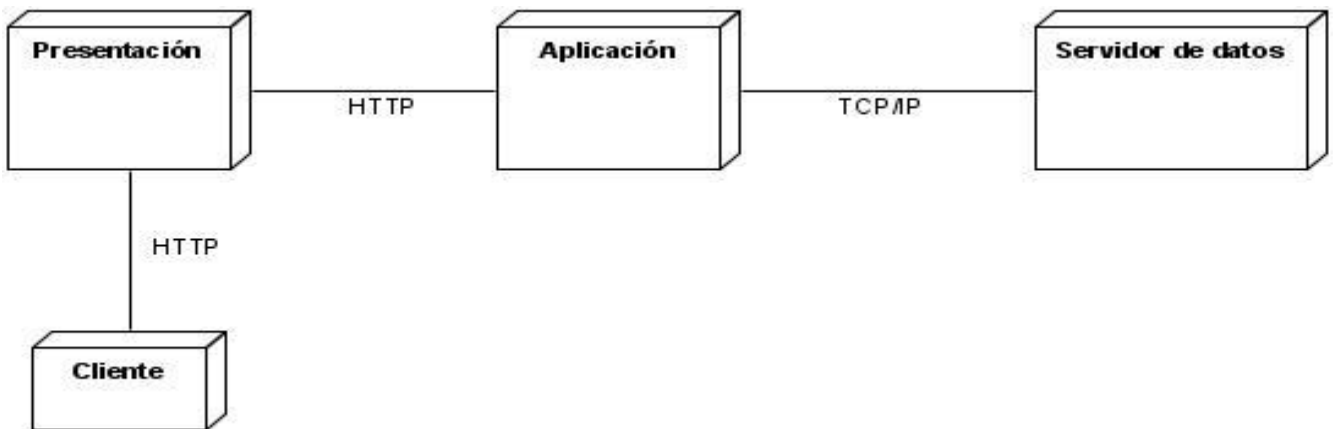


Figura 16. Modelo de despliegue

2.6 Tratamiento de errores

El registro contará con el tratamiento de errores, pues los datos que van a ser introducidos por el usuario serán validados en el lado del servidor mediante archivos .yaml vinculados a los formularios de las interfaces. Una vez que es detectado un error, ya sea un dato escrito de forma errónea, o algún dato que no fue introducido, se lanzará un mensaje de error, indicándole cuál es el dato que está mal o que falta y dirigirá al usuario al campo en el cuál está el error, el usuario podrá seguir introduciendo los datos en el formulario.

Además una vez que se hayan escrito correctamente en el formulario todos los datos es que se enviarán finalmente a la base de datos para ser guardados, contribuyendo de esta forma a que no existan pacientes en la base de datos con datos erróneos, o a medias.

2.7 Seguridad

El registro brinda la posibilidad de restringir la ejecución de una acción a usuarios con ciertos privilegios ya que necesitan estar autenticados antes de poder acceder a alguna acción específica. Añadir esta seguridad a una aplicación requiere dos pasos: declarar los requerimientos de seguridad para cada acción y autenticar a los usuarios con privilegios para que puedan acceder a estas acciones seguras.

Antes de ser ejecutada, cada acción pasa por un filtro especial que verifica si el usuario actual tiene privilegios de acceder a la acción requerida. 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.

Para restringir el acceso a una acción se crea y se edita un archivo de configuración YAML llamado `security.yml`. En este archivo, se especifican los requerimientos de seguridad que los usuarios deben satisfacer para cada acción.

2.8 Conclusiones del capítulo.

El principal resultado del diseño es el Modelo de Diseño cuyo propósito es modelar el registro cumpliendo con las restricciones de los requerimientos. La aplicación de patrones en el diseño permite agilizar y hacer más eficiente el proceso de codificación. Se pudo definir sobre la base de la modelación correspondiente a este capítulo, la estructura que debe seguir la implementación. Se realizó la distribución física del registro a través de un modelo de despliegue, que describe todas las configuraciones sobre las cuales debe implementarse el mismo. Los diagramas de diseño y despliegue generados se consideran la entrada principal para el Flujo de Trabajo de Implementación. Además se explicó la forma de tratar los errores dentro del registro y la seguridad del mismo.

CAPÍTULO 3: IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se abordan las características de la implementación del registro. Se realiza la representación de los diagramas de componentes del registro. Además se brinda una descripción de los principales métodos implementados así como se muestran imágenes de la interfaz del registro.

3.1 Modelo de implementación.

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. (19)

3.1.1 Diagramas de componentes.

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes, etc.) y muestra las dependencias entre estos componentes. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. (22)

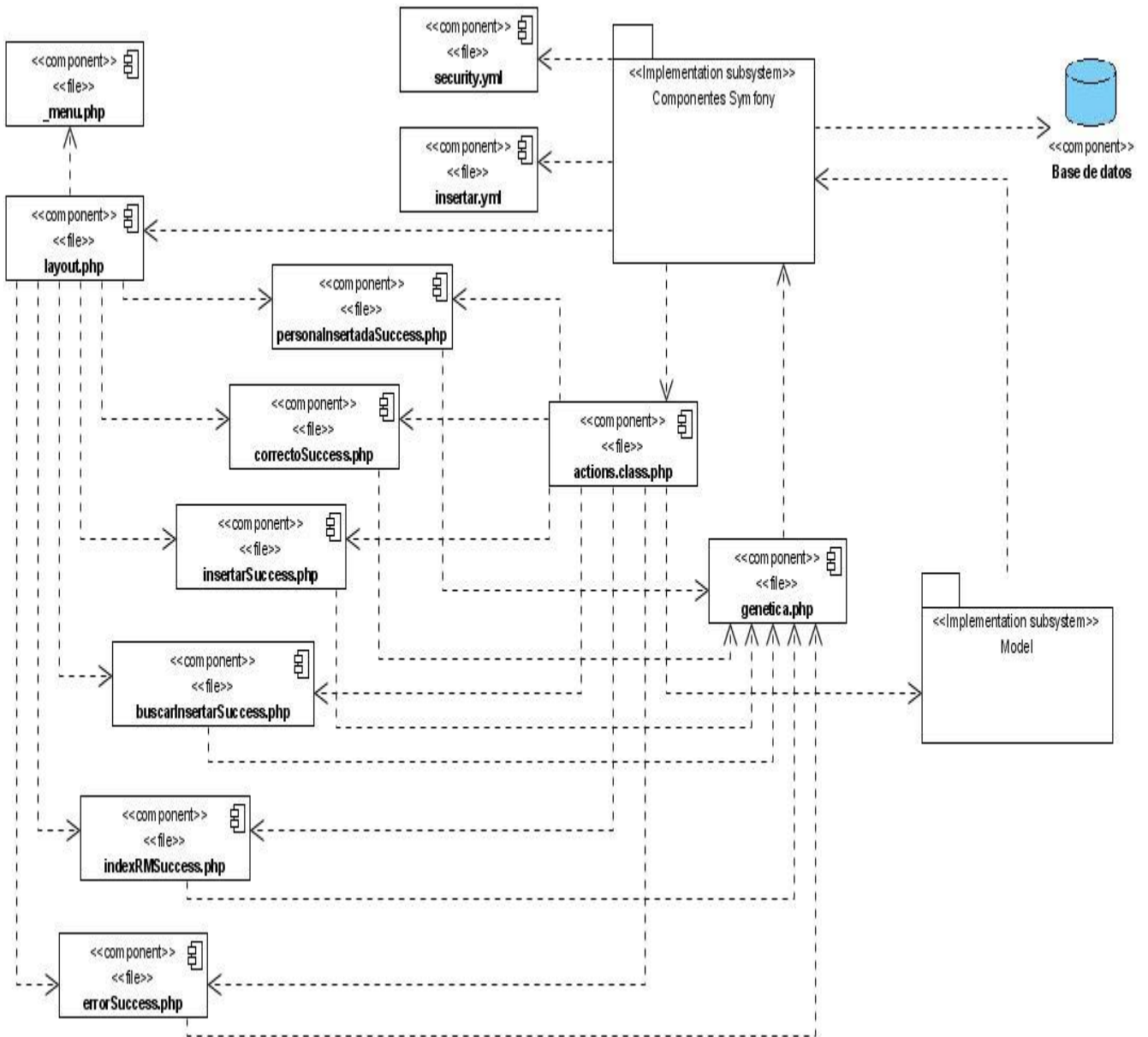


Figura 17. Diagrama de componentes caso de uso Insertar datos complementarios del paciente

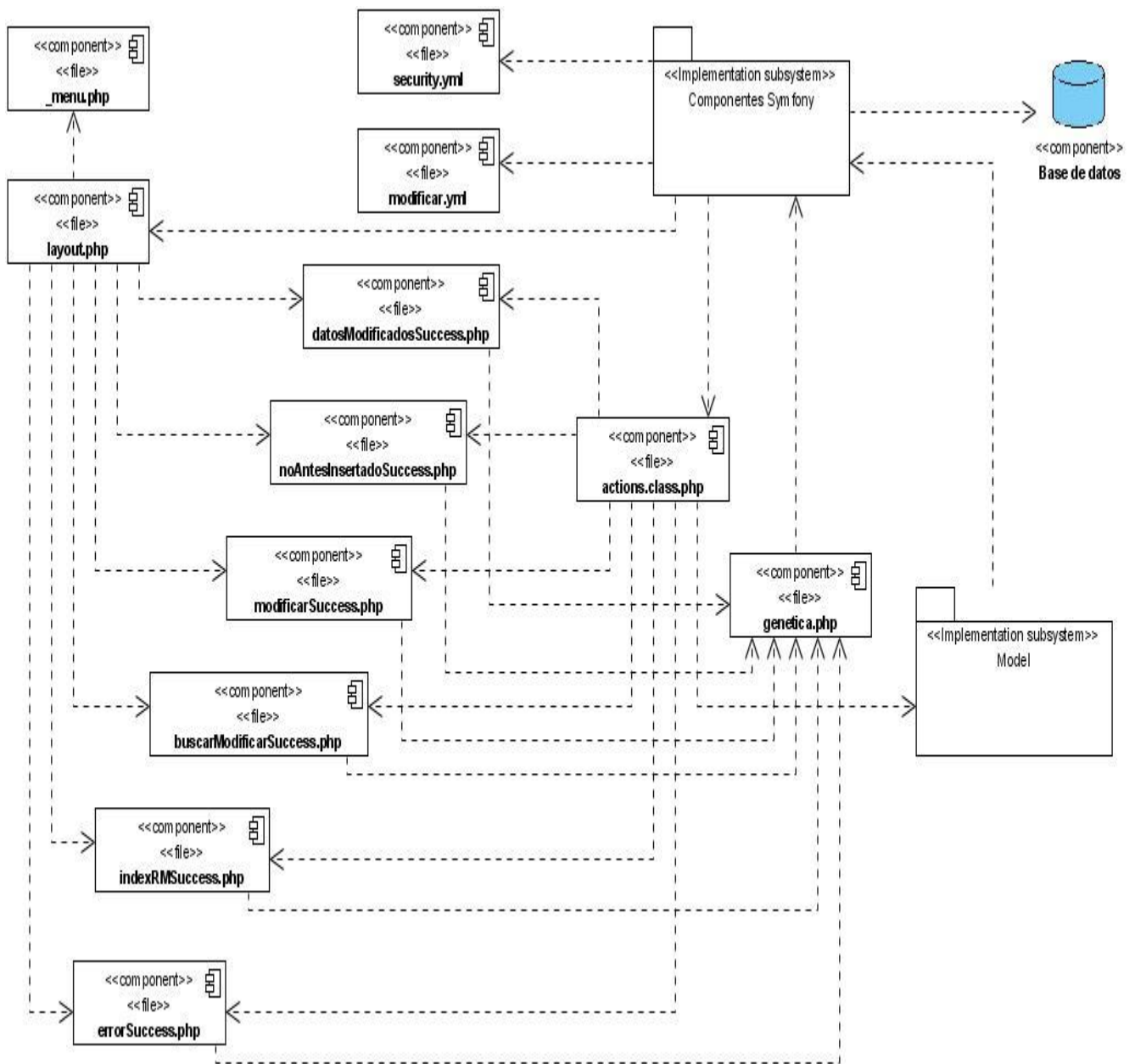


Figura 18. Diagrama de componentes caso de uso Modificar datos complementarios del paciente

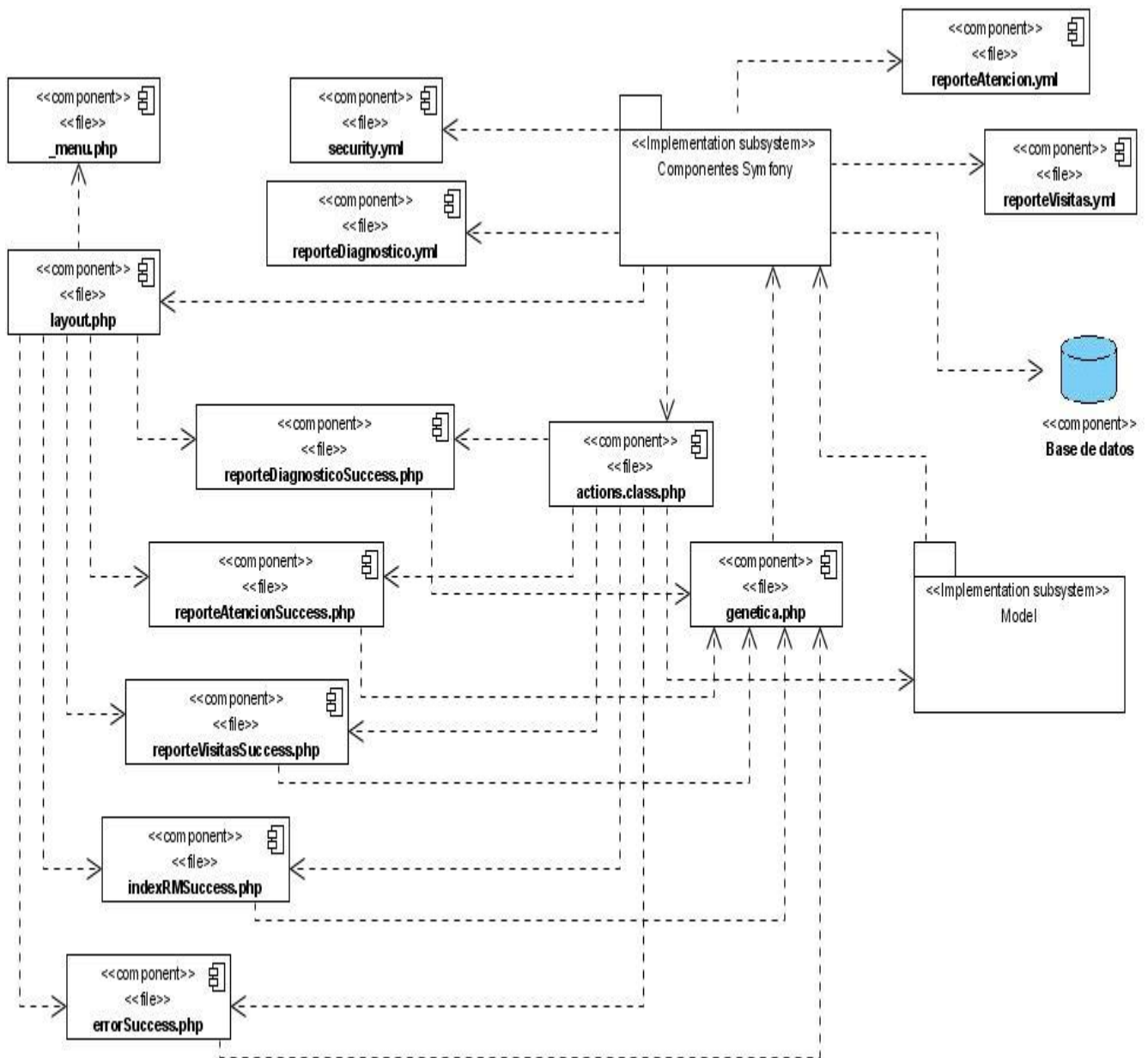


Figura 19. Diagrama de componentes caso de uso Generar reportes

3.2 Estilo de codificación.

A continuación se presenta el estilo de codificación definido por el Sistema Informático de Genética Médica:

1. Todas las etiquetas php deben ser completas (<?php ?>)... no reducidas (<? ?>).
2. Todas las variables deberían ser inicializadas o, al menos, comprobada su existencia utilizando isset() antes de ser utilizadas.

3. **El sangrado** del texto debe ser siempre de 4 espacios. No utilices los tabulador (todos los editores no interpretan el tab de la misma manera).

4. **Los nombres de las variables y funciones** tienen que ser siempre fáciles de leer, procurando que sean palabras en minúsculas con significado claro. Si realmente necesita más de una palabra, póngalas juntas, poniendo la inicial de cada palabra en mayúscula siempre que no sea la primera, pero procure mantenerlas tan breves como sea posible. Utilice nombres en plural para arreglos o matrices de objetos.

5. **Los bloques de código** siempre deben estar encerrados por llaves (incluso si solo constan de una línea) y correctamente identados (a 4 espacios).

6. **Las cadenas** tienen que ser definidas utilizando comillas simples siempre que sea posible, para obtener un mejor rendimiento.

7. **Todas** las funciones y clases deben estar comentariadas. **Los comentarios** deben ser añadidos de forma que resulten prácticos, para explicar el flujo del código y el propósito de las funciones o variables.

En los comentarios de las funciones debe aparecer el autor de la función, el objetivo de la misma, y una descripción de cada uno de los parámetros que se le pasen. (21)

3.3 Código fuente de los principales métodos y su descripción

Seguidamente se muestra un fragmento de código del método Insertar y se brinda una breve descripción del mismo.

```
//permite insertar los datos complementarios de un paciente
public function executeInsertar()
{
    if($this->getRequest()->getMethod() != sfRequest::POST)
    {

        $ins = new Criterias();
        $ins->add(RmDatosClinicoPeer::ID_PERSONA, $this->getRequestParameter('id'));
        $this->aux = RmDatosClinicoPeer::doCount($ins);
        if($this->aux > '0')
        {
            $this->redirect('rm/personaInsertada');
        }
        $this->MostrarDatosPrimarios();
    }
    else
    {

        //Edad de la madre en el periodo del embarazo (en años)
        $edad = new DgHistEmb();
        $persona = new DgPersona();
        $madreEmb= new RmMadreEmb();
        //movimiento fetal
        if($this->getRequestParameter('mov') == 'fuertes')
        {
            $edad->setIdMov(1);
        }
        $edad->save();

        //Cuando todos los datos se inserten correctamente
        $this->redirect('rm/correcto');
    }
}
}
```

Figura 20. Fragmento de código del método Insertar

El método Insertar permite la inserción de los datos complementarios del paciente que padezca de retraso mental. Se realiza una consulta para verificar que al paciente no se le han insertado los datos complementarios con anterioridad, en caso que el paciente haya sido insertado se redirecciona a la página personaInsertada, donde se muestra un mensaje que le informa al usuario que el paciente ya fue insertado con anterioridad. Si al paciente no se le han insertado los datos complementarios se hace una llamada al método MostrarDatosPrimarios el cual carga los datos primarios del paciente, luego se

procede a la inserción de los datos complementarios. Cuando todos los datos se han insertado correctamente se muestra un mensaje que le informa al usuario que los datos complementarios del paciente fueron insertados correctamente.

Otro de los métodos principales es el método Modificar del cual seguidamente se muestra un fragmento de código y se brinda una breve descripción del mismo.

```
public function executeModificar()
{
    if($this->getRequest()->getMethod() != sfRequest::POST)
    {
        $this->persona = DgPersonaPeer::retrieveByPK($this->getRequestParameter('id'));
        $count=0;
        $re=new Criteria();
        $re->add(DgPersonaNmodulosGeneticaPeer::ID_PERSONA,$this->persona->getIdPersona());
        $re->add(DgPersonaNmodulosGeneticaPeer::ID_MODULOS_GENETICA,5);
        $count=DgPersonaNmodulosGeneticaPeer::doCount($re);
        if($count==0)
        {
            $this->redirect('rm/noAntesInsertado');
        }
        //muestro los datos

        $this->MostrarDatosPrimarios();
        $this->CargarDatosComplementarios();

        return sfView::SUCCESS;
    }
    else
    {
        //modificando
        //seccion Datos generales del Instrumento Clínico-Genético
        $person = RmDatosClinicoPeer::retrieveByPK($this->getRequestParameter('id_per'));
        if($person)
        {
            $person->setNombrePersDatos($this->getRequestParameter('nombpers'));
            $person->setCmf($this->getRequestParameter('cmf'));
            $person->setTieneDiagnCausaRm($this->getRequestParameter('diagnos'));
            $person->setEvidenciasDismorfica($this->getRequestParameter('alcohol'));
            $person->save();
        }
        else
        {
            $personn = new RmDatosClinico();
            $personn->setNombrePersDatos($this->getRequestParameter('nombpers'));
            $personn->setCmf($this->getRequestParameter('cmf'));
            $personn->setTieneDiagnCausaRm($this->getRequestParameter('diagnos'));
            $person->setEvidenciasDismorfica($this->getRequestParameter('alcohol'));
            $personn->save();
        }

        //si se modifica correctamente
        return $this->redirect('rm/datosModificados');
    }
}
```

Figura 21. Fragmento de código del método Modificar

El método Modificar permite la modificación de los datos complementarios del paciente que padezca de retraso mental. Se realiza una consulta para verificar que al paciente se le han insertado los datos complementarios con anterioridad. En caso que el paciente no haya sido insertado se redirecciona a la página noAntesInsertado, donde se muestra un mensaje que le informa al usuario que el paciente no ha sido insertado como retrasado mental. Si los datos complementarios del paciente fueron insertados anteriormente se procede a hacer una llamada al método MostrarDatosPrimarios el cual carga los datos primarios del paciente y se le hace otra llamada al método CargarDatosComplementarios que es el encargado de cargar los datos complementarios del paciente, posteriormente se procede a la modificación de los datos complementarios del paciente. Al modificar correctamente todos los datos se muestra un mensaje que le informa al usuario que los datos complementarios fueron modificados correctamente.

3.4 Interfaz del registro.

Se elaboró una interfaz tratando de mantener de la mejor forma posible, el mismo orden de los elementos que aparecen en el Instrumento Clínico-Genético (Ver Anexo 1), contribuyendo a que sea más fácil la interacción del usuario con el registro.

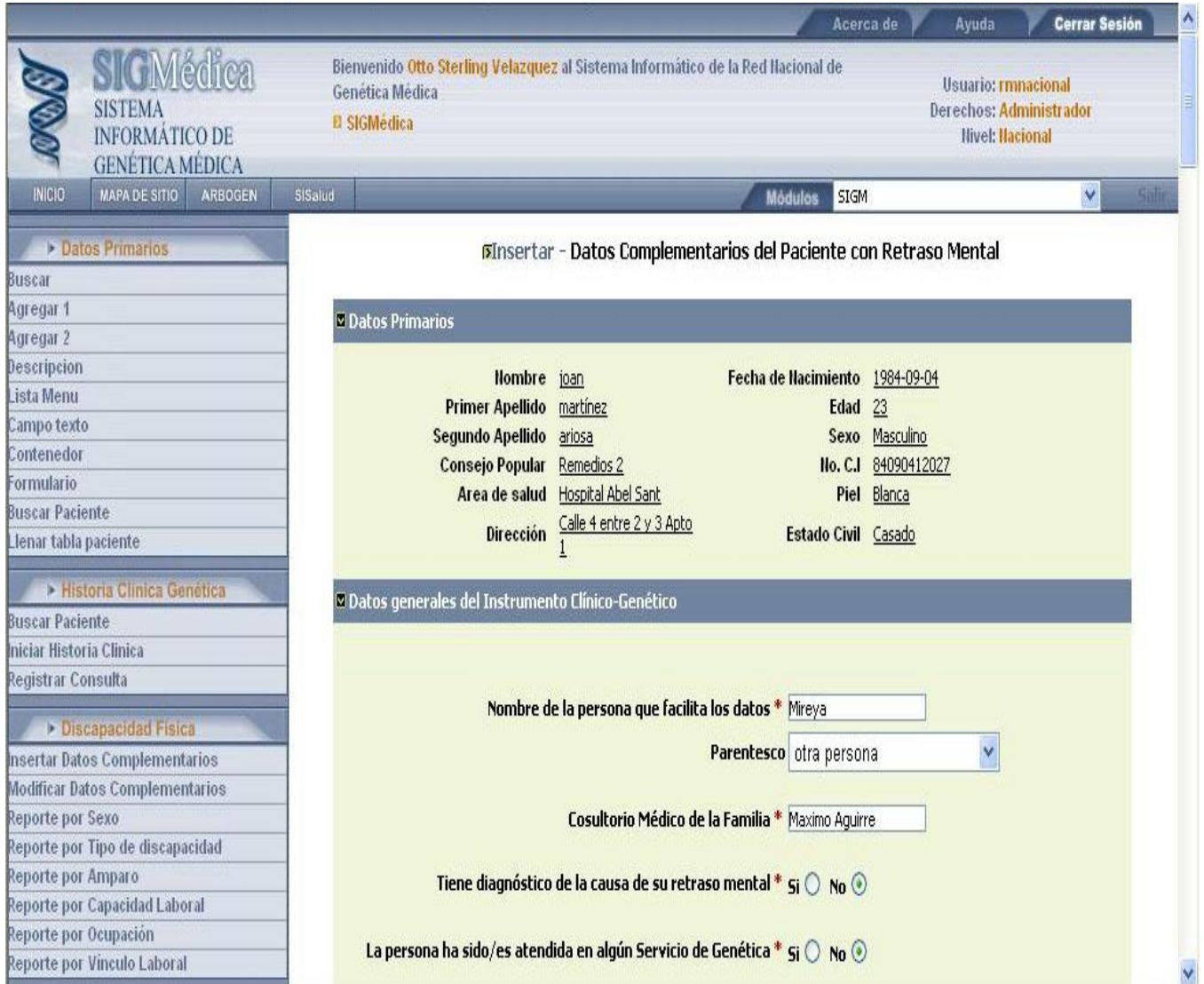


Figura 22. Interfaz Insertar Datos Complementarios

SIGMédica
SISTEMA INFORMÁTICO DE GENÉTICA MÉDICA

Bienvenido **Otto Sterling Velazquez** al Sistema Informático de la Red Nacional de Genética Médica
SIGMédica

Usuario: **rmnacional**
 Derechos: **Administrador**
 Nivel: **Nacional**

INICIO MAPA DE SITIO ARBOGEN SISalud Módulos SIGM

Modificar - Datos Complementarios del Paciente con Retraso Mental

Datos Primarios

Nombre	<u>joan</u>	Fecha de Ilacimiento	<u>1984-09-04</u>
Primer Apellido	<u>martínez</u>	Edad	<u>23</u>
Segundo Apellido	<u>ariosa</u>	Sexo	<u>Masculino</u>
Consejo Popular	<u>Remedios 2</u>	No. C.I	<u>84090412027</u>
Area de salud	<u>Hospital Abel Sant</u>	Piel	<u>Blanca</u>
Dirección	<u>Calle 4 entre 2 y 3 Apto 1</u>	Estado Civil	<u>Casado</u>

Datos generales del Instrumento Clínico-Genético

Nombre de la persona que facilita los datos *

Parentesco *

Cosultorio Médico de la Familia *

Tiene diagnóstico de la causa de su retraso mental * Si No

La persona ha sido/es atendida en algún Servicio de Genética * Si No

Figura 23. Interfaz Modificar Datos Complementarios

SIGMédica
SISTEMA INFORMÁTICO DE GENÉTICA MÉDICA

Bienvenido **nacional** al Sistema Informático de la Red Nacional de Genética Médica
 SIGMédica

Usuario: **rmnacional**
 Derechos: **Administrador**
 Nivel: **Nacional**

INICIO MAPA DE SITIO ARBOGEN SISalud Módulos SIGM

Reporte - Cantidad de personas según el diagnóstico de retraso mental

Reporte según el diagnóstico de retraso mental

		Cantidad de personas:
Provincia*	Matanzas	11
Municipio	Limonar	8
Consejo Popular	Santa Ana	5
Diagnóstico retraso mental*		
	Si <input checked="" type="radio"/> No <input type="radio"/>	

Buscar Cancelar

Figura 24. Interfaz Reporte según el diagnóstico de retraso mental

The screenshot shows the SIGMédica web application interface. At the top, there is a header with the logo and the text 'SISTEMA INFORMÁTICO DE GENÉTICA MÉDICA'. The user is logged in as 'nacional' with 'Administrador' rights and 'Nacional' level. The main menu on the left includes options like 'Datos Primarios', 'Historia Clínica Genética', 'Discapacidad Física', 'Enfermedades Genéticas', 'Gemelo', and 'Malformaciones Congénitas'. The main content area is titled 'Reporte - Cantidad de personas según la atención en algún servicio de genética'. It features a table with the following data:

		Cantidad de personas:
Provincia*	La Habana	30
Municipio	Jaruco	11
Consejo Popular	Bainoa - Casiguas	6

Below the table are 'Buscar' and 'Cancelar' buttons. The interface also includes a search bar and a 'Seguir' button in the top right corner.

Figura 25. Interfaz Reporte según la atención en algún servicio de genética

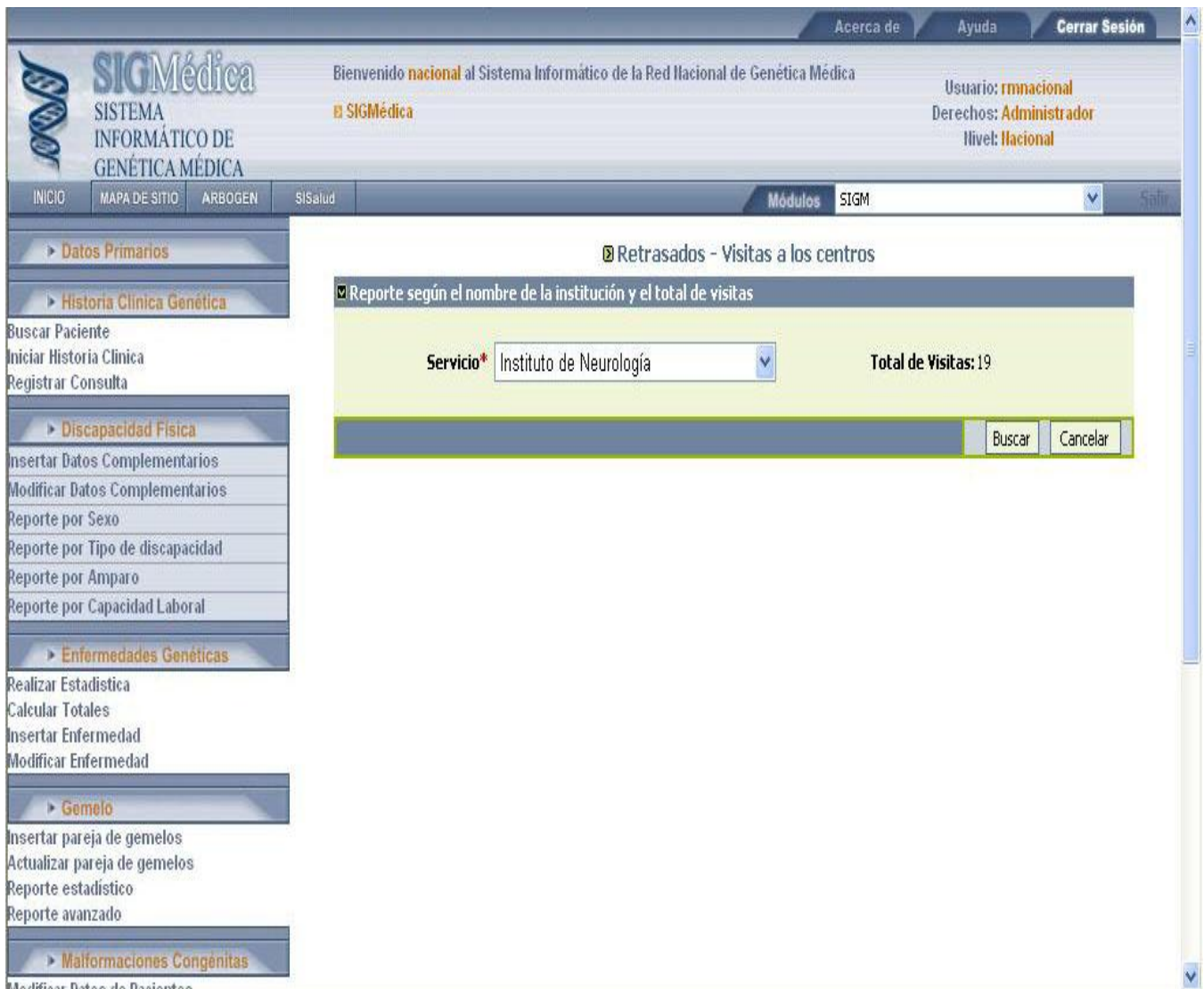


Figura 26. Interfaz Reporte según el nombre de la institución y el total de visitas

3.5 Validación a nivel de desarrollador.

Con el objetivo de lograr mayor calidad y eficiencia el desarrollador debe validar el registro sobre la marcha. Cada implementador tiene su propio estilo para validar sus aplicaciones y comprobar que el camino por el que se conduce lo llevará a una solución satisfactoria.

El lenguaje de programación utilizado y el framework de desarrollo no cuentan con robustos compiladores en tiempo de diseño que permitan la rápida detección y mitigación de los errores, por lo que es necesario que el implementador aplique sus conocimientos y experiencias para detectar estos errores de la forma más rápida.

Para verificar el tipo y el valor de una variable, el implementador puede auxiliarse de la función `var_dump()` de PHP que muestra información en el navegador. Dicha función se muestra a continuación.

```
$cr1 = new Criterio();
$cr1->add(RmServGeneticaRmDatosClinicoPeer::ID_DATOS_CLINICO, $dc_per->getIdDatosClinico());
$cr1->add(RmServGeneticaRmDatosClinicoPeer::ID_SERV_GENETICA, 1);
$this->servs1 = RmServGeneticaRmDatosClinicoPeer::doSelect($cr1);

var_dump($this->servs1);
```

Figura 27. Fragmento de código donde se muestra la función `var_dump()`

La introducción de esta función puede producir efectos no deseados en la página como muestra la imagen inmediata inferior, pero el implementador pudo verificar que la variable contiene el valor que se deseaba.

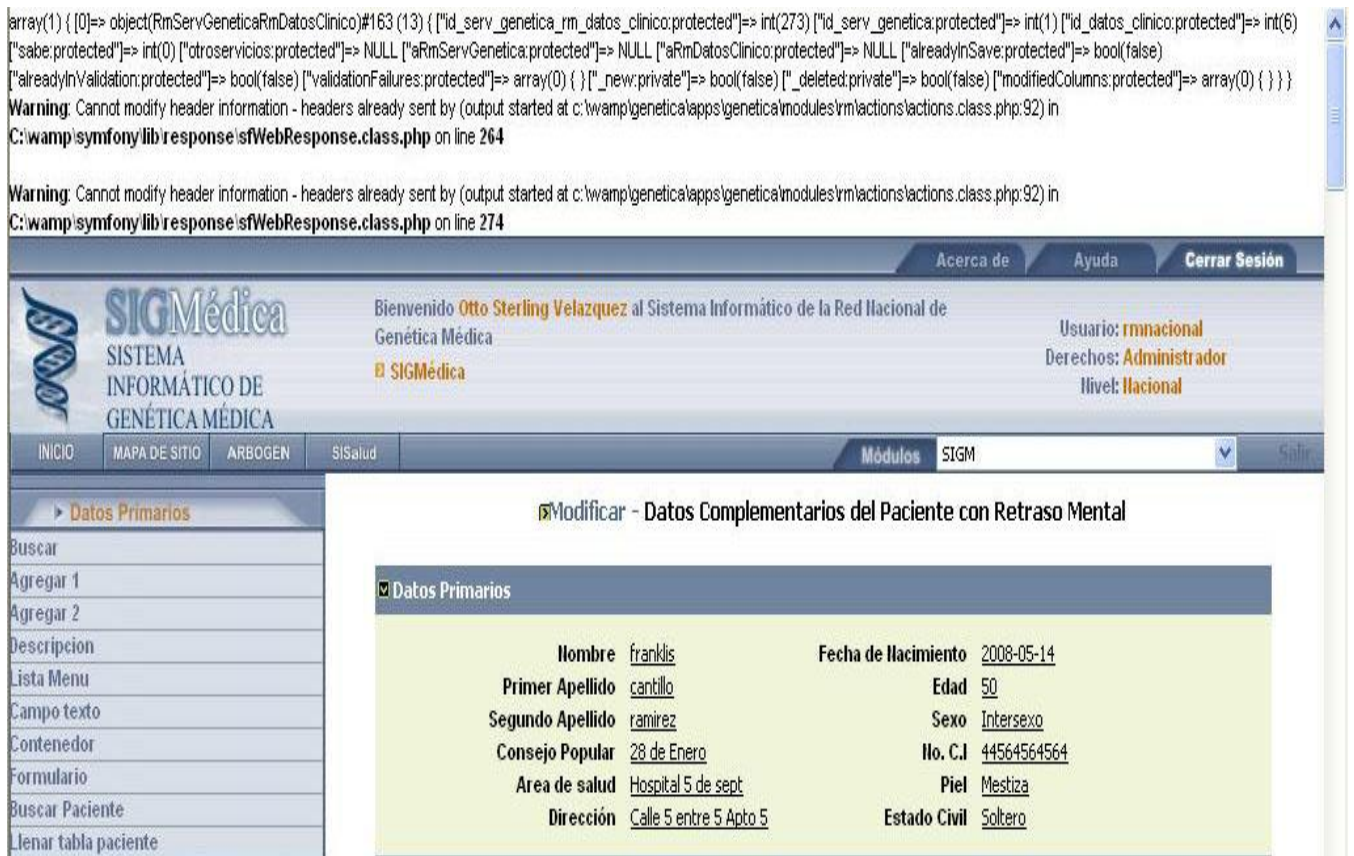


Figura 28. Interfaz con los efectos no deseados

Dejar de introducir datos de entrada obligatoria o introducir datos de forma incorrecta es una manera de verificar la consistencia del registro que se está desarrollando. Se muestra un ejemplo de los mensajes que aparecen en el registro cuando no se introducen datos que son de entrada obligatoria:

☑ Datos generales del Instrumento Clínico-Genético

Nombre de la persona que facilita los datos * **↓ Entre el nombre de la persona ↓**

Parentesco * **↓ Debe seleccionar un tipo de parentesco ↓**

Cosultorio Médico de la Familia * **↓ Entre el nombre del Consultorio ↓**

Tiene diagnóstico de la causa de su retraso mental * Si No

La persona ha sido/es atendida en algún Servicio de Genética * Si No

Figura 29. Interfaz con mensajes asociados a las entradas de datos obligatorias

3.6 Conclusiones del capítulo.

En este capítulo se representaron los diagramas de componentes de cada uno de los casos de usos, los cuales están basados en las definiciones realizadas durante el diseño. Además se brindó una breve descripción de los principales métodos implementados para obtener un mejor entendimiento de los mismos. Se mostraron algunas de las interfaces del registro y se expusieron algunos ejemplos de las validaciones realizadas a nivel de desarrollador, las cuales permitieron detectar y solucionar los errores identificados.

CONCLUSIONES

El estudio de la investigación precedente y de los sistemas existentes para la gestión de la información de las personas que padecen de retraso mental, facilitó identificar los aspectos positivos a tener en cuenta en el presente trabajo y las principales deficiencias que debían ser corregidas.

Se realizó el diseño de todas las clases del registro mediante la utilización patrones de diseño lo que proporcionó una correcta implementación del mismo.

El producto funcional obtenido a partir de la implementación de las clases del diseño permitió integrar el RECURM al SIGM, dando solución al problema científico.

El Registro Cubano de Retraso Mental versión 2.0 constituye un importante aporte al desarrollo de la genética en Cuba ya que le permite a los genetistas gestionar las informaciones de los pacientes que presentan retraso mental desde cualquier lugar del país.

RECOMENDACIONES

Al término de esta investigación se recomienda:

- El enriquecimiento del registro incluyendo la funcionalidad de emitir una clasificación etiopatogénica de los pacientes interpretando determinadas características.

REFERENCIAS BIBLIOGRÁFICAS

1. Fundación para la Investigación del Hospital Universitario La Fe. [En línea] 12 de enero de 2006 [Citado el: 5 de diciembre de 2007.] <http://www.fundacionlafe.org/prensa/notasPrensa.htm>.
2. **Rigó Portillo, Daulemys.** *Sistema Automatizado de Registro Cubano de Retraso Mental.* Facultad de Bioinformática, Universidad de Ciencias Informáticas. Ciudad Habana, 2007.
3. ASOCIACIÓN GIRMOGEN. [En línea] [Citado el: 6 de diciembre de 2007.] <http://girmogen.org/esp/invesesp.html>.
4. ASOCIACIÓN GIRMOGEN. [En línea] Septiembre de 2007 [Citado el: 6 de diciembre de 2007.] <http://girmogen.org/esp/noticias2007esp.html>.
5. Universidad de Salamanca. [En línea] 1998 [Citado el: 7 de diciembre de 2007.] <http://www.usal.es/%7Einico/investigacion/jornadas/jornada3/actas/simp2.pdf>.
6. Centro de Genética Médica. [En línea] 1999 [Citado el: 7 de diciembre de 2007.] <http://www.sld.cu/sitios/genetica/>.
7. PHP. [En línea] 2001 [Citado el: 8 de diciembre de 2007.] <http://www.php.net/>.
8. **Sánchez Perodín, Yusdenis.** *Línea base de la arquitectura del Sistema de Información de Genética Médica.* Facultad de Bioinformática, Universidad de Ciencias Informáticas. Ciudad Habana, 2008.
9. MySQL The world's most popular open source database. [En línea] 1995 [Citado el: 8 de diciembre de 2007.] http://www.mysql.com/news-and-events/news/article_976.html.
10. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* s.l.: McGraw-Hill/Interamericana, 2002.
11. **Oktaba, Hanna.** Facultad de Ciencias, UNAM. Introducción a patrones. [En línea] [Citado el: 11 de diciembre de 2007.] <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>
12. **Gracia, Joaquin.** Patrones de diseño. [En línea] 27 de Mayo de 2005 [Citado el: 22 de enero de 2008.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
13. Conferencia 6 Ingeniería de software 1. Flujo de trabajo Análisis y Diseño. Modelo de diseño. Universidad de las Ciencias Informáticas. Curso 2005-2006.
14. El mundo informático. [En línea] 8 de mayo de 2007 [Citado el: 22 de enero de 2008.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>
15. **Potencier, Fabien y Zaninotto, François.** *Symfony, la guía definitiva.* 2008.
16. **Sánchez Perodín, Yusdenis.** *Pautas de diseño del Sistema de Información de Genética Médica.* Facultad de Bioinformática, Universidad de Ciencias Informáticas. Ciudad Habana, 2008.

17. Conferencia 3 Ingeniería de software 1. Fase de Inicio. Flujo de trabajo Requerimientos. Universidad de las Ciencias Informáticas. Curso 2005-2006.
18. **Larman, Craig.** *Uml y patrones, introducción al análisis y diseño orientado a objetos.* México : Prentice Hall, 1999.
19. Ministerio del Poder Popular para las Telecomunicaciones y la Informática. *Modelo de Implementación.* [En línea] 2008 [Citado el: 15 de febrero de 2008.]
http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297
20. Quanta Plus. [En línea] 2005 [Citado el: 9 de diciembre de 2007.] <http://quanta.kdewebdev.org/>
21. **Sánchez Perodín, Yusdenis.** *Estilo de código del Sistema de Información de Genética Médica.* Facultad de Bioinformática, Universidad de Ciencias Informáticas. Ciudad Habana, 2008.
22. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : PEARSON EDUCACIÓN S.A, 2000.

BIBLIOGRAFÍA

1. Fundación para la Investigación del Hospital Universitario La Fe. [En línea] 12 de enero de 2006 [Citado el: 5 de diciembre de 2007.] <http://www.fundacionlafe.org/prensa/notasPrensa.htm>.
2. **Rigó Portillo, Daulemys.** *Sistema Automatizado de Registro Cubano de Retraso Mental.* Facultad de Bioinformática, Universidad de Ciencias Informáticas. Ciudad Habana, 2007.
3. ASOCIACIÓN GIRMOGEN. [En línea] [Citado el: 6 de diciembre de 2007.] <http://girmogen.org/esp/invesesp.html>.
4. ASOCIACIÓN GIRMOGEN. [En línea] Septiembre de 2007 [Citado el: 6 de diciembre de 2007.] <http://girmogen.org/esp/noticias2007esp.html>.
5. Universidad de Salamanca. [En línea] 1998 [Citado el: 7 de diciembre de 2007.] <http://www.usal.es/%7Einico/investigacion/jornadas/jornada3/actas/simp2.pdf>.
6. Centro de Genética Médica. [En línea] 1999 [Citado el: 7 de diciembre de 2007.] <http://www.sld.cu/sitios/genetica/>.
7. PHP. [En línea] 2001 [Citado el: 8 de diciembre de 2007.] <http://www.php.net/>.
8. **Sánchez Perodín, Yusdenis.** *Línea base de la arquitectura del Sistema de Información de Genética Médica.* Facultad de Bioinformática, Universidad de Ciencias Informáticas. Ciudad Habana, 2008.
9. MySQL The world's most popular open source database. [En línea] 1995 [Citado el: 8 de diciembre de 2007.] http://www.mysql.com/news-and-events/news/article_976.html.
10. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* s.l.: McGraw-Hill/Interamericana, 2002.
11. **Oktaba, Hanna.** Facultad de Ciencias, UNAM. Introducción a patrones. [En línea] [Citado el: 11 de diciembre de 2007.] <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>
12. **Gracia, Joaquin.** Patrones de diseño. [En línea] 27 de Mayo de 2005 [Citado el: 22 de enero de 2008.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
13. Conferencia 6 Ingeniería de software 1. Flujo de trabajo Análisis y Diseño. Modelo de diseño. Universidad de las Ciencias Informáticas. Curso 2005-2006.
14. El mundo informático. [En línea] 8 de mayo de 2007 [Citado el: 22 de enero de 2008.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>
15. **Potencier, Fabien y Zaninotto, François.** *Symfony, la guía definitiva.* 2008.
16. **Sánchez Perodín, Yusdenis.** *Pautas de diseño del Sistema de Información de Genética Médica.* Facultad de Bioinformática, Universidad de Ciencias Informáticas. Ciudad Habana, 2008.

17. Conferencia 3 Ingeniería de software 1. Fase de Inicio. Flujo de trabajo Requerimientos. Universidad de las Ciencias Informáticas. Curso 2005-2006.
18. **Larman, Craig.** *Uml y patrones, introducción al análisis y diseño orientado a objetos.* México : Prentice Hall, 1999.
19. Ministerio del Poder Popular para las Telecomunicaciones y la Informática. *Modelo de Implementación.* [En línea] 2008 [Citado el: 15 de febrero de 2008.]
http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297
20. Quanta Plus. [En línea] 2005 [Citado el: 9 de diciembre de 2007.] <http://quanta.kdewebdev.org/>
21. **Sánchez Perodín, Yusdenis.** *Estilo de código del Sistema de Información de Genética Médica.* Facultad de Bioinformática, Universidad de Ciencias Informáticas. Ciudad Habana, 2008.
22. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : PEARSON EDUCACIÓN S.A, 2000.
23. Ayuda extendida del Rational Rose Enterprise Edition 2003.

ANEXOS

Anexo 1 Instrumento Clínico-Genético

**INVESTIGACION SOBRE PERSONAS CON RETRASO MENTAL
 INSTRUMENTO DE CLASIFICACION INICIAL
 CARACTERIZACION ETIOLOGICA DEL RETRASO MENTAL
 ESTUDIO CLINICO-GENETICO DE LAS PERSONAS CON RETRASO MENTAL**

Municipio: _____

Folio: _____

I. DATOS GENERALES

- 1.- Nombre de la persona que facilita los datos _____
- 2.- Parentesco de la persona entrevistada con el propósito _____
- 3.- Tiene diagnóstico de la causa de su Retraso Mental 1.Si___ 2.No___
- 4.- La persona ha sido/es atendida en algún Servicio de Genética 1.Si___ 2.No___ 3.No sabe___
- 5.- Si respondió Si en la pregunta anterior, especifique en qué Centro ha sido o es atendido:
 1.Centro Nacional de Genética___ 2.Hospital William Soler___ 3.Hospital Juan M Márquez___
 4. Instituto de Neurología _____ 5.Servicio Prenatal González Coro___
 6.Genética de su municipio _____ 7.Genética de su provincia___ 8.Otro ___ 9.No sabe___
 De responder 8.Otro, especifiqueCuál? _____

II. DATOS SOBRE LA MADRE EN EL PERIODO DEL EMBARAZO (EVENTOS AMBIENTALES)

- 6.- Trabajaba 1.Si___ 2.No___ 3.No sabe___
- 7.- Si trabajaba especifique el tipo de trabajo _____
8. Edad de la madre (en el período del embarazo) 1.Menor 20 años___ 2.20-35 años ___
 3.36-38años_____ 4.mayor 40 años___
9. Enfermedades infecciosas de la madre durante el embarazo (De responder Si debe precisarse el (los) trimestre (s) del embarazo en que se produjo la enfermedad). 1.Si___ 2.No___

Enfermedad infecciosa	NO	SI	No Sabe	TRIMESTRES			
				PRIMERO	SEGUNDO	TERCERO	No sabe
3.Erupciones							
4.Rubéola							
5.Otras TORCH*							
6.Dengue							
7.Varicela							

8.Sífilis							
-----------	--	--	--	--	--	--	--

***TORCH (Toxoplasmosis, citomegalovirus, herpes simple)**

10.- Ingestión de medicamentos por la madre durante el embarazo. No incluir tratamiento prenatal cubano establecido por el PAMI.

(Si la respuesta el SI especificar I nombre del medicamento, en el trimestre que lo consumió y el tiempo de ingestión). 1.Si ___ 2.No ___ 3.No Sabe ___

Medicamentos	TRIMESTRES				DIAS DE CONSUMO
	PRIMERO	SEGUNDO	TERCERO	NO SABE	
1.					
2.					
3.					

Hábitos tóxicos durante el embarazo

	1.NO	2.SI	3.No Sabe	TRIMESTRES			
				4. 1ero.	5. 2do.	6. 3ero.	7.No sabe
11.Bebidas alcohólicas							
12.Cigarro/Tabaco							
13.Otras Drogas*							

* Si refiere consumo de Narcóticos u otras drogas, especifique cuáles_____

14.- Exposición a calor intenso 1.Si___ 2.No___ 3.No sabe___

Si respondió Si, especifique tipo de calor 4.Sauna___ 5.Laboral___

6. Fiebre alta y mantenida___

15.- Recibió Radiaciones POR TRATAMIENTO (CANCER) o investigaciones médicas con sustancias radioactivas 1.Si___ 2.No___ 3.No sabe___

Si la respuesta el Si especifique trimestre 1.Primeros___ 2.Segundo___ 3.Tercero___

4. No sabe___

Enfermedades Maternas durante el embarazo

Enfermedad	1.NO	2.SI	3.No Sabe	TRIMESTRES			
				4.PRIMERO	5.SEGUNDO	6.TERCERO	7.No sabe
16.Hipertensión							
17.Diabetes							
18.Tiroidismos							

19.Epilepsia							
20.Fenilcetonuria							
21.Sickleemia							

(EVENTOS PERINATALES)

Otros eventos durante el embarazo	1.N O	2.SI	3.No Sabe	TRIMESTRES			
				4.PRIMERO	5.SEGUNDO	6.TERCERO	7.No sabe
22.Sangramiento							
23.Amenaza de aborto o parto pretérmino							
24.CIUR							
25.Oligo/ Polihidramnios							

26.- Movimientos fetales 1.Fuertes___ 2.Débiles___ 3.No sabe___

II. DATOS DEL PARTO Y DE LOS PRIMEROS 7 DIAS DE VIDA DEL NIÑO

27.- Edad gestacional al parto 1.Pretérmino___ 2.A término___

28.- Lugar del parto 1.Extrainstitucional ___ 2.Institucional___ 3.Dónde?_____

4.Si parto en institución especifique días de nacido al alta _____

29.- Tipo de nacimiento 1.Normal ___ 2.Instrumentado___ 3.Cesárea___

Del recién nacido

30. Peso 1.Menor 2500g ___ 2.Mayor 2500g ___ 3.Macrofetos___

31. Talla___cms

32. Circ.cefálica___cms

33.- Color de la piel 1.Cianótico___ 2.Normal___ 3.Ictero intenso___ 4.No sabe___

34.- Llanto 1.Fuerte ___ 2.Débil ___ 3.Demorado___ 4.no sabe___

35.- Apgar al nacer 1.Menos de 5___ 2.De 6 a 10___ 1.No sabe___

36.- Apgar a los 5 minutos 1.Menos de 5___ 2.De 6 a 10___ 1.No sabe___

Complicaciones del recién nacido

	1.SI	2.NO	3.NO SABE
--	-------------	-------------	------------------

37. Hemorragias			
38. Convulsiones			
39. Traumas perinatales			
40. Kerníctero			
41. Infecciones			
42. Incompatibilidad de grupos sanguíneos			

IV. DATOS RELACIONADOS CON EL NIÑO A PARTIR DEL 8vo. DIA DE VIDA

EVENTOS POSTNATALES

	1.SI	2.EDAD	3.NO	4.NO SABE
43. Infecciones del SNC (Meningitis, Meningoencefalitis)				
44. Traumas o accidentes				
45. Intoxicaciones con metales pesados (plomo, mercurio), medicamentos u otras				
46. Malnutrición severa				
47. Otras infecciones severas en el primer año de vida				

V. INTERROGATORIO Y EXAMEN FISICO AL PROPÓSITO

(EVENTOS PRENATALES INESPECIFICOS Y GENETICOS)

48.- Otras Discapacidades en el propósito

	1.SI	2.EDAD	3.NO	4.NO SABE
1. Ciego				
2. Débil visual				
3. Sordo				
4. Hipoacusico				
5. Defectos motores totales o parciales				
6. Epileptico				

49.- Antecedentes de Retraso Mental en familiares del propósito 1.Si_____ 2.No_____

Si respondió Si, especificar:

Grado de parentesco	1.Si	2.No	3.No sabe	4.Cantidad
Madre				
Padre				
Hermanos				

Hijos				
Abuelos maternos				
Tíos maternos				
Primos maternos (hasta 2da. generación)				
Abuelos paternos				
Tíos paternos				
Primos paternos (hasta 2da. generación)				

50.- Consanguinidad en los padres del propósito 1.Si_____ 2.No_____ 3.No sabe_____

EXAMEN FISICO

51.- Cráneo 1.Normal_____ 2.Microcefalia_____ 3.Macrocefalia_____ 4.Otras_____

52.- Signos dismórficos 1.Si_____ 2. No tiene_____

Si la respuesta es Si: 1.Cara y cuello _____ 2.Extremidades_____
3. Genitales _____ 4.Manos y pies_____ 5.Articulaciones_____

53.- Malformaciones congénitas externas (corregidas quirúrgicamente o no)

1.Si_____ 2.No tiene_____

Si la respuesta es Si: 1.Cara y cráneo _____ 2.Torax y abdomen_____
3.Raquis y extremidades_____ 4.Genitales _____

54.- Malformaciones congénitas internas referidas: 1.Si_____ 2.No sabe_____

Si la respuesta es Si: 1.Sistema Nervioso_____ 2.Corazon _____ 3.Digestivas_____
4. Renales _____ 5.Columna_____ 6.Otras _____

55.- Examen de la piel

	No	Si	Número	Pequeñas (pecas)	Medianas 0.5-1.5 cms	Grandes Más de 1.5 cms
1.Manchas café con leche						
2.Manchas hipopigmentadas						

3.Nevus (lunares)	No	Si	Pocos	mucho s	Grandes y pilosos	Pequeños múltiples	verrucosos
----------------------	----	----	-------	------------	----------------------	-----------------------	------------

4.Hemangiomas grandes (>2cm)	Hemangiomas pequeños (<2cm)	Cara/cráneo	Cuello	Tronco	Extremidades
------------------------------	-----------------------------	-------------	--------	--------	--------------

56.- PSICOSIS PRIMARIAS Y RETRASO MENTAL

1. Si____ 2.No____ 3.No sabe____

Si respondió Si, especifique si tiene diagnóstico de: 4.Autismo_____

5. Esquizofrenia_____

6. Otras Psicosis infantiles_____

PARA SER LLENADO POR EL EQUIPO MULTIDISCIPLINARIO DE CLASIFICACION ETIOLOGICA

57.- CLASIFICACION POR INSTRUMENTO	58.-CLASIFICACION POR VISITA DEL GENETISTA	59.- CLASIFICACION FINAL
1.Prenatal genético	1.Prenatal genético	1.Prenatal genético
2.Prenatal Inespecífico	2.Prenatal Inespecífico	2.Prenatal Inespecífico
3.Prenatal ambiental	3.Prenatal ambiental	3.Prenatal ambiental
4.Perinatal	4.Perinatal	4.Perinatal
5.Postnatal	5.Postnatal	5.Postnatal
6.Psicosis	6.Psicosis	6.Psicosis
7.Inclasificado	7.Inclasificable	7.Inclasificable
8. A visitar por Genetistas		

60.- Evidencias dismórficas de daño fetal por alcohol 1.Si____ 2.No____

61.- Si se sospecha etiología prenatal genética: 1.Monogénico____ 2.Cromosómico____
3. Multifactorial____ 4.Síndrome Down____

62.- Si se clasificó como Prenatal inespecífico 1.Dismorfias 2.Antecedentes familiares____
3. Malformaciones____ 4.CIUR____

63. Si Inclasificado.

1-Faltan datos necesarios para una clasificación etiológica____

2-Se deja como Inclasificable pero es susceptible de otros estudios____

	1.NO	2.SI	3.POSITIVO	4.NORMALES
64.ANALISIS REALIZADOS				
65.ESTUDIO CROMOSOMICO				

66.PRUEBAS METABOLICAS EN ORINA				
67.ESTUDIOS MOLECULARES PARA FRAGIL X				
68. OTROS ESTUDIOS MOLECULARES				
69.OTROS ANALISIS (TSH, T4, PKU, BIOTINA)				

Anexo 2 Descripción detallada de los casos de uso del sistema.

CU-1	Insertar datos complementarios del paciente		
Actores	Genetista Municipal / Unidad de salud (Inicia).		
Propósito	Insertar los datos complementarios del paciente.		
Resumen	El caso de uso se inicia cuando el Genetista Municipal o de Unidad de salud va a insertar los datos complementarios recogidos de un paciente. El registro le muestra la interfaz correspondiente para la búsqueda de un paciente y permite seleccionarlo, mostrando una nueva interfaz para insertar los datos complementarios del paciente. El caso de uso finaliza cuando se emite un mensaje informando que los datos fueron insertados correctamente.		
Referencias	RF1, Gestionar Datos Primarios (include).		
Precondiciones	El usuario se ha autenticado.		
Postcondiciones	Se insertan los datos del paciente.		
Curso Normal de los Eventos			
Acciones del Actor		Respuesta del Sistema	
1. El Genetista Municipal o de Unidad de salud quiere insertar los datos complementarios de un paciente.		2. El registro en dependencia del nivel de acceso del Genetista Municipal o de Unidad de salud, muestra una interfaz para buscar el paciente.	
3. El Genetista Municipal o de Unidad de salud selecciona el paciente.		4. El registro muestra la interfaz para insertar los datos complementarios del paciente.	
5. El Genetista Municipal o de Unidad de salud inserta los datos complementarios del paciente.		6. El registro muestra un mensaje especificando que los datos del paciente fueron insertados satisfactoriamente.	

Flujo Alterno	
Acción del Acto	Respuesta del Sistema
3. El Genetista Municipal o de Unidad de salud selecciona un paciente que ya se encuentra insertado.	4. El registro muestra mensaje “El paciente ya se encuentra insertado. Si desea modificar sus datos dirijase al vínculo correspondiente.”
5. El Genetista Municipal o de Unidad de salud introduce datos complementarios no válidos.	6. El registro muestra un mensaje de error por cada campo de datos no válidos.
5. El Genetista Municipal o de Unidad de salud no introduce todos los datos complementarios necesarios.	6. El registro muestra mensaje de error por cada campo de datos no insertados.
3. El Genetista Municipal o de Unidad de salud busca un paciente que no se encuentra insertado.	4. El registro muestra un mensaje “No existen resultados acordes a la búsqueda realizada”, seguido por el vínculo “Registrar” que redirecciona a la interfaz para la inserción de los datos primarios.
Prioridad: Crítico	

Tabla 1: Descripción del caso de uso del sistema Insertar datos complementarios del paciente

CU-2	Modificar datos complementarios del paciente
Actores	Genetista Nacional / Provincial (Inicia)
Propósito	Modificar los datos complementarios existentes en el registro de un determinado paciente.
Resumen	El caso de uso se inicia cuando el Genetista Nacional o Provincial va a modificar alguno de los datos complementarios existentes en el registro

	del paciente. El registro le muestra la interfaz correspondiente para la búsqueda de un paciente y permite seleccionarlo, mostrando una nueva interfaz para modificar los datos complementarios del paciente. El caso de uso finaliza cuando se emite un mensaje informando que los datos fueron modificados correctamente.
Referencias	RF2, Gestionar Datos Primarios (include).
Precondiciones	El usuario se ha autenticado.
Postcondiciones	Se modifican los datos del paciente.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Genetista Nacional o Provincial quiere insertar los datos complementarios de un paciente.	2. El registro en dependencia del nivel de acceso del Genetista Nacional o Provincial, muestra una interfaz para buscar el paciente.
3. El Genetista Nacional o Provincial quiere modificar los datos complementarios existentes en el registro de un paciente determinado.	4. El registro en dependencia del nivel de acceso del Genetista Nacional o Provincial, muestra un formulario para modificar los datos complementarios del paciente.
5. Actualiza los datos complementarios del paciente.	6. El registro muestra un mensaje especificando que los datos del paciente fueron modificados satisfactoriamente.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
1. El Genetista Nacional o Provincial introdujo datos complementarios no válidos.	2. El registro muestra un mensaje de error por cada campo de datos no válidos.
Prioridad: Secundario	

Tabla 2: Descripción del caso de uso del sistema Modificar datos complementarios del paciente

CU-3	Generar reportes
-------------	------------------

Actores	Genetista Nacional / Provincial (Inicia).	
Propósito	Obtener reportes por distintas categorías.	
Resumen	El caso de uso se inicia cuando el Genetista Nacional o Provincial va a solicitar algún reporte. El registro le muestra la interfaz correspondiente para escoger un criterio de búsqueda. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.	
Referencias	RF3	
Precondiciones	El usuario se ha autenticado.	
Postcondiciones	Se muestran los resultados de los reportes.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Genetista Nacional o Provincial desea obtener algún reporte por alguna categoría	2. El registro en dependencia del nivel de acceso del Genetista Nacional o Provincial, muestra un formulario para escoger la categoría de reporte que vaya a solicitar.	
3. Selecciona la categoría por la cual quiere solicitar reporte.	4. El registro muestra el reporte según la categoría seleccionada anteriormente.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
1. El Genetista Nacional o Provincial solicita un reporte de datos que no han sido ingresados	2. El registro muestra un mensaje de error: "Ha solicitado un reporte de datos que no han sido ingresados".	
Prioridad: Secundario		

Tabla 3: Descripción del caso de uso del sistema Generar reportes

La descripción completa del caso de uso Gestionar datos primarios se puede encontrar en el Expediente de proyecto del Registro Cubano de Historias Clínicas (RECUHCL).

CU-4	Gestionar datos primarios
Actores	Especialista Genético (Inicia), Registro de Ciudadano (RC)
Propósito	Muestra los datos de la planilla correspondiente al paciente.

Resumen	Registrar los datos personales de los pacientes del Centro Nacional de Genética Médica.	
Referencias	2.1.1, 2.1.2, 2.1.3	
Precondiciones	El Especialista Genético debe haberse autenticado.	
Postcondiciones	Se registran los datos primarios del Paciente.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Especialista Genético accede a la opción Datos Primarios.	2. El sistema muestra una interfaz de búsqueda para buscar al paciente.	
3. El Especialista Genético introduce los datos de búsqueda correspondientes.	4. El sistema muestra una interfaz con los resultados de la búsqueda en el RC o en el sistema.	
5. El Especialista Genético selecciona el paciente y marca la opción Historia Clínica y Registra.	6. El sistema valida e introduce los datos y crea la Historia Clínica.	
Prioridad: Crítico		

Tabla 4: Descripción del caso de uso del sistema Gestionar datos primarios

GLOSARIO DE TÉRMINOS

Apache: Es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

API: Application Programming Interface (en español: Interfaz de Programación de Aplicaciones).

Base de datos: Conjunto de datos organizados entre los cuales existe una correlación y que están almacenados con criterios independientes de los programas que los utilizan. La filosofía de las bases de datos es la de almacenar grandes cantidades de datos de una manera no redundante y que permita las posibles consultas de acuerdo a los derechos de acceso.

Caso de uso: Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

CGI: Common Gateway Interface, importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web.

Clasificación etiopatogénica: Se refiere a la clasificación de las patologías o enfermedades según su origen o causa.

CNGM: Son las siglas del Centro Nacional de Genética Médica.

CPU: Central Processing Unit (en español: Unidad Central de Procesamiento), es la denominación oficial del procesador de una computadora.

Copyright: Derecho de autor, es un conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores.

CSS: Cascading Style Sheets (en español: Hojas de Estilo en Cascada). Son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.

Enfermedad rara: Enfermedades raras, incluidas las de origen genético, son aquellas enfermedades con peligro de muerte o de invalidez crónica, que tienen una frecuencia (prevalencia) baja, menor de 5 casos por cada 10.000 habitantes en la comunidad.

Formulario: Parte de una página Web que el usuario completa y devuelve al servidor para su procesamiento.

FTP: File Transfer Protocol. Es un protocolo de transferencia de archivos entre sistemas conectados a una red basado en la arquitectura cliente-servidor.

Genética: Rama de las ciencias biológicas que trata de comprender cómo la herencia biológica es transmitida de una generación a la siguiente, y cómo se efectúa el desarrollo de las características que controlan estos procesos.

GoF: Gang of Four, conjunto de patrones de diseño.

GPL: General Public License (en español: Licencia Pública General), es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre.

GRASP: General Responsibility Assignment Software Patterns (en español: Patrones de software de asignación de responsabilidades a objetos).

Herramienta CASE: (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador). Aplicación informática destinada a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

HTML: Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTTP: Protocolo usado para la transferencia de documentos a través de la World Wide Web. Estas transferencias requieren un programa cliente http en un extremo de la comunicación y un servidor http en el otro.

Infomed: Red de Salud de Cuba.

Interfaz: Es la parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

KDE: K Desktop Environment o Entorno de Escritorio K, entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

Macs: Computadoras personales diseñadas, desarrolladas, construidas y comercializadas por Apple Inc.

MySQL: Es un sistema gestor de base de datos, multihilo y multiusuario, software libre en un esquema de licenciamiento dual.

MVC: Patrón arquitectónico Modelo Vista Controlador.

Navegador: Es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet. Esta red de documentos es denominada World Wide Web (WWW) o Telaraña Mundial. Los navegadores actuales permiten mostrar y/o ejecutar: gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

ODBC: Open Database Connectivity, estándar de acceso a Bases de Datos desarrollado por Microsoft.

PC: Personal Computer (en español: computadora personal).

PDF: Portable Document Format, (en español: Formato de Documento Portátil).

PDT: PHP Development Tool.

Plugin: Es un programa de ordenador que interactúa con otro programa para aportarle una función o utilidad, generalmente muy específica.

Protocolo: Conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red. En este contexto, las entidades de las cuales se habla son programas de computadora o automatismos de otro tipo, tales y como dispositivos electrónicos capaces de interactuar en una red.

Reporte: Informe detallado sobre alguna información, o sobre el estado de la información.

SAAA: Componente de seguridad basado en el modelo de Autenticación, Autorización y Auditoría.

Servidor: Una computadora que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final.

Servidor Web: Programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir los llamados hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

SGBD: Sistema Gestor de Base de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

SIGM: Sistema Informático de Genética Médica.

SISalud: Sistema de Información para la Salud.

Software: Es la parte lógica del ordenador, esto es, el conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Es el conjunto de instrucciones que permite la utilización del equipo.

Versión: Término que nombra las actualizaciones de un producto, se utiliza cuando se saca al mercado, o bien, cuando las modificaciones que se hacen del antiguo son muy numerosas o de gran alcance.

XML: Extensible Markup Language (en español: Lenguaje de Marcas Extensible). Es un metalenguaje extensible de etiquetas.

YAML: Es un formato para serializar datos que son fáciles de procesar por las máquinas, fácil de leer para las personas y fácil de interactuar con los lenguajes script.