

Universidad de Ciencias Informáticas
"Facultad 6"



**TÍTULO: “SISTEMA INFORMÁTICO PARA
LA RED NACIONAL DE GENÉTICA MÉDICA: REGISTRO
CUBANO DE DISCAPACITADOS, VERSIÓN 2.0”.**

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

Autores: Alelí Martínez Franco
Raciel Evelio Ramírez Romero

Tutor: Ing. Yadira Robles Aranda
Co-tutor: Dra. Liliana Argelia Casar Espino

Ciudad de La Habana, Cuba.

Junio, 2008.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de esta tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

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

Alelí Martínez Franco

Raciel Evelio Ramírez Romero

Firma del autor

Firma del autor

Ing. Yadira Robles Aranda

Firma del tutor

"El hombre crece con el trabajo que nace de sus manos".

José Martí



Agradecimientos

Alelí Martínez Franco

Ya han pasado cinco septiembres de sacrificios, de sueños, y al fin se materializan las horas de estudios con la realización de estas páginas, las cuales sin el apoyo de aquellos que me tendieron sus manos y me mostraron su confianza aún cuando el camino parecía intransitable no hubieran sido posibles.

Agradezco a mi madre por entregarme su vida, por posponer sus sueños en aras de realizar los míos, por hacer de mi lo que soy, pero sobre todo por enseñarme que los valores humanos son lo esencial para el mejoramiento diario.

A mi padre por apoyarme, por el cariño, por dejarme ser parte del centro de su mundo, por estar cuando lo necesité.

A mi hermano por ser mi amigo, mi ídolo, mi segundo papá, por hacer realidad los versos Martianos: Nunca es un niño mas bello que cuando lleva del brazo a su hermana, para que nadie se la ofenda.

A mi abuelita Enélida por dedicar su existencia a construirme un mundo maravilloso, por demostrarme que las tristezas de la vida no pueden endurecer el alma.

A mi abuelita Andrea por el cariño, por hacerme sentir tan importante.

A mi hermanito Carlos por hacerme sentir responsable de él, por quererme tanto.

A Ángel, por ser el amor de mi vida y llenarme de ilusiones. Por el apoyo en este tiempo turbulento, por cuidarme como la niña de sus ojos.

A mis tías y primos por amarme tanto, por confiar en mí. Al tío que aun sin haberlo visto vivió siempre conmigo.

A mis suegros por permitirme formar parte de su familia, por ser maravillosos.

A mis amigos de siempre Liane, Jiesie, Liseidis, Héctor, por darme la oportunidad de conocer el sentimiento más sano y profundo del mundo, por aceptarme y quererme.

A la UJC por profundizar mis sentimientos Revolucionarios, por hacer de mi una joven responsable y orgullosa de mi Patria.

A mis tutoras, por rectificar con cariño y amor mis errores, por asesorarme a lo largo de la tesis y acompañarme en este camino que hoy culmina.

A mis profesores durante estos 5 años, por educarme y contribuir con amor y dedicación a mi preparación profesional.

A Raciél, por aceptar mis majaderías durante estos 5 años, por ser una de las personas que más quiero.

A todos aquellos que forman parte de mi mundo, que aportaron un poco de amor y apostaron sus sueños para hacer de mi vida una experiencia maravillosa.

Raciel Evelio Ramirez Romero

Para agradecer con la verdad en una mano y el corazón en la otra, no bastan simples palabras, confío mas en los hechos, lo que pueda demostrar y sea capaz de hacer a cambio, aunque no me pidan nada.

Siempre estaré en deuda con la Revolución y el ejemplo de Fidel, principal protagonista hoy y siempre de nuestra formación como revolucionarios.

A mis padres y abuelos por darme todo lo que necesité, por siempre estar a mi lado, pendientes de todo, disfrutando cada alegría y aconsejándome en cada error. A toda mi familia, amigos y vecinos.

A la universidad que me acogió como mi segundo hogar y me aportó más que fríos conocimientos.

A la facultad que me formó como profesional de bien y a su dirección que con certera guía nos permitió llegar a tan esperada meta.

A la Dra. Liliana por permitirme el honor de trabajar junto a ella todos estos años.

A mis tutoras por estar siempre al tanto del trabajo

A la dirección de mi proyecto y al Polo, por su exigencia, constancia y eficiencia en la producción, que permitieron lograr los resultados que exponemos hoy.

No me olvido de extensión universitaria, que me permitió representar el movimiento de artistas aficionados de la universidad dentro y fuera de la misma. A mi profesora Milagros y otros que tanto me ayudaron.

Si escribiera todos los nombres que vienen a mi mente cuando redacto estas líneas, no tendría fin mencionando compañeros, dirigentes y amigos que siempre han estrechado mi mano y han estado pendientes de mí.

La deuda es inmensa, solo puedo pagar con eterna gratitud.

A todos, muchas gracias.

Dedicatoria

A nuestros padres por su entrega y confianza, y a nuestro Comandante en Jefe.

Resumen

El Centro Nacional de Genética Médica, tiene entre sus objetivos principales realizar un estudio psico-social de las personas que presentan discapacidades físicas y mentales en el país. Para hacer efectiva la investigación es fundamental la recolección, almacenamiento, gestión y actualización de la información. Para dar solución a esta necesidad se realizó en el año 2007 una primera versión de un Sistema Informático por parte de la UCI, titulado: Sistema Informático para la Gestión de Información del Registro Cubano de Discapacitados, el cual no se integró a la red de salud cubana, imposibilitando que los especialistas del país pudieran acceder al Sistema Informático desde la misma. Con el propósito de solucionar esta problemática se realiza la presente investigación para crear El Registro Cubano de Discapacitados, versión 2.0, como parte del Sistema Informático para la Red Nacional de Genética Médica (SIGM), el cual posibilitará a los genetistas del país la realización de los estudios antes mencionados desde la red nacional de salud.

El estar integrados al SIGM, permite una óptima distribución y estandarización del trabajo, de forma tal que posibilita la elaboración conjunta y optimizada de la base de datos, el diseño de la arquitectura y la gestión de software. El nuevo diseño e implementación del Registro Cubano de Discapacitados, se espera que conforme una herramienta para contribuir al desarrollo de la genética en Cuba.

Palabras claves: Registro, discapacitados, sistema informático.

ÍNDICE

Agradecimientos	II
Dedicatoria.....	IV
Resumen	V
Introducción.	- 1 -
Capítulo 1: Fundamentación Teórica.....	- 4 -
1.1 Gestión de información de salud.....	- 4 -
1.1.1 Sistemas vinculados a la gestión de la información de la salud.....	- 4 -
1.1.2 Normativas en la creación de sistemas informáticos para la Salud Pública en Cuba.....	- 5 -
1.1.3 Sistema Informático para la Red Nacional de Genética Médica.	- 6 -
1.2 Metodología.....	- 6 -
1.3 Lenguaje Unificado de Modelado (UML).....	- 8 -
1.4 Tecnologías a utilizar.....	- 8 -
1.4.1 Lenguaje de programación.....	- 8 -
1.4.2 PHP 5.....	- 9 -
1.4.3 JavaScript.....	- 9 -
1.4.4 Servidor Web HTTP Apache 2.0.....	- 10 -
1.4.5 Sistema gestor de base de datos MySQL 5.0.....	- 10 -
1.5 Herramientas Utilizadas.....	- 11 -
1.5.1 Herramienta CASE: Visual Paradigm for UML Community Edition 3.1.....	- 11 -
1.5.2 Framework (Symfony 1.0).....	- 11 -
1.5.3 Entorno de desarrollo: Eclipse.....	- 12 -
1.5.4 Sistema de control de versiones: Subversion 1.4.5.....	- 13 -
1.6 Roles y artefactos	- 13 -
1.6.1 Rol Diseñador.....	- 14 -
1.6.2 Rol Implementador.....	- 15 -
1.7 Patrones de Diseño.....	- 15 -
1.7.1 Patrones GRASP.....	- 15 -
1.7.2 Patrones GOF.....	- 16 -
1.8 Conclusiones	- 17 -

Capítulo 2: Diseño	- 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	- 22 -
2.2 Diseño.....	- 23 -
2.2.1 Patrones GRASP implementados.....	- 23 -
2.2.2 Tipos de patrones GOF que implementa Symfony y otros.....	- 24 -
2.2.3 Arquitectura MVC.	- 25 -
2.2.5 Arquitectura MVC utilizada por Symfony 1.0.....	- 26 -
2.2.6 Diagrama de Clases del diseño Web.....	- 27 -
2.2.7 Diagrama de interacción.....	- 32 -
2.2.6 Diagrama de Despliegue.....	- 37 -
2.2.7 Seguridad.....	- 37 -
2.2.8 Pautas del diseño.....	- 38 -
2.2.9 Prototipo de Interfaz	- 40 -
2.2.10 Validación del sistema.....	- 42 -
2.2.11 Conclusiones.....	- 42 -
Capítulo 3: Implementación	- 43 -
3.1 Diagrama de Componentes.	- 43 -
3.2 Estilos de codificación definidos para el SIGM.....	- 46 -
3.3 Código fuente de las principales clases.	- 47 -
3.4 Interfaces de la aplicación.....	- 52 -
3.5 Validación del sistema a nivel de desarrollador.....	- 61 -
3.6 Conclusiones	- 61 -
Conclusiones	- 62 -
Recomendaciones	- 63 -
Referencias bibliográficas	- 64 -
Bibliografía.....	- 67 -
Índice de anexos.....	- 67 -
Glosario de Términos.....	- 91 -

Introducción.

Desde el triunfo de la Revolución cubana en 1959, ha sido prioridad del estado el empeño permanente en el mejoramiento de la calidad de vida del pueblo. En el cumplimiento de esta primicia ha desempeñado un papel protagónico el Sistema Nacional de Salud llevando a cabo importantes reformas con el objetivo de incrementar la eficiencia y calidad de sus servicios. Han sido creados nuevos centros de investigación para promover el desarrollo de la producción científica del país, entre los que se destaca el Centro Nacional de Genética Médica (CNGM), producto de las numerosas e importantes investigaciones realizadas.

El CNGM, institución científica dedicada a la asistencia, la docencia, la investigación y la promoción de servicios de salud, tiene como objetivo fundamental llevar a cabo acciones en el campo de los problemas de salud de carácter genético, desarrollando importantes estudios que han permitido aumentar la calidad de vida del pueblo cubano y minimizar la influencia de las enfermedades genéticas en el cuadro de mortalidad del país. Este centro de investigación orienta, evalúa y controla la ejecución de los distintos subprogramas del Programa de Diagnóstico y Prevención de Enfermedades Genéticas en el país; incluye diferentes registros creados con el objetivo de almacenar la información necesaria para llevar a cabo estudios en el campo de las enfermedades genéticas tales como: el Registro Cubano de las Personas con Retraso Mental, el Registro Cubano de Gemelos, el Registro Cubano de Malformaciones Congénitas y el Registro Cubano de Discapacitados, siendo este último de vital importancia, ya que en Cuba uno de los objetivos fundamentales se centran en la atención permanente de las personas que presentan discapacidades físicas y mentales, así como su integración a la sociedad y el CNGM, tiene la misión realizar el estudio psico-social con el objetivo de identificar las principales causas de discapacidad así como las necesidades fundamentales de estas personas.

En principio, para lograr almacenar y gestionar la información referente a las personas con discapacidad del país, se realizó una primera versión de un sistema informático por parte de la Universidad de Ciencias Informáticas, titulado: Sistema Informático para la Gestión de la Información del Registro Cubano de Discapacitados. El alcance de dicho proyecto permitió la conformación y estructuración de la información, definir las funcionalidades, analizar, diseñar e implementar el mismo, con el inconveniente de que no fue publicado en la red de salud cubana (*INFOMED*), puesto que contaba con una base de datos independiente que almacenaba información existente en el Sistema de Información para la Salud, provocando la replicación de dichos datos.

Es por esto que surge el problema científico **¿Cómo obtener un producto funcional a partir de los requerimientos identificados para el Registro Cubano de Discapacitados versión 2.0 integrado a la red nacional de salud?**

Debido a esto el objeto de estudio de la investigación es **El Proceso de gestión de información de la salud** y como campo de acción de la misma: **Proceso de desarrollo de software para la gestión de la información de discapacidades físicas-motoras del Sistema Informático para la Red Nacional de Genética Médica.**

De ahí surge como objetivo general: **Desarrollar el diseño y la implementación del Registro Cubano de Discapacitados versión 2.0 del Sistema Informático para la Red Nacional de Genética Médica.**

Para lo que se proponen los siguientes objetivos específicos:

1. Diseñar el Registro Cubano de Discapacitados versión 2.0 del Sistema Informático para la red Nacional de Genética Médica.
2. Implementar el Registro Cubano de Discapacitados versión 2.0 del Sistema Informático para la red Nacional de Genética Médica.

Para darle solución al problema científico planteado y lograr el cumplimiento de los objetivos se llevaron a cabo las siguientes tareas:

- Análisis de las Normas para el desarrollo de sistemas informáticos para la Salud en Cuba.
- Análisis del diseño del Registro Cubano de Discapacitados versión 2.0 del SIGM.
- Diseño del Registro Cubano de Discapacitados versión 2.0. del SIGM.
- Implementación de los componentes del Registro Cubano de Discapacitados versión 2.0 del SIGM.
- Validación del sistema a nivel de desarrollador.

La implementación del Registro Cubano de Discapacitados (RECUDIS), versión 2.0 del SIGM, hará posible que los especialistas puedan gestionar la información de las personas con discapacidad del país de forma rápida y segura. El mismo formará parte de la nueva integración del Sistema Informático de Genética Médica.

El presente trabajo de diploma se estructura en introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

En el **Capítulo 1: Fundamentación Teórica**, se analizan y describen, la metodología, los roles que se emplearán producto de las necesidades actuales, patrones de diseño, herramientas, lenguajes de programación definidos para el SIGM y demás elementos asociados al estado del arte.

En el **Capítulo 2: Diseño**, se aborda el desarrollo de esta fase, se muestran, los detalles relacionados con el diseño del sistema informático propuesto y se utilizan para su modelado los diagramas de clases de diseño y los diagramas de iteración para los diferentes casos de uso, así como el diagrama de despliegue. Se toman en cuenta aspectos de seguridad, prototipo de interfaz y validación.

En el **Capítulo 3: Implementación**, se muestra la implementación del sistema, se analizan los estilos de codificación, y se brinda una solución a los requisitos especificados. Se obtienen los diagramas de componentes, se muestra el código fuente de las principales clases así como Interfaces de la aplicación.

Capítulo 1: Fundamentación Teórica.

En el presente capítulo se realiza una breve descripción de Sistemas Informáticos que contribuyen a la gestión de la información de la salud, haciendo énfasis en aquellos que fueron desarrollados para gestionar la información de la salud en Cuba. En este capítulo también se describe el objeto de estudio, y se hace una breve explicación de sistemas automatizados vinculados al campo de acción. Además, se describen brevemente las tecnologías, las herramientas, los roles que se desarrollan y los artefactos generados, así como los patrones de diseño utilizados.

1.1 Gestión de información de salud.

En aras de lograr mantener el avance en el campo de la salud al vertiginoso nivel del progreso científico técnico mundial, se hace indispensable un equilibrio entre el conocimiento científico y la dinámica del contexto actual en el uso de las tecnologías de la informática y las comunicaciones. Por esta razón se evidencia la necesidad de nuevas aplicaciones informáticas dirigidas a elevar la calidad en la gestión de la información referente a la salud, partiendo del uso de los recursos científicos y tecnológicos disponibles.

1.1.1 Sistemas vinculados a la gestión de la información de la salud.

En la actualidad no existe a nivel mundial un sistema informático que garantice la gestión de la información de personas con discapacidades físicas para realizar estudios psicosociales.

En bases de datos bibliográficos entre los que se encuentra MedLine considerada una de las mayores del mundo en Literatura Internacional en Ciencias de la Salud, no existen referencias sobre estudios realizados a nivel mundial con relación a las personas con discapacidades vinculadas a un sistema informático para la gestión de ese tipo de información.

Los sistemas de gestión de información de salud existentes en la actualidad se centran en la gestión hospitalaria, automatización, bibliotecas virtuales y telemedicina, entre ellos se encuentran: ArchiMED, para gestión de las consultas médicas, A-medic, para la gestión de datos acerca de las filiaciones, diagnósticos, antecedentes e historiales de pacientes. En software libre se encontró OpenHIS, Sistema Informático Hospitalario.

Cuba desarrolla proyectos de informatización de salud, como son: El portal cubano de árbol genealógico, el Registro de Malformaciones Congénitas, de enfermedades genéticas, el Registro de Gemelos, el Registro de Historias Clínicas y el Registro cubano de discapacitados.

En el año 2007 se realizó una primera versión de un sistema informático por parte de la Universidad de Ciencias Informáticas, titulado: Sistema Informático para la gestión de la información del Registro Cubano de Discapacitados, cuyo objetivo era desarrollar un sistema Informático capaz de almacenar y gestionar toda la información referente a las personas con discapacidad del país, el mismo fue implementado sobre tecnología LAMP (*Sistema operativo Linux, Servidor Web Apache 2.0, Gestor de bases de datos MySql 5.0 y lenguaje de programación PHP 5*).

Esta primera versión creada por la Universidad de Ciencias Informáticas no se pudo integrar a la red de salud cubana.

1.1.2 Normativas en la creación de sistemas informáticos para la Salud Pública en Cuba

El Ministerio de Salud Pública (MINSAP) establece un grupo de premisas, estándares y herramientas con el objetivo de dar continuidad y sostenibilidad a los sistemas informáticos que emplea, por lo que se hace necesario cumplir con una serie de aspectos que constituyen guía inherente en la construcción de un sistema informático para la gestión de la información de la salud en Cuba, de los que se hará una breve mención.

Debe ser basado en el principio de software libre y acorde a la infraestructura, mantener las capas de la presentación, negocio y datos en tres servidores dedicados, por último el diseño e implementación retoman el concepto de la arquitectura orientada a servicios y el despliegue en tres capas. En el caso de la presentación, esta contendrá la interfaz de usuario y negocio de la validación de formularios de entrada de datos, la capa del negocio que contendrá los servicios que dan cumplimiento a los requisitos funcionales del sistema y la capa de datos que contiene la Base de Datos, todos los componentes siguen también principios de alta cohesión y bajo acoplamiento. Se desarrollará una arquitectura para el SIGM, teniendo en cuenta estos elementos. [1]

1.1.3 Sistema Informático para la Red Nacional de Genética Médica.

Con el objetivo de lograr un producto funcional mediante la integración de los proyectos independientes del CNGM acorde a la arquitectura de salud, surge el Sistema Informático para la Red Nacional de Genética Médica, donde cada registro de dicho centro representa una parte fundamental dentro del nuevo sistema que los integra, compartiendo el mismo tipo de arquitectura en su diseño, así como la Base de Datos que utilizan para el almacenamiento y gestión de la información. Dada la semejanza que existe entre los registros en cuanto a características comunes respecto a los datos de las personas que en ellos se archivan, además de pertenecer a un mismo cliente y compartir el objetivo común de ser utilizados por los genetistas de todo el país, se hace necesario a tales efectos la integración de los mismos, quedando conformado el Sistema Informático para la Red Nacional de Genética Médica. (Ver anexo 1)

1.2 Metodología.

Las metodologías de desarrollo de software definen un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de software. Una metodología organiza durante el proceso de desarrollo las actividades que se deben realizar para lograr la obtención de un producto final, definiendo las personas que participan, sus responsabilidades, cuándo y cómo cumplirlas.

Se eligió la metodología: Rational Unified Process (RUP), debido a que unifica los mejores elementos de las metodologías existentes, y es una de las más empleadas en la actualidad. La misma está diseñada para desarrollar grandes y complejos proyectos.

RUP es un proceso de desarrollo de software (*conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software*) basado en componentes (*parte física y reemplazable de un sistema que conforman un conjunto de interfaces y proporcionan la implementación de dicho conjunto*) [2] y utiliza el lenguaje unificado de modelado (UML) para representar todos los esquemas de un sistema de software.

RUP se divide en 4 fases del desarrollo del software:

Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.

Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.

Construcción: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

Transición: El objetivo es llegar a obtener la liberación del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. [2- 4]

Las actividades en RUP se han agrupado en 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los 3 últimos como de apoyo, siendo estos: [2, 5]. (Ver Anexo 2)

Modelo del Negocio.

Requerimientos.

Análisis y Diseño.

Implementación.

Pruebas.

Instalación.

Administración del proyecto.

Administración de configuración y cambios.

Ambiente.

El Proceso Unificado tiene 6 características principales [2,3]:

1. Guiado/Manejado por casos de uso.
2. Centrado en arquitectura.
3. Iterativo e Incremental.
4. Desarrollo basado en componentes.
5. Utilización de un único lenguaje de modelación.
6. Proceso Integrado.

1.3 Lenguaje Unificado de Modelado (UML).

UML (Unified Modeling Language) o Lenguaje Unificado de Modelado 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*). UML es un lenguaje muy detallado y uniforme para el diseño Orientado a Objetos, incrementando las potencialidades de éxito en el desarrollo de los mismos. [6, 7].

Sus principales características son:

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

1.4 Tecnologías a utilizar.

Se debe cumplir con ciertas características establecidas para el diseño y la implementación del Registro Cubano de Discapacitados versión 2.0, de modo que sea posible su integración al Sistema Informático para la Red Nacional de Genética Médica. Para llevar a cabo este propósito se utilizará PHP 5 como lenguaje de programación, MySQL 5.0 como Sistema Gestor de Base de Datos y Apache 2.0 como servidor de aplicaciones.

1.4.1 Lenguaje de programación.

Para el desarrollo de aplicaciones Web es necesario conocer a fondo diferentes lenguajes de programación como PHP 5 y JavaScript. Los mismos fueron seleccionados en el trabajo de diploma Sistema Informático de Genética Médica: Diseño de la Arquitectura el cual definió los elementos de la arquitectura en el diseño del SIGM.

A continuación se describirán los lenguajes utilizados en esta investigación.

1.4.2 PHP 5.

PHP 5 es un lenguaje de programación usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa "*PHP Hypertext Pre-processor*", el mismo permite embeber sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz además soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape y iPlanet, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd entre otros.

PHP 5 es la última versión del lenguaje, una renovación total que se ha extendido a numerosos aspectos, como la interacción con otras tecnologías y la nueva sintaxis de orientación a objetos.

PHP un lenguaje "código abierto" y puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía Web para una base de datos es una tarea simple con PHP. [Martínez] El resultado es normalmente una página HTML, por lo que al usuario le parecerá que está visitando una página HTML que cualquier navegador puede interpretar.

Al ser un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. [8]

1.4.3 JavaScript.

JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, es decir, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador. Es un lenguaje utilizado principalmente en páginas Web.

No es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.[9]

1.4.4 Servidor Web HTTP Apache 2.0.

Un servidor web no es más que un programa que ejecuta de forma continua en un ordenador (también se utiliza el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador de internet) y que contesta a estas peticiones de forma adecuada, sirviendo una página web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error.

El servidor HTTP Apache 2.0 es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección *core* (*núcleo*) y mucha de la funcionalidades que podrían considerarse básicas para un servidor web es provista por módulos.

Las ventajas que presenta entre otras son: Modular, Código abierto, Multi-plataforma, Extensible, Popular (fácil conseguir ayuda/soprote) y gratuito. [10]

1.4.5 Sistema gestor de base de datos MySQL 5.0.

MySQL 5.0 es un sistema de gestión de base de datos relacional (*modelo de datos basado en la lógica de predicado y en la teoría de conjuntos.*), multihilo (*es una característica que permite a una aplicación realizar varias tareas concurrentemente*) y multiusuario (*o al sistema operativo o programa que es capaz de proveer servicio y procesamiento a múltiples usuarios simultáneamente*), MySQL 5.0 es muy utilizado en aplicaciones web.

Su popularidad como aplicación web está muy ligada a PHP. MySQL es gestor de base de datos muy rápido en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja

conurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL 5.0 sea ideal para este tipo de aplicaciones. [8]

1.5 Herramientas Utilizadas.

Se utiliza Visual Paradigm 6.0 para realizar los diagramas, el Framework (Symfony 1.0) para un desarrollo más rápido y simple del sistema informático, como entorno de desarrollo Eclipse, Servidor Web HTTP Apache 2.0, Sistema gestor de base de datos MySQL 5.0 y el Sistema de control de versiones: Subversion 1.4.5 para facilitar la administración de las distintas versiones de cada producto desarrollado. Las mismas fueron seleccionadas en el trabajo de diploma Sistema Informático de Genética Médica: Diseño de la Arquitectura el cual definió la línea base de la arquitectura en el diseño del SIGM.

1.5.1 Herramienta CASE: Visual Paradigm for UML Community Edition 3.1.

Visual Paradigm for UML Community Edition 3.1 para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software utiliza como lenguaje de modelado UML, ayuda a construir con mayor rapidez y eficiencia aplicaciones con alta calidad. 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.

Permite realizar ingeniería tanto directa como inversa ya que a partir de un modelo relacional en MySQL 5.0, es capaz de desplegar todas las clases asociadas a las tablas (*siguiendo el patrón de diseño Una Clase-Una Tabla*). [11]

1.5.2 Framework (Symfony 1.0).

Un Framework es un conjunto de herramientas que pueden ser reutilizadas, de forma tal que el desarrollo de grandes proyectos se realiza de una manera más simple y rápida. Para PHP 5 existe un gran número de framework, dentro de los cuales se encuentra Symfony un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. 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, lo que permite al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. [12]

1.5.3 Entorno de desarrollo: Eclipse

Eclipse es un entorno de desarrollo integrado (IDE) de código abierto independiente de una plataforma para desarrollar aplicaciones basadas en navegadores. Eclipse es gratuito. Esta plataforma, usada típicamente para desarrollar entornos de desarrollo integrados, Sin embargo, también se puede usar para otros tipos de aplicaciones cliente. En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in).

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La versión actual de Eclipse dispone de las siguientes características:

Editor de texto

Resaltado de sintaxis

Compilación en tiempo real

Pruebas unitarias con JUnit

Control de versiones con CVS

Integración con Ant

Asistentes (*wizards*): para creación de proyectos, clases, tests, etc.

Refactorización

Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversión e Integración con Hibernate. [13]

1.5.4 Sistema de control de versiones: Subversion 1.4.5

Un 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 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. Esto 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. [14]

1.6 Roles y artefactos

Se mostrarán los roles y los artefactos que se generan en las dos dimensiones que propone RUP de acuerdo al ciclo de diseño e implementación correspondiente: Los flujos de trabajo de diseño e implementación en las fases de elaboración y construcción. [2,15]. (Ver Anexo 3)

Fase de elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

Fase construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene una o varias liberaciones del producto que han pasado las pruebas. Se ponen estas liberaciones a consideración de un subconjunto de usuarios.

Flujo de trabajo de Diseño: Describe cómo el sistema informático será realizado a partir de la funcionalidad prevista y las restricciones impuestas, indicando con precisión lo que se debe programar.

Flujo de trabajo de 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.

1.6.1 Rol Diseñador

En la metodología RUP el diseñador es el responsable de diseñar una parte del sistema informático 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 el presente trabajo 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 análisis: las clases de análisis representan un modelo de concepto tempranamente dado para los elementos que en el sistema, tienen responsabilidades y comportamiento.

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

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.

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

1.6.2 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 el presente trabajo 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) y ficheros de datos.

1.7 Patrones de Diseño

Los patrones de diseño (design patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. [16,17]

1.7.1 Patrones GRASP.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Constituyen un apoyo para 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. [18,19]

1.7.1.1 Patrones Básicos de GRASP.

Los patrones básicos se refieren a cuestiones y aspectos fundamentales del diseño, ellos son: [17, 18,19]

Experto: Este 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 la 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 de métodos, con funcionalidad altamente relacionada y no realiza mucho trabajo. 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 tanto como sea posible.

Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.

1.7.2 Patrones GOF.

Los Patrones GOF, se dividen en 3 grupos, presentados a continuación [19]:

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.

1.8 Conclusiones

Se realizó un análisis sobre sistemas automatizados vinculados al campo de acción y lo que constituye el Sistema Informático de Genética Médica. Conjuntamente se llevó a cabo un estudio de las herramientas utilizadas y de los procesos de desarrollo, utilizando RUP como metodología a seguir. Se mostraron los roles y artefactos generados en los flujos y fases correspondientes al ciclo de desarrollo de diseño e implementación y se realizó una explicación de los patrones de diseño utilizados.

Capítulo 2: Diseño.

En este capítulo se mostrará lo referente a la fase de diseño de RECUDIS 2.0. Se representarán, el Modelo de Clases del Análisis, el Modelo de Clases del Diseño, los diagramas de secuencia de los diferentes escenarios de cada Caso de Uso, así como el Modelo de Despliegue. Se define un nuevo diseño acorde al uso de la arquitectura Modelo- Vista- Controlador implementada por el framework utilizado (Symfony).

2.1 Características del Sistema

Se toman como punto de partida inicial los requisitos funcionales identificados por el trabajo de diploma precedente. Se muestran los requisitos no funcionales identificados para el SIGM.

2.1.1 Requerimientos funcionales.

1. Insertar datos complementarios de un discapacitado.
2. Modificar datos complementarios de un discapacitado existente.
3. Buscar los datos primarios de un discapacitado.
4. Determinar la cantidad de discapacitados según la clasificación de su discapacidad.
5. Determinar la cantidad de discapacitados según el sexo.
6. Determinar la cantidad de discapacitados según su ocupación
7. Determinar la cantidad de discapacitados según el vínculo laboral.
8. Determinar la cantidad de discapacitados según su capacidad laboral
9. Determinar la cantidad de discapacitados con o sin amparo filial.

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.

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 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 clientes 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 RAM o superior.

Para el Cliente:

Microprocesador Pentium a 233 MHz.

64 MB de RAM o superior.

52 MB de espacio de disco duro o superior.

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

8. Disponibilidad:

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.

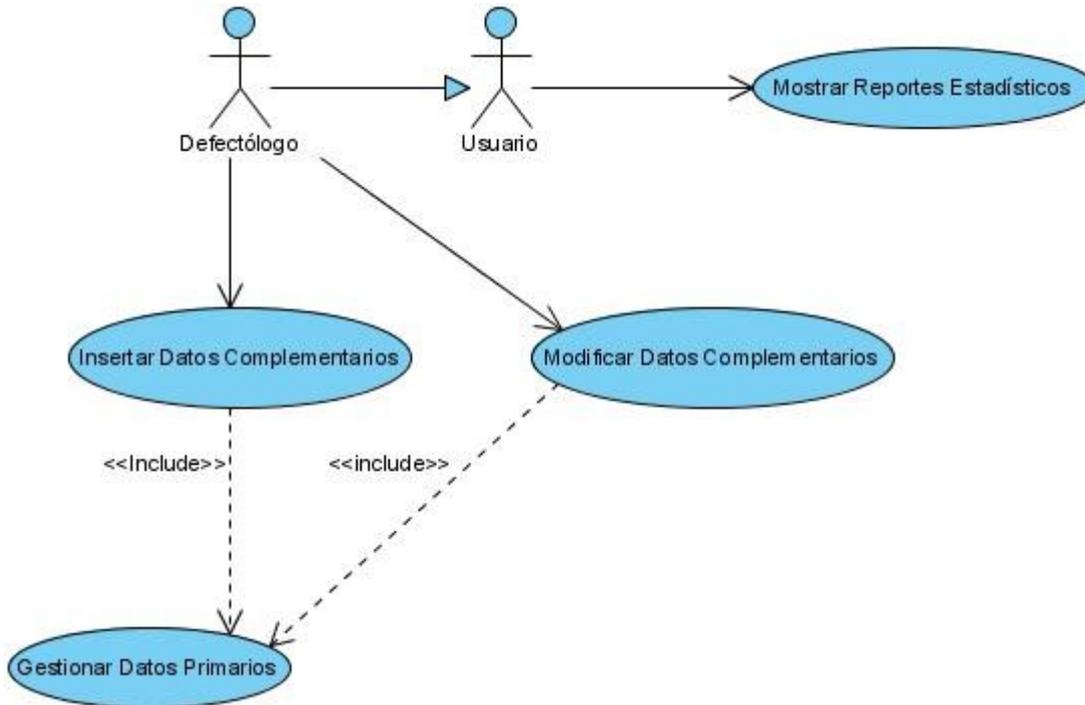
9. Requisitos Legales:

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



Descripción de casos de uso, (ver Anexo 4- 5).

2.1 Diseño

2.1.1 Patrones GRASP implementados.

Creador

En la clase `disActions` se encuentran las funciones definidas para el Registro Cubano de Discapacitados. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase `disActions` es “creador” de dichas entidades. Un ejemplo de su uso es en la función `DatosComplementariosDis ()` donde se crea una instancia de `DisDiscapitado`, convirtiendo a `disAction` en creadora de dicha entidad.

Experto

Este es utilizado en el modelo, puesto que Propel es la librería externa que utiliza Symfony para realizar su ORM (object-relational mapping), o mapeo de objetos a bases de datos. El mismo garantiza acceder a la base de datos como si fuera orientada a objetos a partir de su estructura relacional. El framework Symfony divide el modelo en dos capas, la primera de acceso a datos y la segunda de abstracción de datos donde las clases generadas contienen todos los atributos y funcionalidades únicas para acceder a los datos correspondientes y modificarlos.

Alta Cohesión

Symfony permite una excelente organización del trabajo en cuanto a la estructura y responsabilidades bien definidas para todos sus componentes, posibilitando que cada uno se centre en las tareas asignadas. Un ejemplo clave es el controlador, el cual delega en sus diferentes componentes, funciones para el manejo de los eventos del sistema estrechamente relacionados entre si y donde la clase `disActions` solo se encarga de definir las acciones a realizar.

Bajo acoplamiento

En la implementación MVC de Symfony, es en el modelo donde se manifiesta el uso de este patrón, ya que en las clases de abstracción que se relacionan con las de acceso a datos, los cambios que puedan realizarse en ellas tienen muy poca repercusión en el resto de los componentes asociados, dándole un alto grado de reutilización.

Controlador

Todas las peticiones al sistema son manejadas por un controlador que se divide en varios componentes encargados de la seguridad, validaciones, configuración y enrutamiento, de igual modo se definen las acciones a ejecutar por el sistema en la clase `disActions`, parte fundamental de dicho controlador.

2.1.2 Tipos de patrones GOF que implementa Symfony y otros.

Se utilizará el framework Symfony para el desarrollo del sistema informático, este framework utiliza una serie de patrones GOF como son:

En la categoría Creacionales:

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En la Clase `sfRouting` hay una llamada a `sfRouting::getInstance()`, que garantiza que exista una única instancia, creándola en caso que no exista o retornando la existente.

Prototype (Prototipo): Tiene como finalidad crear nuevos objetos duplicándolos, clonando una instancia previamente creada. Es usado para abstraer la lógica que decide que tipos de objetos se utilizarán, de la lógica que luego usarán esos objetos en su ejecución.

Cada clase base del modelo contiene una definición para el método `Copy()` y `CopyInto()`, para realizar la clonación de objetos.

En la categoría Estructurales:

Decorator (Envoltorio): Añade funcionalidad a una clase, dinámicamente. Por ejemplo un archivo `layout.php`, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación del patrón de este patrón de diseño. (Ver Anexo 6)

En la categoría de comportamiento:

Command (Comando): Manipular las peticiones de los usuarios y enviarlas a un objeto encargado de darle respuesta a las mismas.

Dentro del controlador existe una clase denominada sfRouting que contiene un método llamado Parser, el cual identifica o desglosa la dirección URL que recibe como parámetro en módulo, acción

Template Method (Método Plantilla): Define una estructura algorítmica en la súper clase, delegando la implementación a las subclases. Es decir, define una serie de pasos, en donde los pasos serán redefinidos en las subclases (polimorfismo). En symfony se implementan de tipo abstract (abstractas) para poder permitir redefinir sus métodos como por ejemplo las clases bases generadas en el modelo.

Otros

Controlador Frontal ("Front Controller"): Todas las peticiones Web son manejadas por un controlador frontal, que es el punto de entrada único para toda la aplicación en un entorno determinado.

Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción definida en la clase disActions y el nombre de un módulo con la URL entrado por el usuario, mientras otros componentes se encargan de la seguridad y validación mediante el uso de los archivos YML. Es decir se origina un flujo de eventos a partir del controlador frontal y se van delegando responsabilidades a cada componente hasta terminar en el disActions.

Registry (Registro): Mecanismos para almacenar datos globales.

En la clase sfConfig se crea como mecanismo para almacenar las variables globales de configuración para una aplicación de symfony.

2.1.3 Arquitectura MVC.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software 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 la Lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. [12]

2.1.5 Arquitectura MVC utilizada por Symfony 1.0.

Symfony realiza una arquitectura MVC de la siguiente forma [12]:

El modelo: Solo se encarga del acceso a los datos almacenados en el gestor de base de datos. Ha sido dividido en dos capas, la capa de acceso a los datos y en la capa de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no utilicen sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. Así, si se cambia de sistema gestor de bases de datos, solamente es necesario actualizar la capa de abstracción de la base de datos.

La vista: Las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabeceras de la página, el layout genérico, el pie de página y la navegación global. En la mayor parte de las veces sólo cambia el interior de la página. Por este motivo, la vista se separa en un layout y en una plantilla. Normalmente, el layout es global en toda la aplicación o al menos en un grupo de páginas. La plantilla sólo se encarga de visualizar las variables definidas en el controlador. Para que estos componentes interaccionen entre sí correctamente, es necesario añadir cierto código, código que será añadido a través de la lógica de la vista.

El controlador: Una parte importante de su trabajo es común a todos los controladores de la aplicación. Entre las tareas comunes se encuentran el manejo de las peticiones del usuario, el manejo de la

seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador se ha dividido en un controlador frontal, que se encarga de realizar las tareas comunes y las acciones, que incluyen el código específico del controlador de cada página. (Ver anexo 7)

2.1.6 Diagrama de Clases del diseño Web.

DIAGRAMA DE CLASES DEL DISEÑO. INSERTAR DATOS COMPLEMENTARIOS.

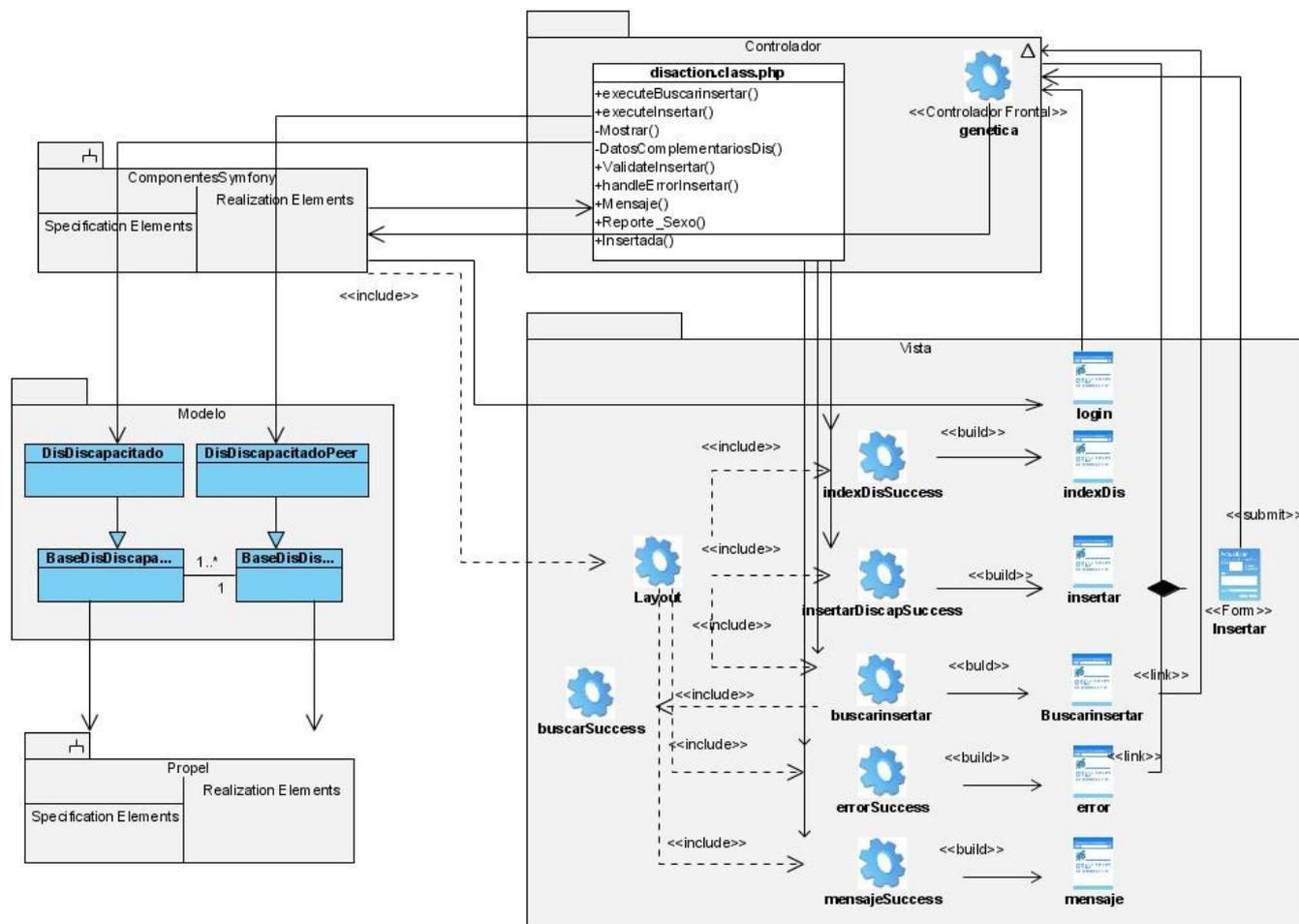


DIAGRAMA DE CLASES DEL DISEÑO. MODIFICAR DISCAPACITADO.

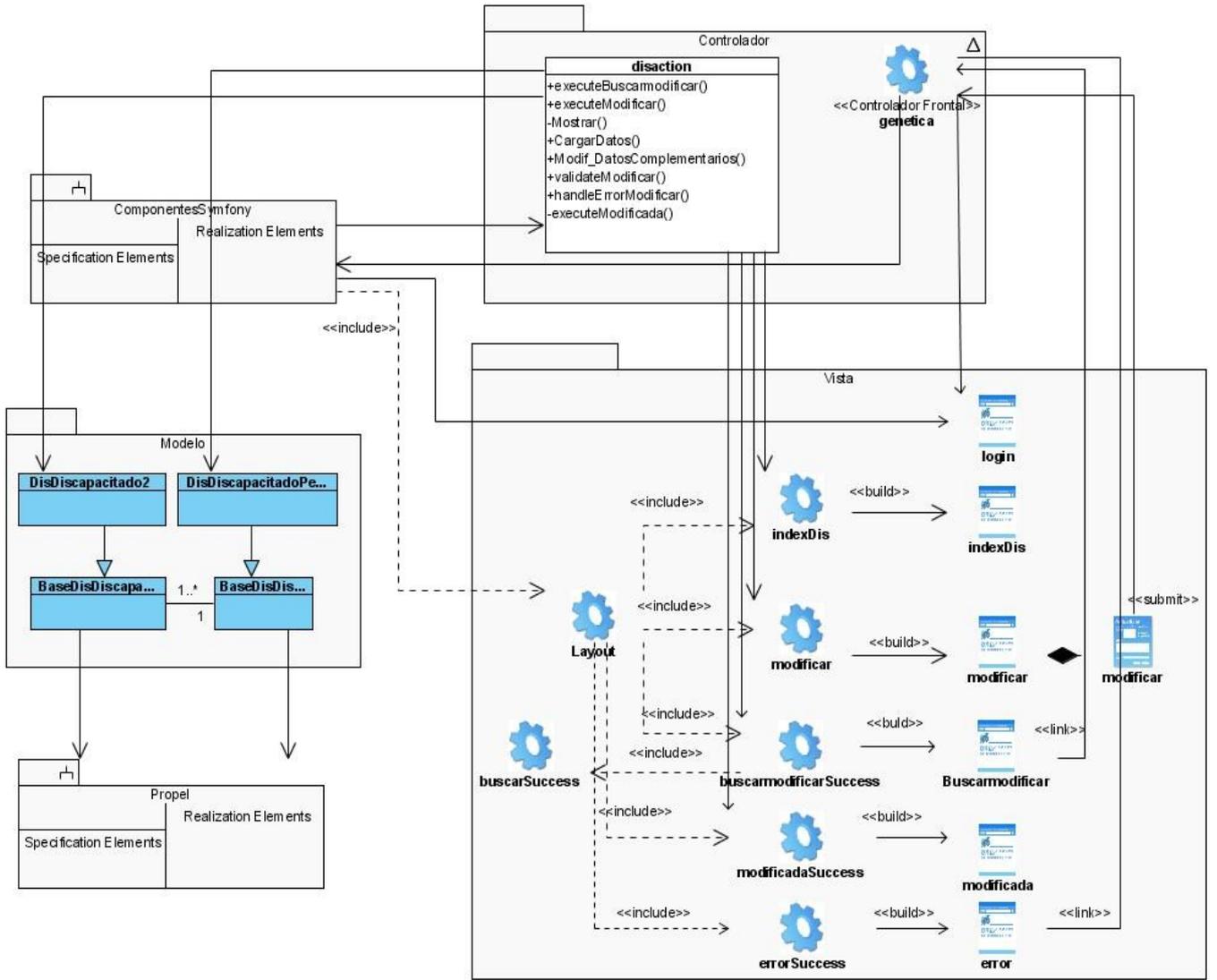


DIAGRAMA DE CLASES DEL DISEÑO. MOSTAR REPORTES ESTADÍSTICOS.

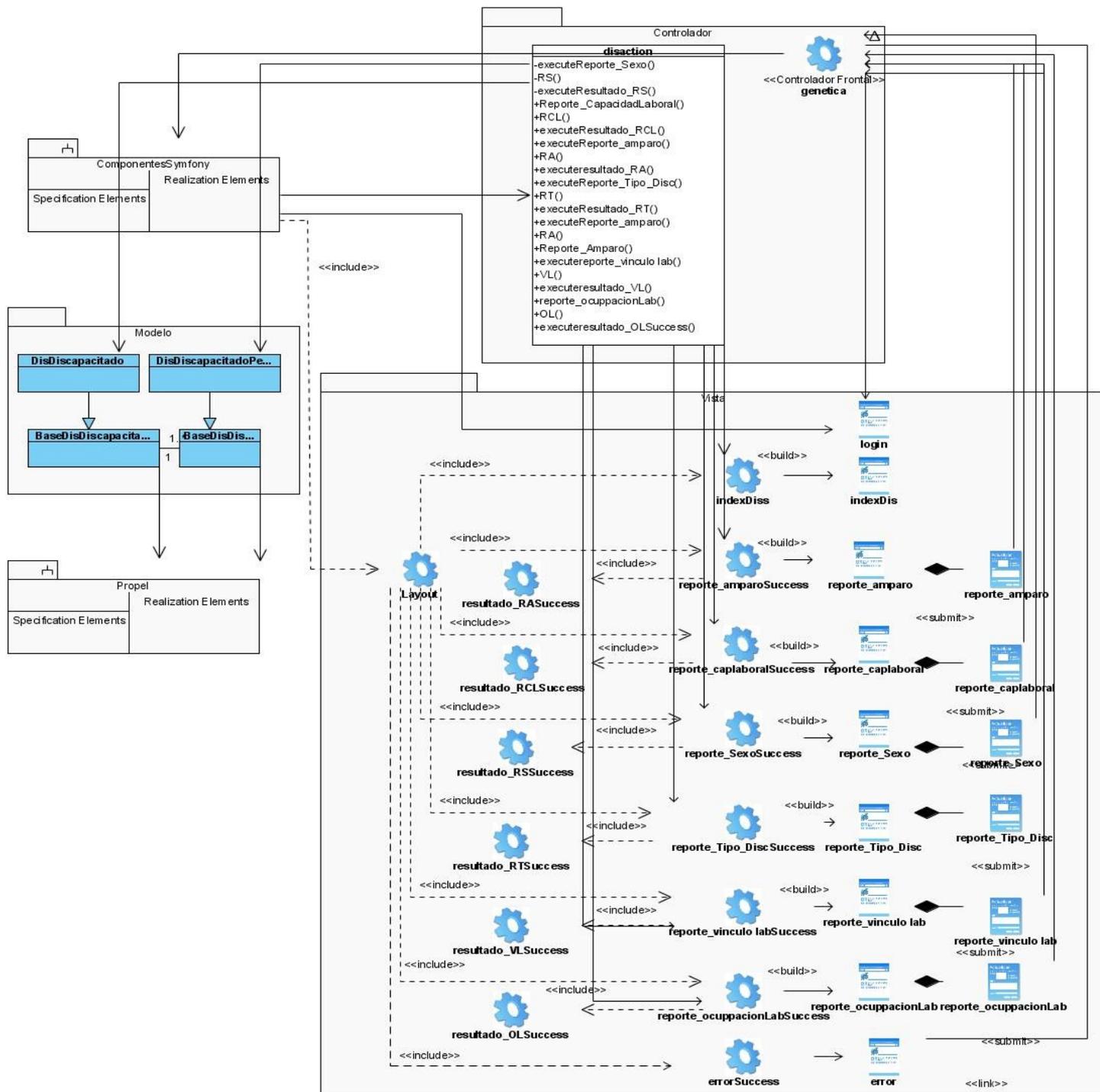


DIAGRAMA DE CLASES DEL DISEÑO. PAQUETE MODELO. Primera parte.

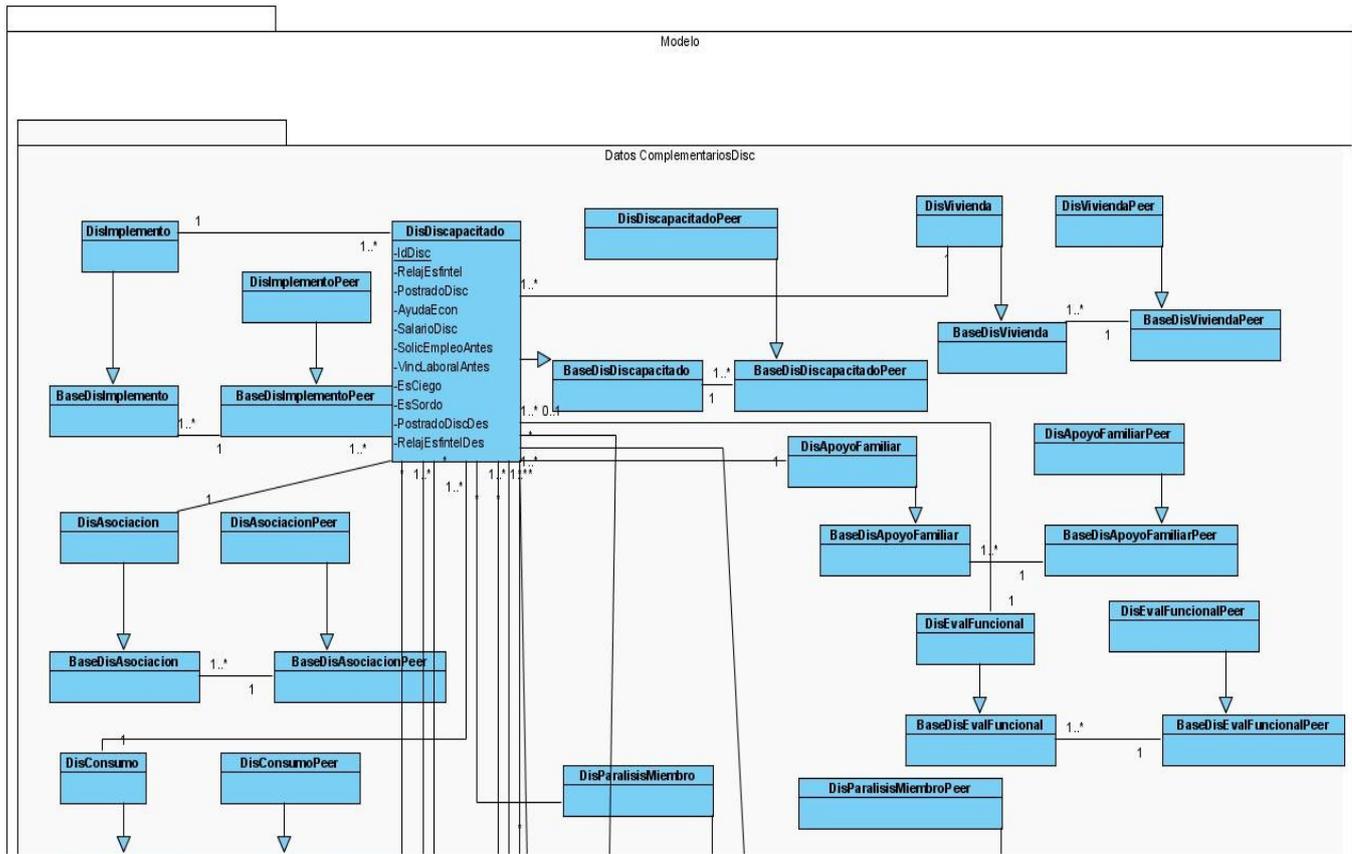
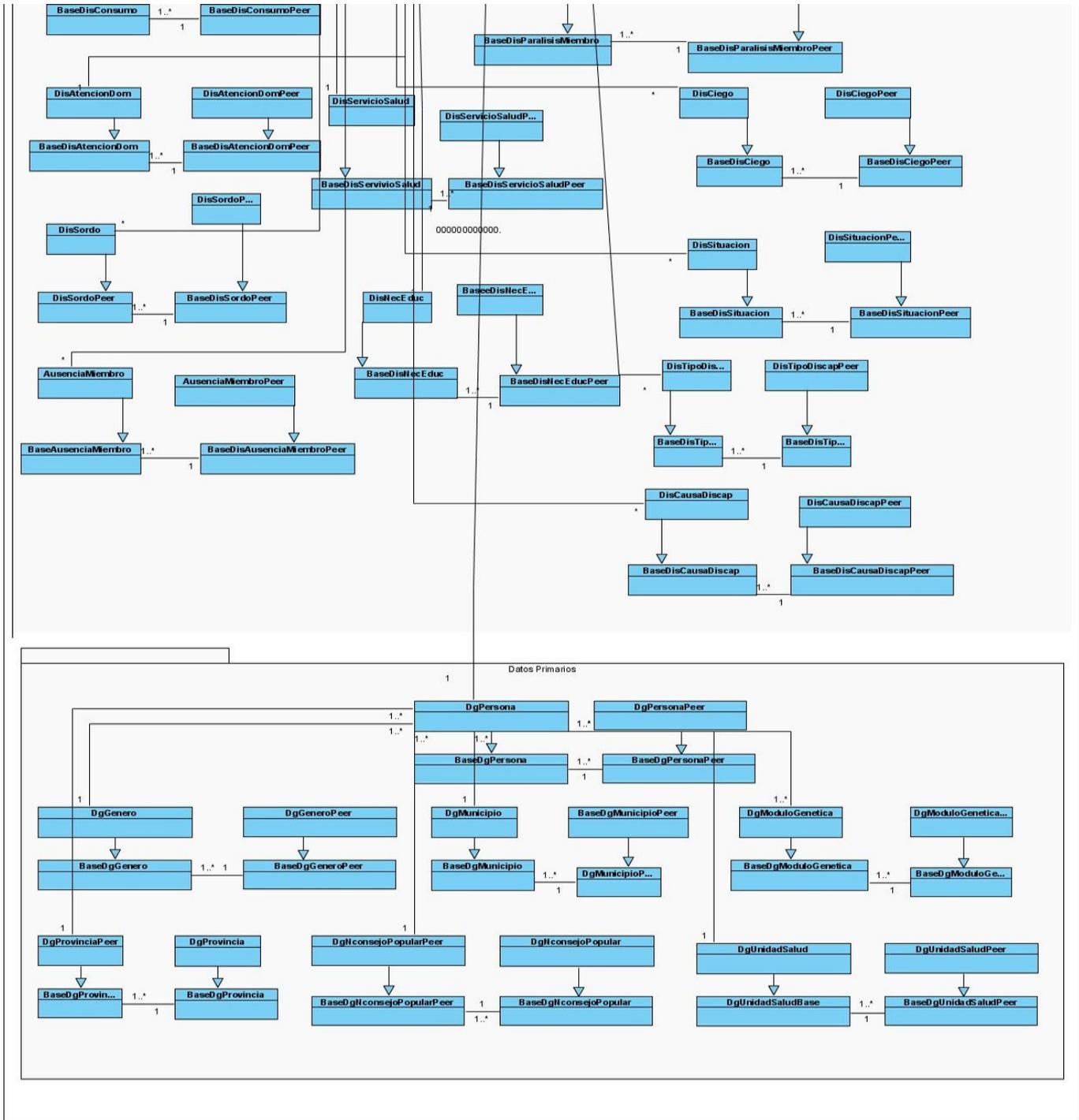


DIAGRAMA DE CLASES DEL DISEÑO. PAQUETE MODELO. Segunda parte.



2.1.7 Diagrama de interacción.

DIAGRAMA DE SECUENCIA. INSERTAR DATOS COMPLEMENTARIOS. Escenario 1

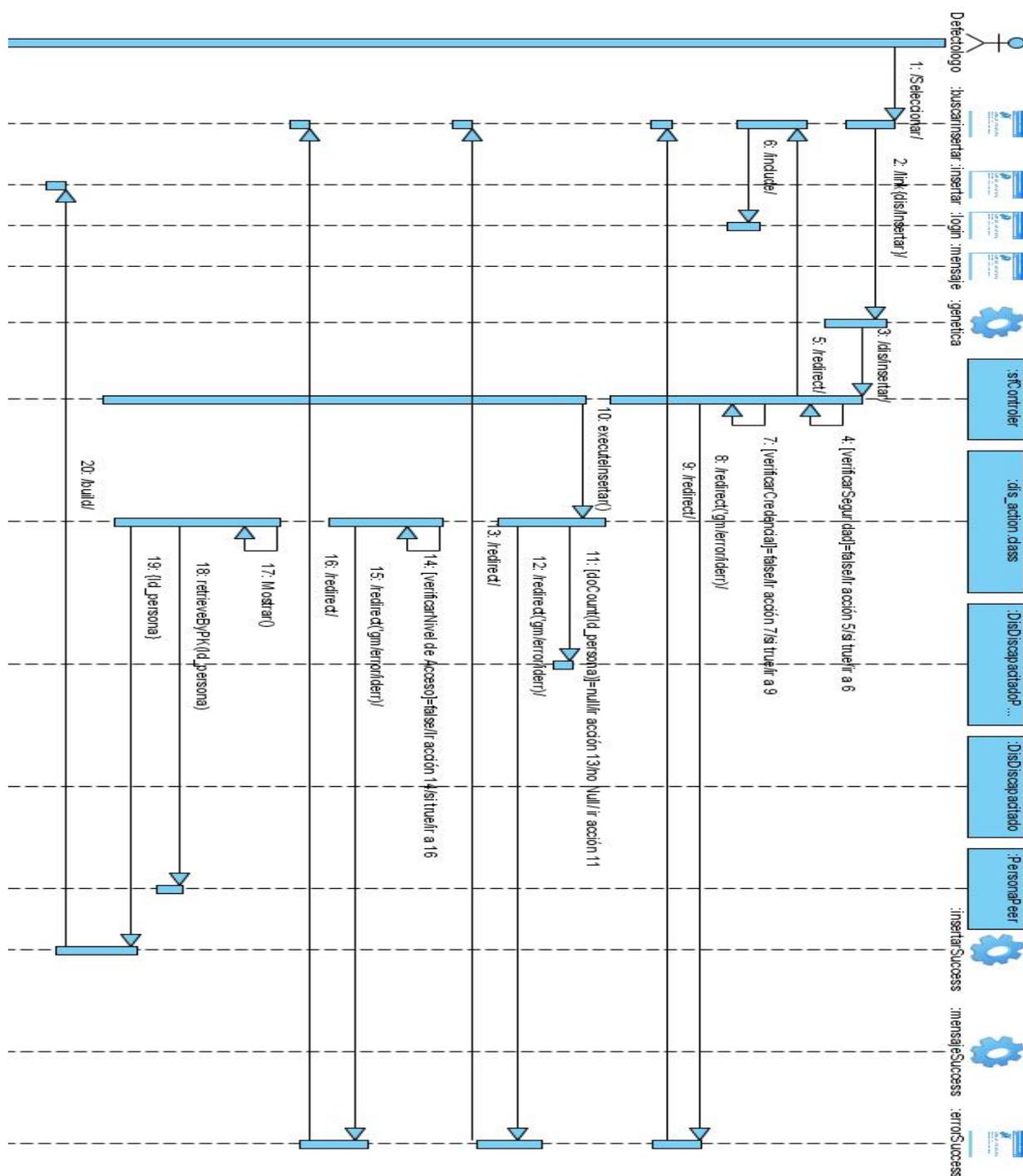


DIAGRAMA DE SECUENCIA. INSERTAR DATOS COMPLEMENTARIOS. Escenario 2

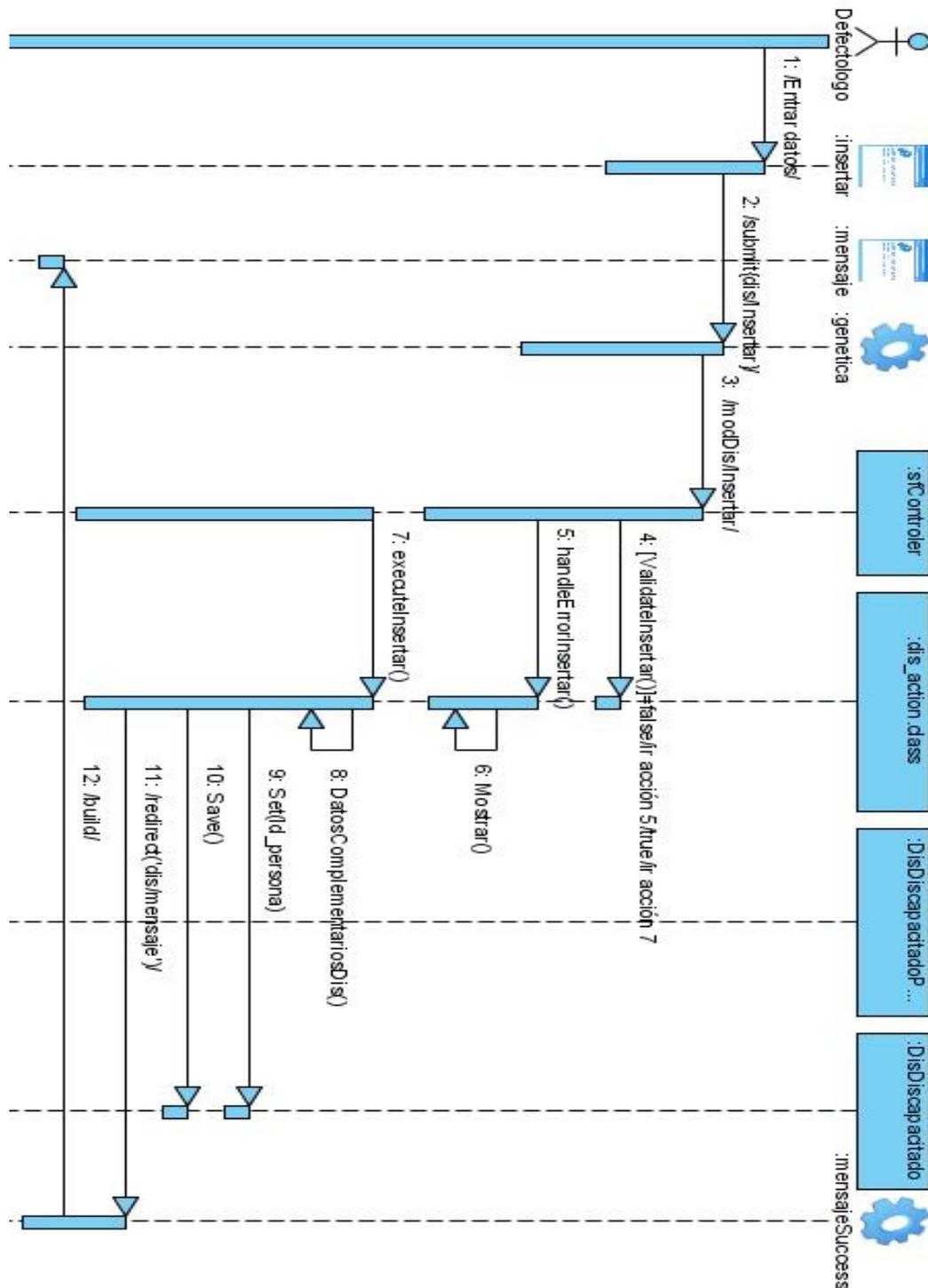


DIAGRAMA DE SECUENCIA. MODIFICAR DISCAPACITADO. Escenario 1

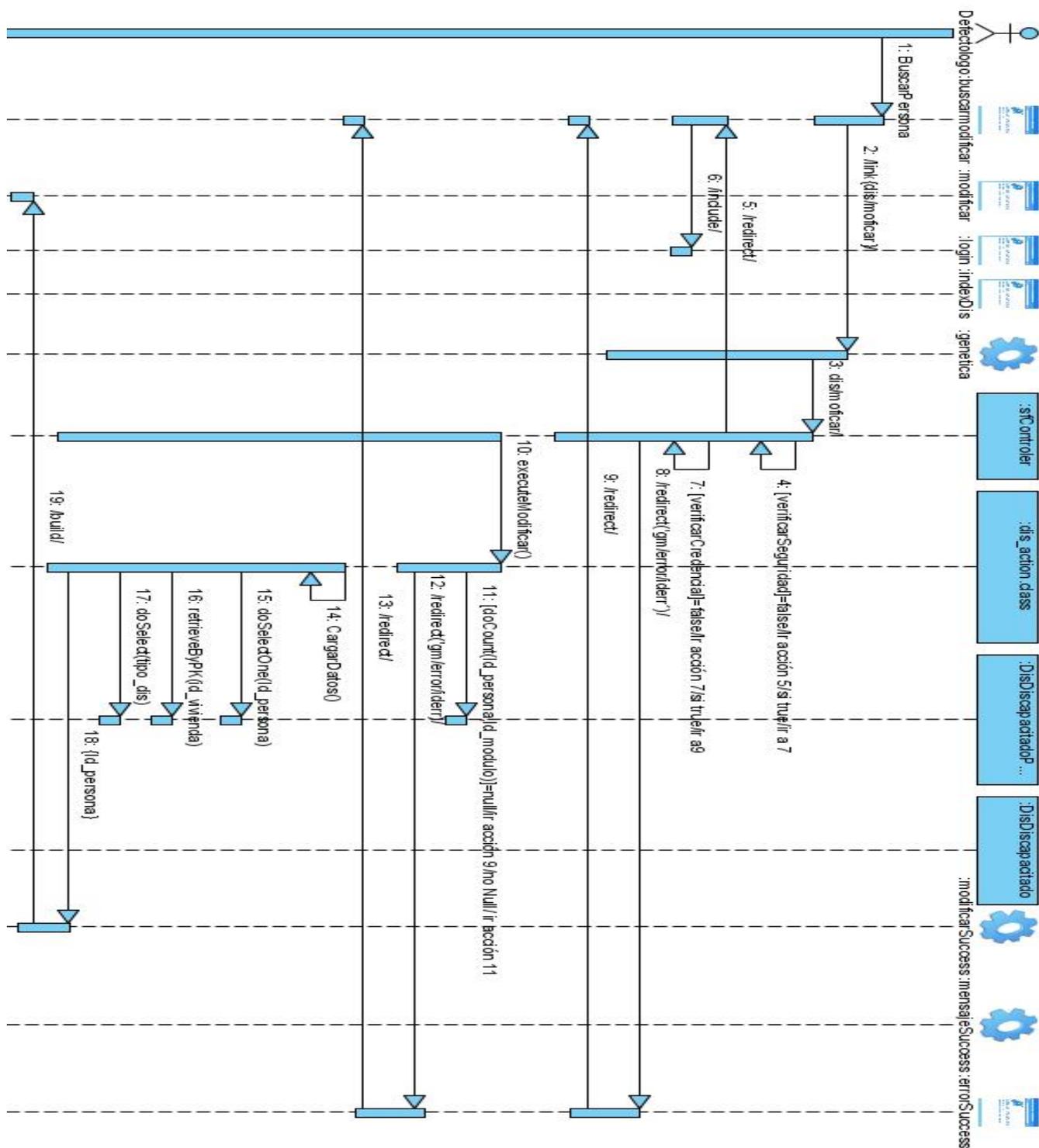


DIAGRAMA DE SECUENCIA. MODIFICAR DISCAPACITADO. Escenario 2

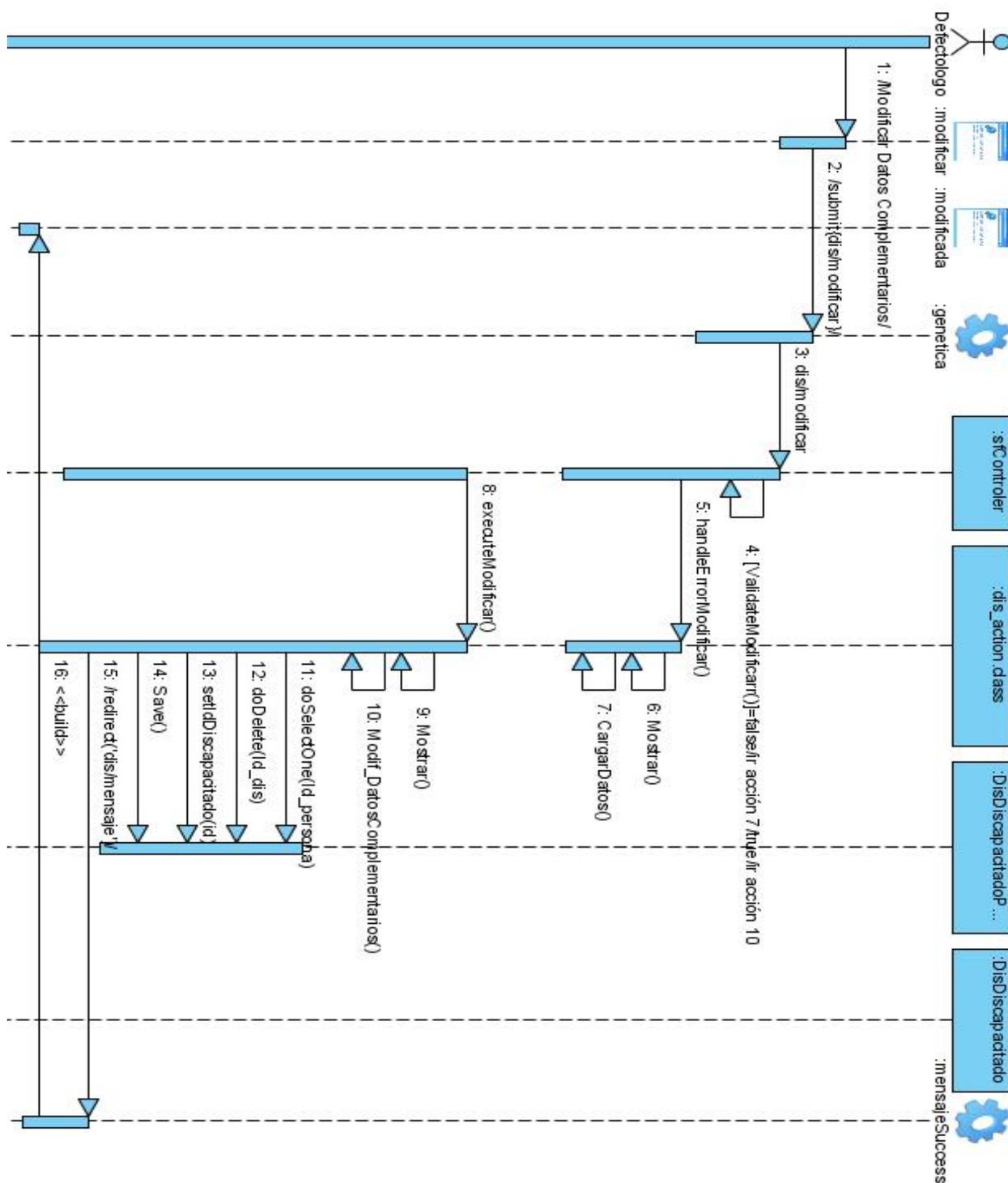
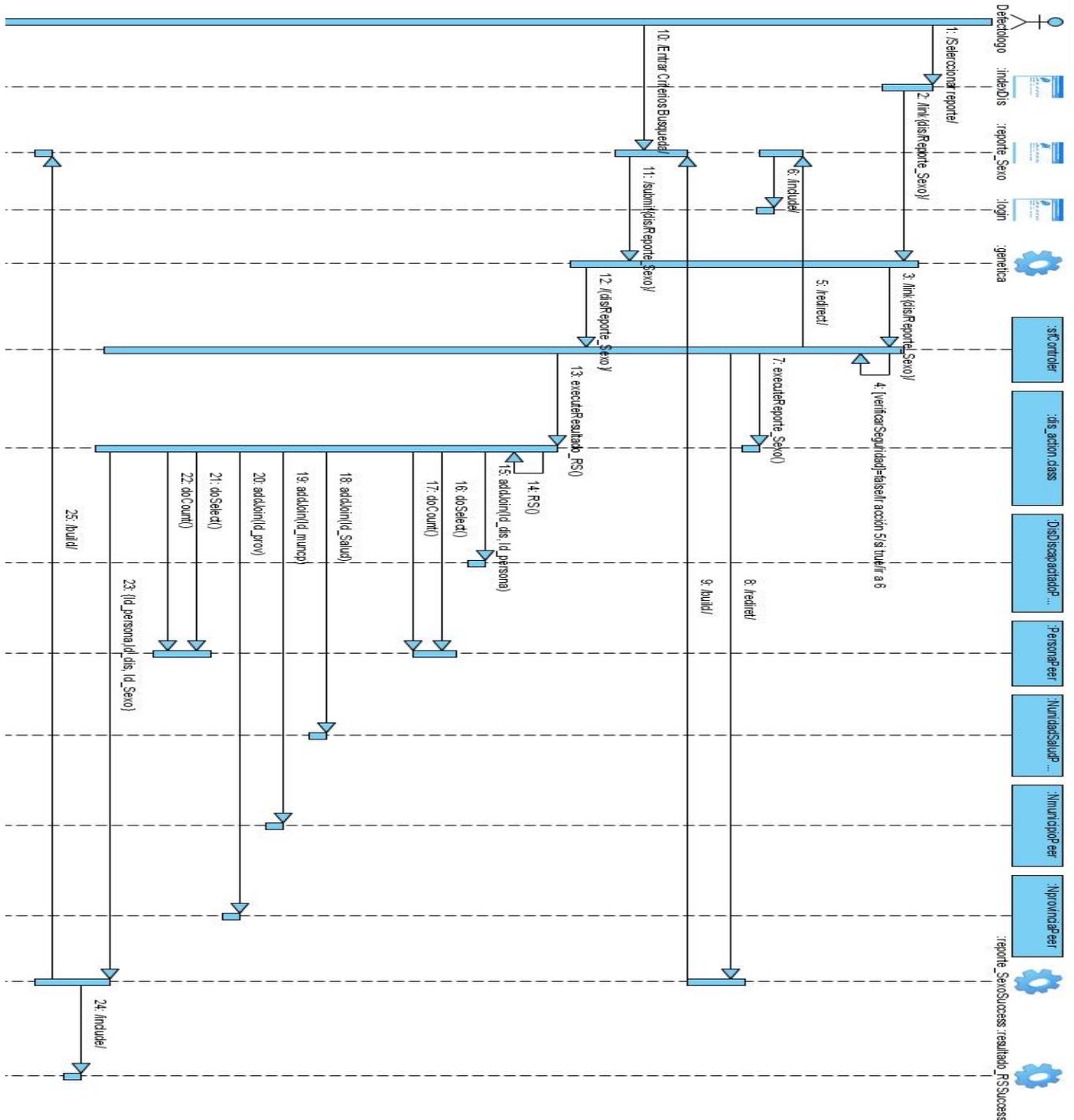


DIAGRAMA DE SECUENCIA. MOSTAR REPORTES ESTADÍSTICOS. Determinar la cantidad de discapacitados según el sexo.

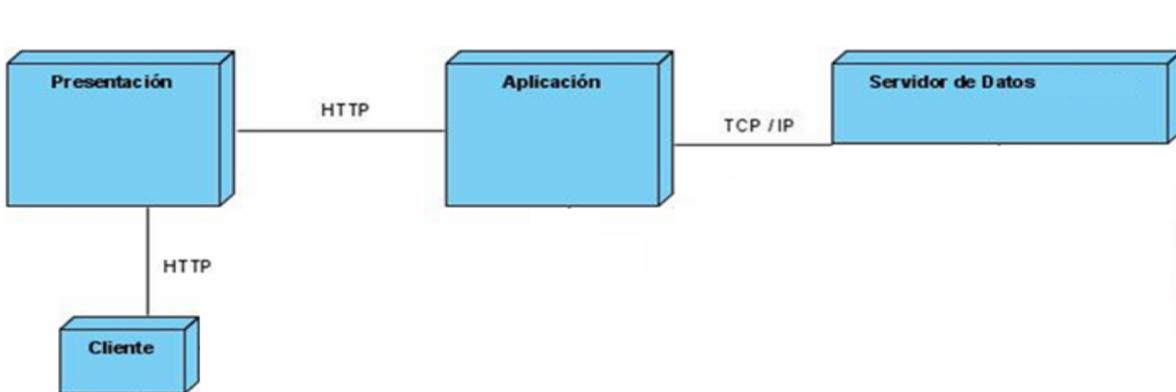


El comportamiento de los eventos en todos los reportes estadísticos es el mismo, diferenciándose solamente en las tablas que se consultan en la base de datos y el nombre de la plantilla Success que crea la página cliente.

2.2.6 Diagrama de Despliegue.

Configuración de nodos físicos sobre los cuales se desplegará el módulo como parte del SIGM, según la definición realizada.

DIAGRAMA DE DESPLIEGUE DEL SISTEMA.



2.2.7 Seguridad

Se garantiza la seguridad mediante la autenticación y la gestión de credenciales, simplificando la creación de secciones restringidas y la gestión de la seguridad de usuario, ya que antes de ser ejecutada cada acción pasa por un filtro especial que verifica si el usuario tiene privilegios de acceder a la misma. El controlador frontal se encarga del manejo de la seguridad, siendo un punto de entrada único para toda la aplicación.

Esos privilegios de usuarios son:

- 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" donde se especifican los requerimientos de seguridad que los usuarios deberán satisfacer para cada acción o para todas.

Por ejemplo:

- Si el usuario está autenticado y tiene las credenciales apropiadas, entonces la acción se ejecuta.
- Si el usuario no está autenticado, es redireccionado a la acción de login.
- Si el usuario está autenticado, pero no posee las credenciales apropiadas (es decir, que no están definidas en el security.yml asociado a la página en cuestión), será redirigido a la acción segura por defecto.

2.2.8 Pautas del diseño.

Se utilizarán las pautas de diseño definidas para el SIGM.

Los formularios que sean creados deben estar centrados y tener bordes con valores 1

La primera fila del formulario debe contener el nombre de la operación que se pretende realizar con el formulario. El estilo a utilizar para dicha fila es " hOperation".

Por ejemplo si el formulario tuviera la finalidad de buscar un estudiante, el encabezado sería el siguiente:

```
<td class="hOperation">Buscar Estudiante </td>
```

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. El estilo a utilizar en la columna derecha es “nombrecampo”, el de la izquierda “valorcampo” y el de los componentes “entradaplana”

Por ejemplo, se le solicita al usuario su nombre.

```
<tr>
  <td align="right" class="nombrecamp5o">Nombre</td>
  <td class="valorcampo">
    <input t class="entradaplana" type="text" name="textfield" />
  </td>
</tr>
```

Los botones para efectuar operaciones sobre el formulario se ubicaran en la parte inferior derecha del formulario y utilizarán el estilo “sbbtn”. Se posicionarán de izquierda a derecha teniendo en cuenta el peso de la operación que representan.

Ejemplo un formulario que los botones de las operaciones son enviar y cancelar.

```
<tr align="right">
  <td>
    <!-- Botón enviar se ubica más a la derecha que el botón cancelar -->
    <input name="Enviar" type="button" class="sbbtn" value="Enviar" />
    <input name="Cancelar" type="button" class="sbbtn" value="Cancelar" />
  </td>
</tr>
```

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

Por ejemplo el campo nombre debe ser obligatorio

```
<tr>
  <td class="nombrecampo" width="40%" height="20" align="right">nombre</td>
  <td class="valorcampo">
```

```
<input class="entradaplana" type="text" name="textfield" />  
<span class="entradaobligatoria" *</span>  
</td>  
</tr>
```

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.

2.2.9 Prototipo de Interfaz

Para el SIGM se elaboró un prototipo de interfaz, planificando un diseño orientado a las necesidades de sus futuros usuarios y al ambiente en el que será desplegado. Se ha tenido en cuenta el orden de los elementos que aparecen en el instrumento, acorde al orden de prioridad de los elementos que lo componen.

Descripción de los elementos que componen las interfaces del prototipo diseñado para el SIGM.



Sección 1:

Menú de Operaciones: presenta todas las operaciones correspondientes a cada modulo clasificadas en categorías.

Sección 2:

Área de visualización de datos: presenta la información correspondiente a cada operación invocada en el Menú de Operaciones.

2.2.10 Validación del sistema.

El sistema contará con la validación de entradas de datos, mediante el uso de archivos de extensión yml que son ejecutados cuando se llama la acción correspondiente a la plantilla que se ejecuta en un momento dado. Una vez realizado el envío de datos y detectado un error sea por dejar vacíos campos obligatorios o caracteres no adecuados, la aplicación mostrará una página de error, indicando específicamente cuál es el error y permitiendo al usuario la nueva entrada de los campos señalados.

El uso de JavaScript permitirá que se visibilicen o no diferentes categorías en la entrada de datos permitiendo que el usuario introduzca los datos con mayor precisión.

2.2.11 Conclusiones

Mediante la modelación del diseño del sistema se describe la estructura que deriva la implementación. Se obtuvieron los diagramas de clases del diseño, diagramas de secuencia, así como el prototipo de interfaz y la distribución del sistema mediante el modelo de despliegue. Se definieron los patrones de diseño utilizados. Quedó conformado el diseño del sistema como base fundamental para la implementación.

Capítulo 3: Implementación

En este capítulo se muestran las características de la implementación del sistema en términos de componentes, manteniendo como objetivo principal desarrollar la arquitectura acorde al diseño realizado. Se analizan y aplican los estilos de codificación definidos para el SIGM, se muestra el código fuente de la clase principal así como interfaces de la aplicación. Se incluye la validación a nivel de desarrollador.

3.1 Diagrama de Componentes.

DIAGRAMA DE COMPONENTES. CUS Insertar Datos Complementarios.

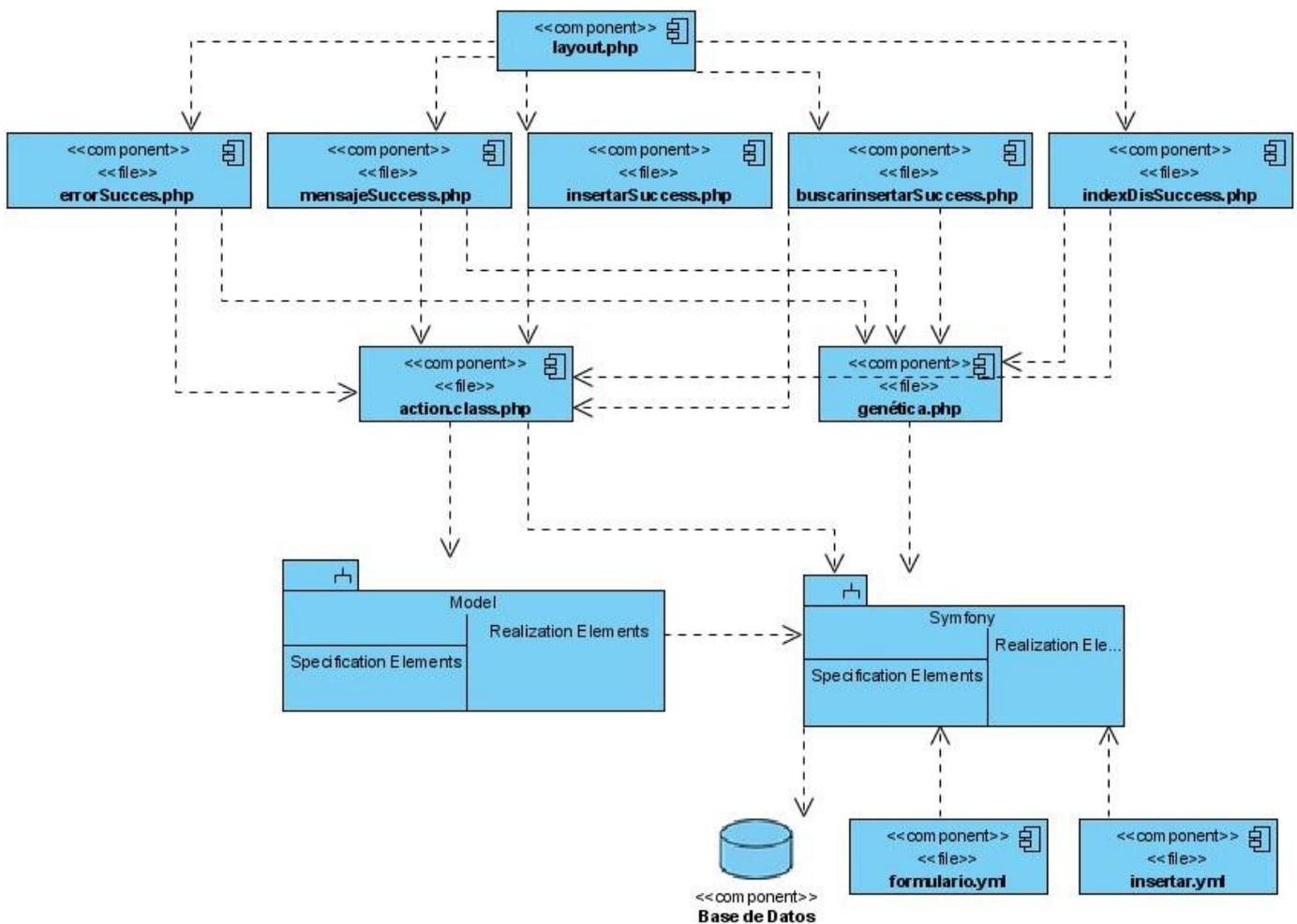


DIAGRAMA DE COMPONENTES. Modificar Datos Complementarios.

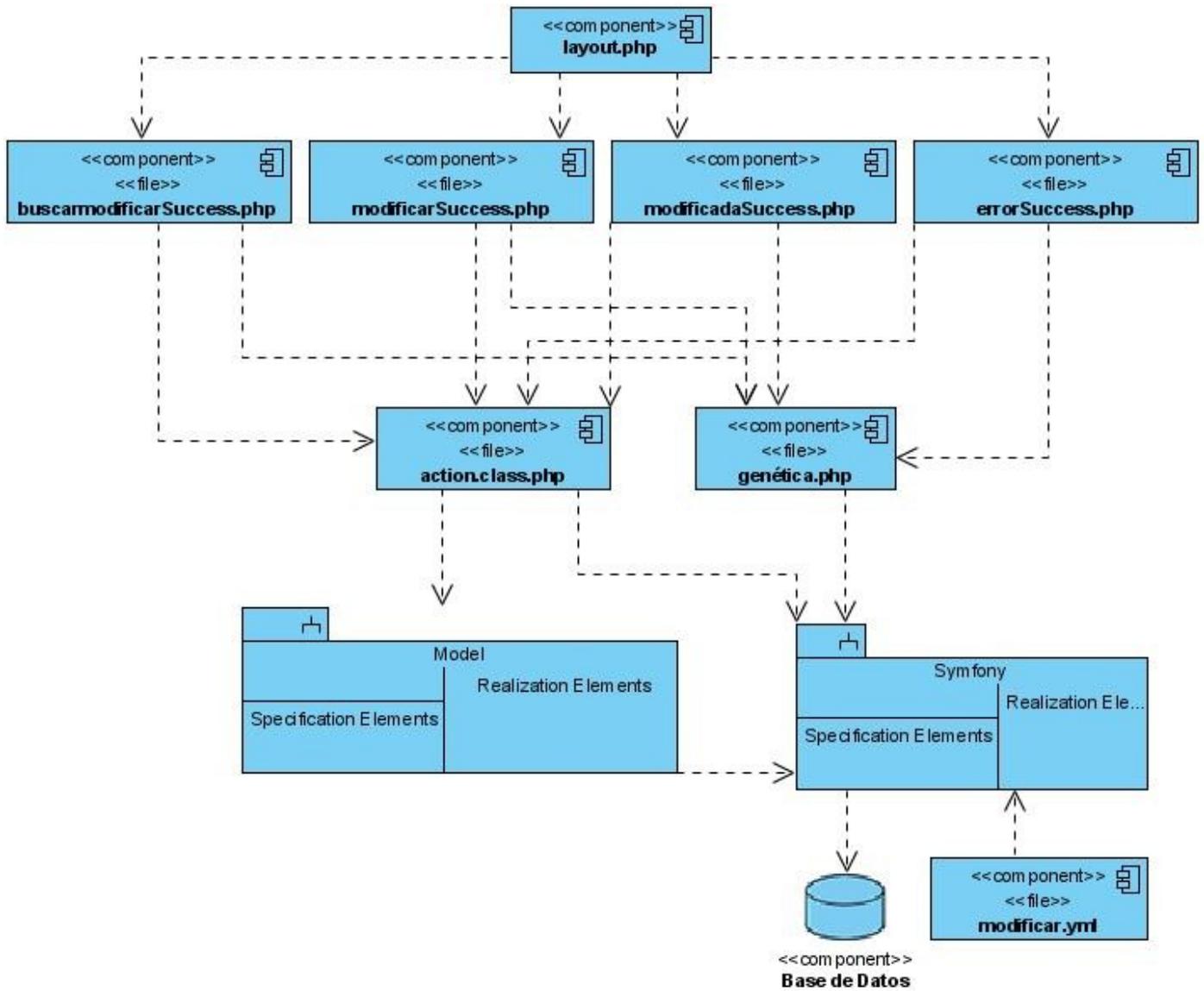
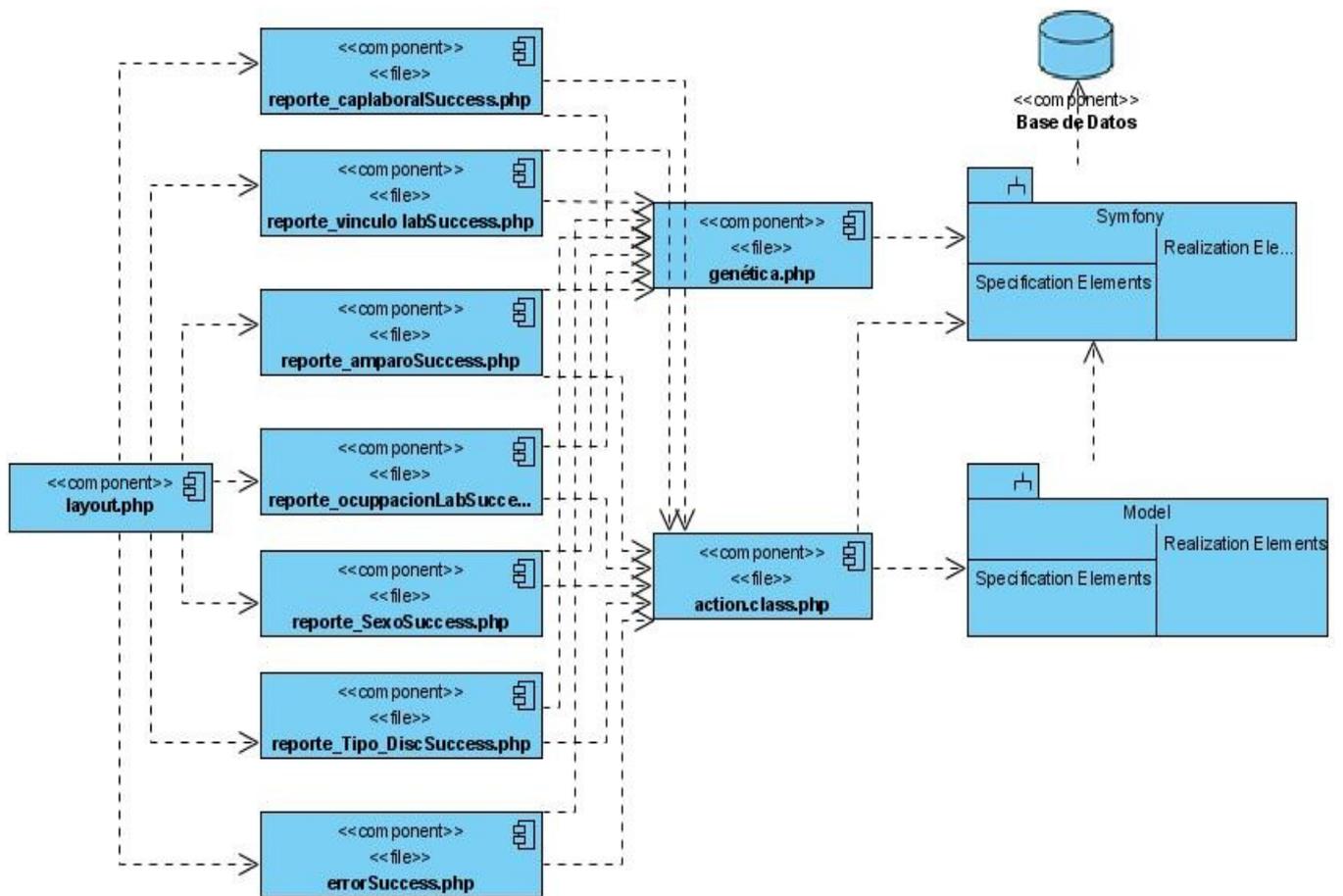


DIAGRAMA DE COMPONENTES. Reportes Estadísticos.



3.2 Estilos de codificación definidos para el SIGM.

Para la codificación dentro de la implementación de los componentes, se definen los siguientes estilos:

Todas las etiquetas php deben ser completas (<?php ?>)... no reducidas (<? ?>).

Todas las variables deberían ser inicializadas o, al menos, comprobada su existencia utilizando isset() antes de ser utilizadas.

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).

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. Ejemplos:

\$edad

\$fechaNacimiento

\$personas

Los bloques de código siempre deben estar encerrados por llaves (incluso si solo constan de una línea).

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

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. Ejemplo:

```
/** Esta funcion tiene un objetivo determinado...
```

```
* @author pepito
```

```
* @param $edad edad de la persona
```

```
* @param $sexo sexo de la persona
```

```
*/
```

```
private function MostrarDatos($edad,$sexo)
```

```
{
```

```
}

```

3.3 Código fuente de las principales clases.

La clase `disActions` constituye una clase controladora puesto que contiene todas las funcionalidades que son asociadas a cada plantilla, a continuación se muestran los métodos que la componen conjuntamente con su descripción.

```

* dis actions
class disActions extends sfActions
{
    /**
     * Executes index action
     */

    //Pagina Principal de Discapacitados
    public function executeIndexDis()

        //Mostrar los datos primarios en las paginas de Insetar y Modificar
    private function Mostrar()

    //Cargar los datos del discapacitado en la página Modificar
    private function CargarDatos()

    //Objetos,Set() y Save() del Insertar
    public function DatosComplementariosDis()

    public function executeBuscarinsertar()

    public function executeInsertar()

    public function validateInsertar()

    public function handleErrorInsertar()

    //Pagina para mostrarle al usuario un mensaje de que ha insertado los datos
correctamente

    public function executeMensaje()

    public function executeBuscarmodificar()
    
```

```

//Set() y Save(), para Modificar

public function Modif_DatosComplementarios()

public function executeModificar()

public function validateModificar()

public function handleErrorModificar()

//Pagina para mostrarle al usuario un mensaje de que se han modificado los
datos correctamente

public function executeModificada()

public function Reporte_Sexo()

public function executeReporte_Sexov()

public function handleErrorReporte_Sexov()

public function validateReporte_Sexov()

//Reporte por Tipo de Discapacidad

public function RT()

public function executeReporte_Tipo_Disc()

public function executeResultado_RT()

public function validateReporte_Tipo_Discp()

public function handleErrorReporte_Tipo_Discp()

public function Reporte_Amparo()

public function executeReporte_amparo()

public function validateReporte_amparo()

public function handleErrorReporte_amparo()

public function Reporte_CapacidadLaboral()

public function executeReporte_caplaboral()

public function validateReporte_caplaboral()

public function handleErrorReporte_caplaboral()

//Reporte por Sexo
    
```

```

public function executeReporte_Sexo()

public function RS()

public function executeResultado_RS()

}
    
```

Clase DisDiscapitadoPeer, generada en la abstracción de datos.

Hereda de la clase BaseDisDiscapitadoPeer.

```

class DisDiscapitado extends BaseDisDiscapitado
{
}
    
```

Clase BaseDisDiscapitadoPeer, generada en la abstracción de datos.

Define los métodos para la consulta y actualización a la base de datos.

```

abstract class BaseDisDiscapitadoPeer {

    const DATABASE_NAME = 'propel';

    const TABLE_NAME = 'dis_discapitado';

    const CLASS_DEFAULT = 'lib.model.DisDiscapitado';

    const NUM_COLUMNS = 19;

    const NUM_LAZY_LOAD_COLUMNS = 0;

    const ID_DISCAPITADO = 'dis_discapitado.ID_DISCAPITADO';

    const ID_PERSONA = 'dis_discapitado.ID_PERSONA';

    const ID_VIVIENDA = 'dis_discapitado.ID_VIVIENDA';
}
    
```

```

const CONSUMO = 'dis_discapitado.CONSUMO';

const ESCOLARIDAD = 'dis_discapitado.ESCOLARIDAD';

const SOLICEMPLEOANTES = 'dis_discapitado.SOLICEMPLEOANTES';

const CAPACIDAD_LABORAL = 'dis_discapitado.CAPACIDAD_LABORAL';

const POSTRADO_DISC = 'dis_discapitado.POSTRADO_DISC';

const RELAJ_ESFINTEL = 'dis_discapitado.RELAJ_ESFINTEL';

const AYUDA_ECON = 'dis_discapitado.AYUDA_ECON';

const SALARIO_DISC = 'dis_discapitado.SALARIO_DISC';

const OBSERVACION = 'dis_discapitado.OBSERVACION';

const AMPARO_FILIAR_DISC = 'dis_discapitado.AMPARO_FILIAR_DISC';

const VINCLABORALANTES = 'dis_discapitado.VINCLABORALANTES';

const EVALUACION_FUNCIONAL = 'dis_discapitado.EVALUACION_FUNCIONAL';

const DESEOS_ASOCIARSE = 'dis_discapitado.DESEOS_ASOCIARSE';

const COND_VIDA_RESTO_FMLIA = 'dis_discapitado.COND_VIDA_RESTO_FMLIA';

const OCUPACION = 'dis_discapitado.OCUPACION';

const CMF = 'dis_discapitado.CMF';

private static $phpNameMap = null;
public static function doCount(Criteria $criteria, $distinct = false, $con =
null)
    
```

```

public static function doSelectOne(Criteria $criteria, $con = null)

public static function doSelect(Criteria $criteria, $con = null)

public static function doSelectRS(Criteria $criteria, $con = null)

public static function populateObjects(ResultSet $rs)

public static function doCountJoinDgPersona(Criteria $criteria, $distinct =
false, $con = null)

        public static function doCountJoinDisVivienda(Criteria $criteria,
$distinct = false, $con = null)

        public static function doSelectJoinAllExceptDgPersona(Criteria $c, $con =
null)

        public static function doSelectJoinAllExceptDisVivienda(Criteria $c, $con =
null)

public static function getTableMap()

public static function getOMClass()

public static function doInsert($values, $con = null)

public static function doUpdate($values, $con = null)

public static function doDeleteAll($con = null)

public static function doDelete($values, $con = null)

public static function doValidate(DisDiscapitado $obj, $cols = null)

public static function retrieveByPK($pk, $con = null)

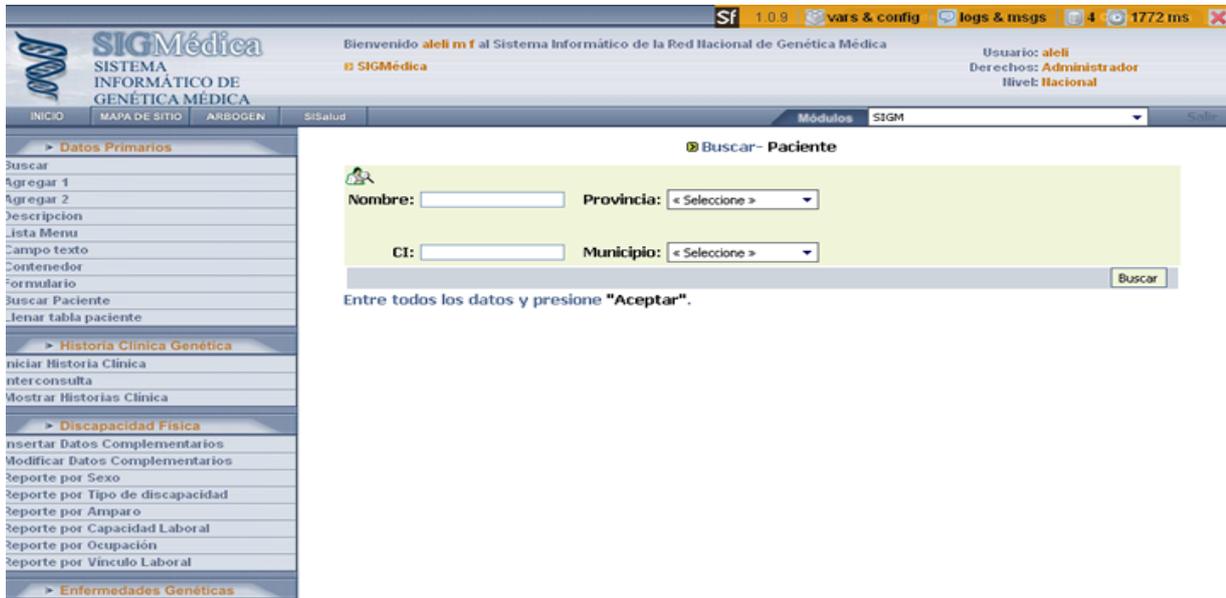
public static function retrieveByPKs($pks, $con = null)
}
if (Propel::isInit()) {
    try {
        BaseDisDiscapitadoPeer::getMapBuilder();
    } catch (Exception $e) {
        Propel::log('Could not initialize Peer: ' . $e->getMessage(),
Propel::LOG_ERR);
    }
} else {
    require_once 'lib/model/map/DisDiscapitadoMapBuilder.php';
    Propel::registerMapBuilder('lib.model.map.DisDiscapitadoMapBuilder');
}
    
```

3.4 Interfaces de la aplicación.

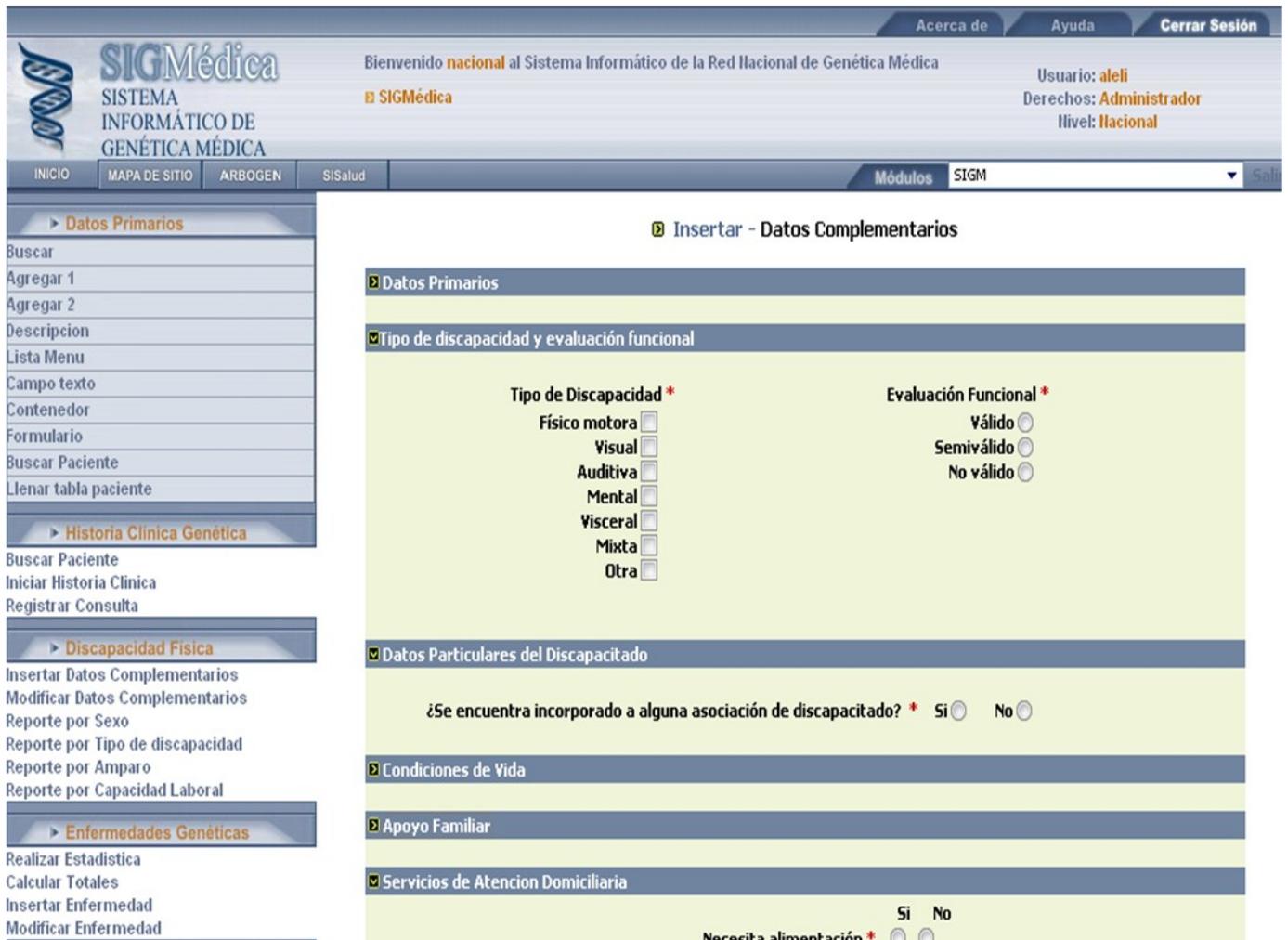
Interfaz de entrada al sistema informático.



Interfaz de búsqueda de datos primarios de la persona.



Interfaz del sistema informático para el caso de uso Insertar Datos Complementarios.



Insertar - Datos Complementarios

Datos Primarios

Tipo de discapacidad y evaluación funcional

<p>Tipo de Discapacidad *</p> <p>Físico motora <input type="checkbox"/></p> <p>Visual <input type="checkbox"/></p> <p>Auditiva <input type="checkbox"/></p> <p>Mental <input type="checkbox"/></p> <p>Visceral <input type="checkbox"/></p> <p>Mixta <input type="checkbox"/></p> <p>Otra <input type="checkbox"/></p>	<p>Evaluación Funcional *</p> <p>Válido <input type="radio"/></p> <p>Semiválido <input type="radio"/></p> <p>No válido <input type="radio"/></p>
---	---

Datos Particulares del Discapitado

¿Se encuentra incorporado a alguna asociación de discapacitado? * Si No

Condiciones de Vida

Apoyo Familiar

Servicios de Atención Domiciliaria

Necesita alimentación * Si No

Insertar Datos Complementarios
 Modificar Datos Complementarios
 Reporte por Sexo
 Reporte por Tipo de discapacidad
 Reporte por Amparo
 Reporte por Capacidad Laboral

► **Enfermedades Genéticas**

Realizar Estadística
 Calcular Totales
 Insertar Enfermedad
 Modificar Enfermedad

► **Gemelo**

Insertar pareja de gemelos
 Actualizar pareja de gemelos
 Reporte estadístico
 Reporte avanzado

► **Malformaciones Congénitas**

Modificar Datos de Pacientes
 Insertar Datos Complementarios
 Insertar Caso Interrupcion
 Buscar Planilla
 Prevalencia al Ilacimiento
 Factores de riesgo y Defectos Congénitos

► **Discapacidad Intelectual**

Insertar Datos Complementarios
 Modificar Datos Complementarios

► **Teleconsulta Genética**

Nueva Solicitud
 Casos para Hoy

Estado de salud y necesidad de implementos, medicamentos y ayudas técnicas

Necesidad de servicios de salud

Necesidad Educacional(Menores de 18 años)

Causa referida de la discapacidad

Clasificación en relación con el consumo de bebidas alcohólicas del paciente y familiares

Observaciones

(Datos significativos brindados por el individuo que respondió la encuesta, no incluidos en la misma, así como alguna observación relevante que el encuestador considere oportuno comunicar)



advanced.path: p

(*) Los campos señalados son de entrada obligatoria

Aceptar

Cancelar

Interfaz del sistema informático para el caso de uso Modificar Datos Complementarios.

Datos Primarios

Buscar

Agregar 1

Agregar 2

Descripcion

Lista Menu

Campo texto

Contenedor

Formulario

Buscar Paciente

Llenar tabla paciente

Historia Clínica Genética

Buscar Paciente

Iniciar Historia Clínica

Registrar Consulta

Discapacidad Física

Insertar Datos Complementarios

Modificar Datos Complementarios

Reporte por Sexo

Reporte por Tipo de discapacidad

Reporte por Amparo

Reporte por Capacidad Laboral

Enfermedades Genéticas

Realizar Estadística

Calcular Totales

Insertar Enfermedad

Modificar Enfermedad

Gemelo

Insertar pareja de gemelos

Actualizar pareja de gemelos

Reporte estadístico

Reporte avanzado

Malformaciones Congénitas

Modificar - Datos Complementarios

Datos Primarios

Nombre: <u>franklis</u>	Fecha de nacimiento: <u>05/14/08</u>
Primer Apellido: <u>cantillo</u>	Edad: <u>45645</u>
Segundo Apellido: <u>ramirez</u>	Sexo: <u>Intersexo</u>
Estado Civil: <u>Soltero</u>	No. C.I.: <u>44564564564</u>
Consejo Popular: <u>Las Vegas</u>	Piel: <u>Roja</u>
Area de salud: <u>Hospital 5 de sept</u>	
Dirección: <u>Calle 5 entre 5 Apto 5</u>	Escolaridad: <u>Sin escolarizar</u> ▼*

Tipo de discapacidad y evaluación funcional

Tipo de discapacidad *	Evaluación funcional *
Físico motora <input checked="" type="checkbox"/>	Válido <input type="radio"/>
Visual <input checked="" type="checkbox"/>	Semiválido <input checked="" type="radio"/>
Auditiva <input type="checkbox"/>	No válido <input type="radio"/>
Mental <input type="checkbox"/>	
Visceral <input type="checkbox"/>	
Mixta <input type="checkbox"/>	
Otra <input type="checkbox"/>	

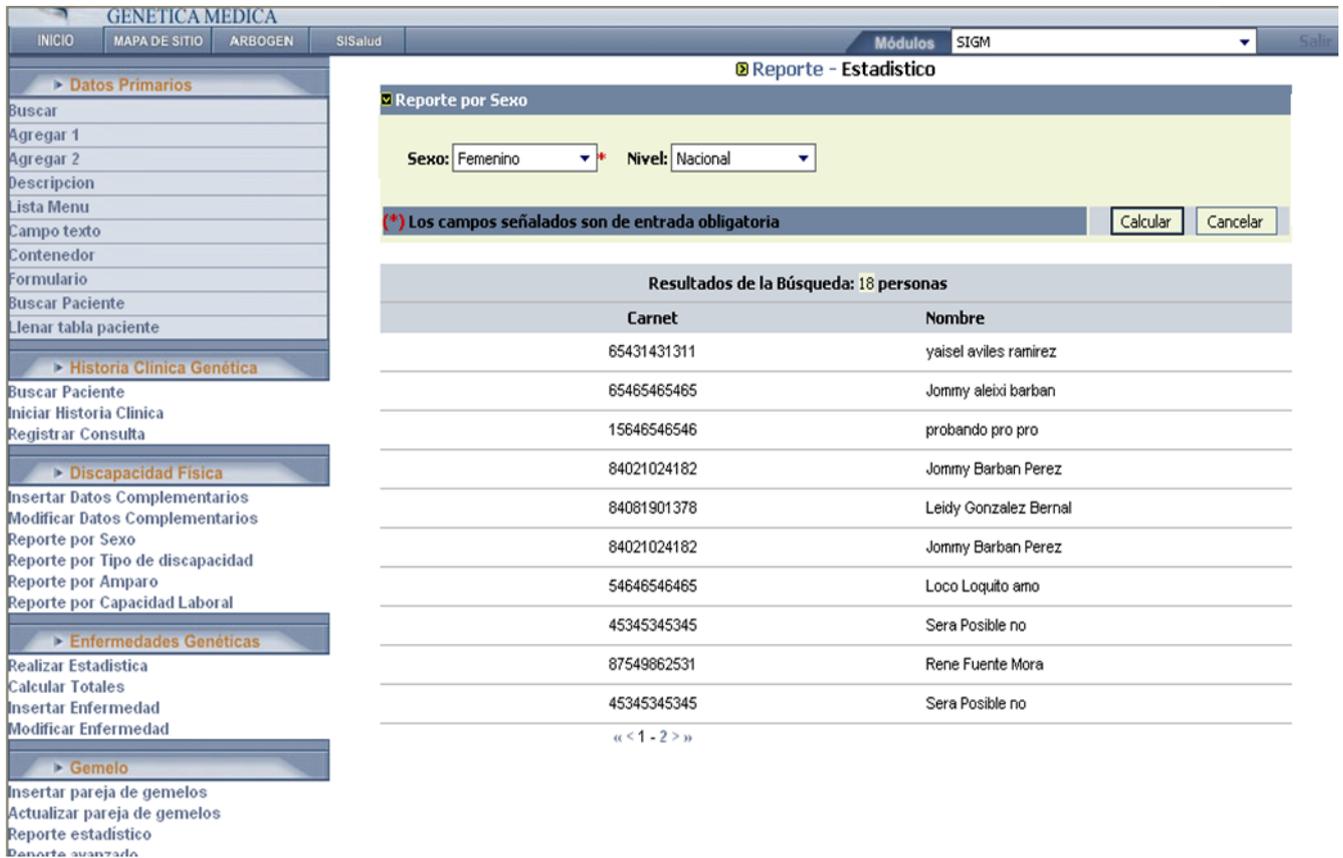
¿Tiene ausencias de miembros?	Sup. Der. <input type="checkbox"/>	Sup. Izq. <input type="checkbox"/>	Inf. Der. <input type="checkbox"/>	Inf. Izq. <input type="checkbox"/>
¿Tiene Parálisis de miembros?	Sup. Der. <input type="checkbox"/>	Sup. Izq. <input type="checkbox"/>	Inf. Der. <input type="checkbox"/>	Inf. Izq. <input type="checkbox"/>
¿Es Ciego?	Ojo Der. <input type="checkbox"/>	Ojo Izq. <input type="checkbox"/>		
¿Es Débil Visual?	Ojo Der. <input checked="" type="checkbox"/>	Ojo Izq. <input checked="" type="checkbox"/>		

Datos Particulares del Discapacitado

- Agregar 2
- Descripcion
- Lista Menu
- Campo texto
- Contenedor
- Formulario
- Buscar Paciente
- Llenar tabla paciente
- > Historia Clínica Genética**
- Buscar Paciente
- Iniciar Historia Clínica
- Registrar Consulta
- > Discapacidad Física**
- Insertar Datos Complementarios
- Modificar Datos Complementarios
- Reporte por Sexo
- Reporte por Tipo de discapacidad
- Reporte por Amparo
- Reporte por Capacidad Laboral
- > Enfermedades Genéticas**
- Realizar Estadística
- Calcular Totales
- Insertar Enfermedad
- Modificar Enfermedad
- > Gemelo**
- Insertar pareja de gemelos
- Actualizar pareja de gemelos
- Reporte estadístico
- Reporte avanzado
- > Malformaciones Congénitas**
- Modificar Datos de Pacientes
- Insertar Datos Complementarios
- Insertar Caso Interrupcion
- Buscar Planilla
- Draculancia al Nacimiento

Tipo de discapacidad y evaluación funcional	
Datos Particulares del Discapitado	
Condiciones de Vida	
Apoyo Familiar	
Servicios de Atención Domiciliaria	
Situación Laboral	
Estado de salud y necesidad de implementos, medicamentos y ayudas técnicas	
Necesidad de servicios de salud	
	Si No
Necesita ingreso en Hospital Psiquiátrico de Larga Estadía *	<input checked="" type="radio"/> <input type="radio"/>
Se encuentra ingresado en Hospital Psiquiátrico de Larga Estadía *	<input checked="" type="radio"/> <input type="radio"/>
Necesita ingreso en Hogar de Ancianos *	<input checked="" type="radio"/> <input type="radio"/>
Se encuentra ingresado en Hogar de Anciano *	<input checked="" type="radio"/> <input type="radio"/>
Necesita ingreso en Centro Diurno *	<input checked="" type="radio"/> <input type="radio"/>
Se encuentra Seminternado en el Centro Diurno *	<input checked="" type="radio"/> <input type="radio"/>
Necesita atención especializada para su discapacidad *	<input checked="" type="radio"/> <input type="radio"/>
Esta recibiendo atención especializada para su discapacidad *	<input checked="" type="radio"/> <input type="radio"/>
Necesita servicio de rehabilitación física *	<input checked="" type="radio"/> <input type="radio"/>
Esta recibiendo el servicio de reavitiación física *	<input checked="" type="radio"/> <input type="radio"/>
Necesita servicio Estomatológicos *	<input checked="" type="radio"/> <input type="radio"/>
Esta recibiendo los servicios Estomatológicos *	<input checked="" type="radio"/> <input type="radio"/>
Ha necesitado el Servicio de atención de Urgencias *	<input checked="" type="radio"/> <input type="radio"/>
Ha recibido el Servicio de Urgencia sin dificultad *	<input checked="" type="radio"/> <input type="radio"/>
Necesidad Educativa(Menores de 18 años)	

Interfaz del sistema informático para el caso de uso Reporte Estadístico. Reporte por sexo.



GENETICA MEDICA

INICIO | MAPA DE SITIO | ARBOGEN | SISalud | Módulos | SIGM | Salir

Reporte - Estadístico

Reporte por Sexo

Sexo: * Nivel:

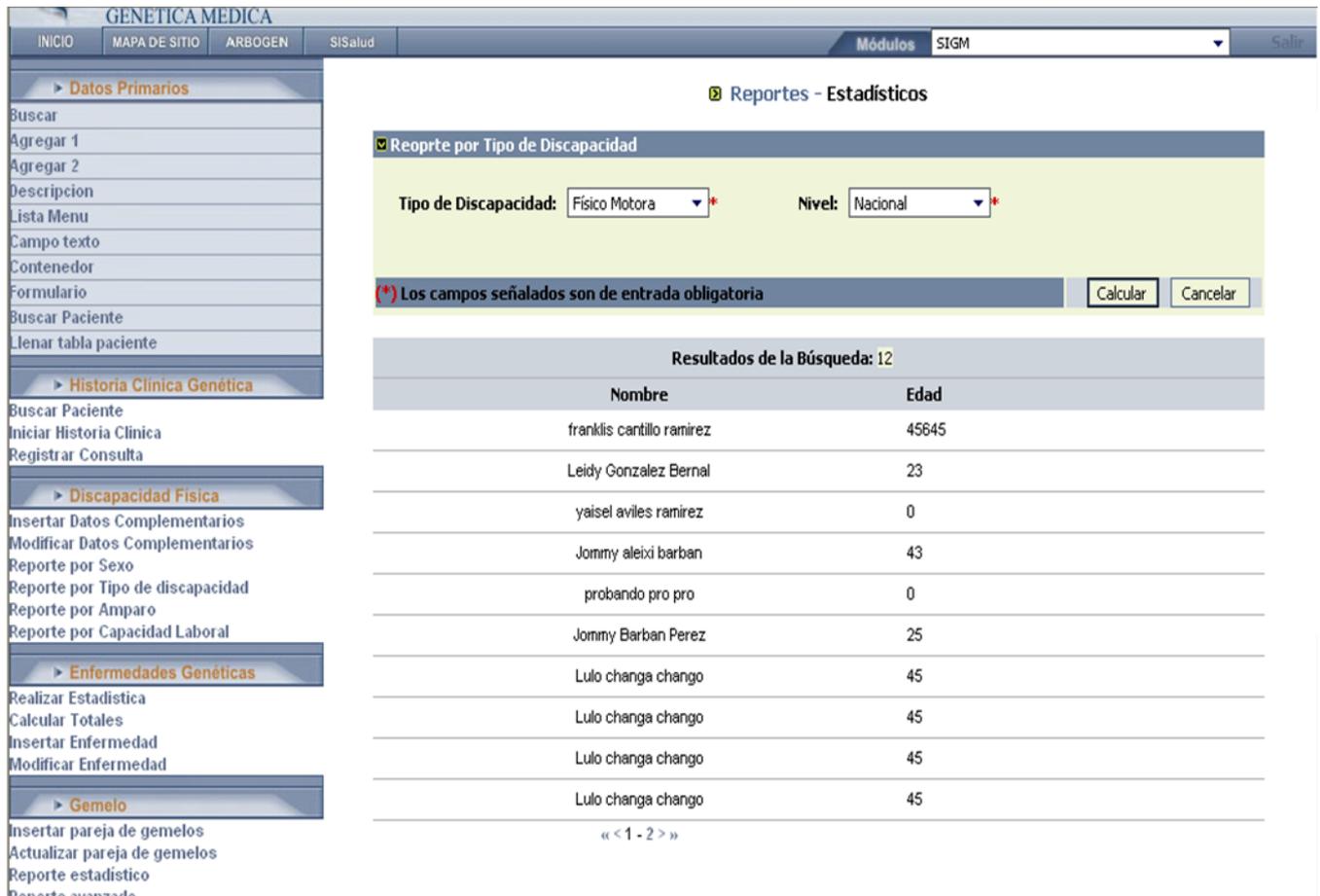
(*) Los campos señalados son de entrada obligatoria

Resultados de la Búsqueda: 18 personas

Carnet	Nombre
65431431311	yaisel aviles ramirez
65465465465	Jommy alexi barban
15646546546	probando pro pro
84021024182	Jommy Barban Perez
84081901378	Leidy Gonzalez Bernal
84021024182	Jommy Barban Perez
54646546465	Loco Loquito amo
45345345345	Sera Posible no
87549862531	Rene Fuente Mora
45345345345	Sera Posible no

« < 1 - 2 > »

Interfaz del sistema informático para el caso de uso Reporte Estadístico. Reporte tipo de discapacidad.



GENETICA MEDICA

INICIO | MAPA DE SITIO | ARBOGEN | SISalud | Módulos: SIGM | Salir

Reportes - Estadísticos

Reopрте por Tipo de Discapacidad

Tipo de Discapacidad: Físico Motora* Nivel: Nacional*

(*) Los campos señalados son de entrada obligatoria

Resultados de la Búsqueda: 12

Nombre	Edad
franklis cantillo ramirez	45645
Leidy Gonzalez Bernal	23
yaisel aviles ramirez	0
Jommy alexi barban	43
probando pro pro	0
Jommy Barban Perez	25
Lulo changa chango	45

« < 1 - 2 > »

Interfaz del sistema informático para el caso de uso Reporte Estadístico. Reporte amparo filial.

The screenshot shows the 'GENETICA MEDICA' application interface. The top navigation bar includes 'INICIO', 'MAPA DE SITIO', 'ARBOGEN', 'SISalud', 'Módulos', and 'SIGM'. The left sidebar contains a menu with categories: 'Datos Primarios', 'Historia Clínica Genética', 'Discapacidad Física', 'Enfermedades Genéticas', and 'Gemelo'. The main content area is titled 'Reportes - Estadísticos' and features a search form for 'Reporte por Amparo Filial'. The search criteria are 'Amparo Filial: Con Amparo' and 'Nivel: Nacional'. A message states: '(*) Los campos señalados son de entrada obligatoria'. Below the search form is a table of search results with 25 entries.

Resultados de la Búsqueda: 25		
Nombre	Edad	CI
franklis cantillo ramirez	45645	44564564564
Leidy Gonzalez Bernal	23	84081901378
yaisel aviles ramirez	0	65431431311
Jommy aleixi barban	43	65465465465
probando pro pro	0	15646546546
Jommy Barban Perez	25	84021024182
Jommy Barban Perez	25	84021024182
Loco Loquito amo	3	54646546465
Sera Posible no	4	45345345345
Rene Fuente Mora	24	87549862531

3.5 Validación del sistema a nivel de desarrollador

Durante la implementación, se le realizó a la aplicación web diferentes comprobaciones con el objetivo de probar que el Sistema cumple con sus funcionalidades específicas; se intentó encontrar fallos en la aplicación web, corregirlos y así lograr que las funciones del software sean operativas, que las entradas se acepten de forma adecuada, se produzca una salida correcta y que la integridad de la información externa se mantenga.

Las comprobaciones realizadas se centraron en lo que se espera del Sistema, se chequeó de forma periódica su comportamiento durante la ejecución, se verificó sistemáticamente que cada una de las funcionalidades necesarias para el cumplimiento de los requerimientos del sistema hicieran lo esperado, para lograr de ese modo satisfacer las necesidades del cliente, comprobándose además el funcionamiento correcto de las validaciones ante entradas de datos erróneos.

Se trataron las validaciones exhaustivamente debido a que el sistema informático fue desarrollado con el propósito fundamental de llevar a cabo estudios psico-sociales, por lo que era necesario lograr la entrada correcta de datos para su posterior uso en la obtención de reportes y de la información en general.

Para lograr lo antes mencionado se hizo uso de las diferentes técnicas de validación, imponiéndose el no dejar campos vacíos o sin seleccionar, introducir datos correctos donde se requería y que cada entrada de valor estuviera en el rango establecido. (Ver anexo 8)

3.6 Conclusiones

Se concluye este capítulo mostrando los diagramas de componentes entre otros elementos que conforman una idea menos abstracta de la arquitectura la cual fue optimizada en la organización de dichos componentes, se implementaron las clases definidas y se validó el sistema a nivel de desarrollador

Conclusiones

- Los artefactos de diseño obtenidos facilitaron la implementación del módulo Registro Cubano de Discapacitados versión 2.0 del Sistema Informático para la Red Nacional de Genética Médica.
- Con la implementación del módulo Registro Cubano de Discapacitados versión 2.0 del Sistema Informático para la Red Nacional de Genética Médica, se obtuvo un producto funcional que cumple con los requerimientos establecidos
- El producto obtenido integrado al SIGM contribuirá al avance de los estudios genéticos.

Recomendaciones

- Realizar el despliegue del Registro Cubano de Discapacitados versión 2.0, como parte fundamental del Sistema Informático para la Red Nacional de Genética Médica.
- Implementar nuevas funcionalidades para obtener gráficas a partir de los reportes estadísticos generados y una interfaz para la administración de las tablas nomencladoras.

Referencias bibliográficas

- [1]. **Sourd, Frank, Pompa.** *Arquitectura, normas y tecnologías para el desarrollo de aplicaciones Informáticas para la salud.* 2007.
- [2] **Jacobson Ivar, Booch Grady, Rumbaugh James.** *El Proceso Unificado de Desarrollo de Software.* España. 2000.
- [3] **Informatizate.** [En línea] 27 de Noviembre de 2002. [Citado el: 30 de Octubre de 2007.]
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- [4] **Molpeceres, Alberto.** *Procesos de desarrollo: RUP, XP y FDD.* 2002.
- [5] **Valverde, Pedro Hernández.** *RUP y su relación con las técnicas y métodos de la ingeniería de usabilidad del software.* 2005.
- [6] **Alarcón, Raúl.** *Diseño Orientado a Objetos con UML.* 2000.
- [7] **Rumbaugh J., Jacobson I., Booch G.** "El Lenguaje Unificado de Modelado. Manual de Referencia"
- Editorial Addison-Wesley - 2000.
- [8] **Gutiérrez, Juan Diego Gallardo.** *Desarrollo web con PHP 5 y MySQL.* 2004
- [9] **Orós, JC.** *Diseño de páginas Web interactivas con JavaScript.* 2005.
- [10] **Linux-OS.** [En línea] 12 de Enero de 2003. [Citado el: 2 de Diciembre de 2007.] <http://www.linux-os.com.ar/linuxos/category/apache/>
- [11] **Sitio de descargas de software** [Citado 8 de Febrero de 2008]
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(LE\)_14725_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(LE)_14725_p/)
- [12] **Zaninotto François, Potencier Fabien.** *Symfony la guía definitiva* <http://www.librosweb.es> .2007.
- [13] **Eclipse.** [En línea] 2008. [Citado 8 de Febrero de 2008]
<http://www.eclipse.org/>
- [14] **Softonic** [Citado 8 de Febrero de 2008]
<http://subversion.softonic.com/linux>
- [15] Rational Unified Process.. Ayuda del RUP. Suite del Rational 2003.
- [16] **Díaz, Moisés Daniel.** *Diseño de aplicaciones Internet usando los Patrones de diseño.* 2003
- [17] **Larman C.** *UML y Patrones.* 2000.
- [18] **El Mundo Informático** [Citado 8 de Febrero de 2008]
<http://jorgesaavedra.wordpress.com/category/patrones-grasp/>
- [19] **Gamma Eric, Hel Richard, Johnson Ralph, Vlissides John.** *Design Patterns: Elements of Reusable Object-Oriented Software. Cuarta Edición (Enero, 2004)*

Bibliografía.

- 1- **Zaninotto François, Potencier Fabien.** *Symfony la guía definitiva* <http://www.librosweb.es> (8/12/2007).
- 2- **Jacobson Ivar, Booch Grady, Rumbaugh James.** *El Proceso Unificado de Desarrollo de Software.* España. 2000.
- 3- **Rumbaugh J., Jacobson I., Booch G.,** "El Lenguaje Unificado de Modelado. Manual de Referencia" - Editorial Addison-Wesley - 2000.
- 4- **Schach Stephe .** *Ingeniería del Software Orientada a Objetos (6ªED.)*México. 2006.
- 5- **MedLine Literatura Internacional en Ciencias de la Salud** [En línea] 1997-2008. [Citado el 8 de Febrero de 2008]
<http://bases.bireme.br/cgi-bin/wxislind.exe/iah/online/>
- 6- **ArchiMED, Gestion integral de centros medicos** [Citado el 8 de Marzo de 2008]
<http://www.grupdes.com/prarchim.htm>
- 7- **A-medic, Gestion médica,** [Citado el 8 de Marzo de 2008]
<http://www.a-medic.com/prampc.htm>
- 8- **OpenHis, Sistema Informático Hospitalario oficial del Grupo BioLinux,** [En línea] 1996-2008 [Citado el 8 de Marzo de 2008] <http://www.openhis.sourceforge.net>
- 9- Rational Unified Process.. Ayuda del RUP. Suite del Rational 2003.
- 10- **Ciberaula** [Citado 8 de Febrero de 2008].
http://java.ciberaula.com/articulo/disenio_patrones_j2ee/
- 11- **Larman C.** "UML y Patrones" - Segunda Edición - Editorial Prentice-Hall .2003.
- 12- **Gutiérrez Gallardo Juan Diego.** *Desarrollo web con PHP 5 y MySQL.* 2004
- 13- **Puerto, Roberto.** *Avanzando hacia la sociedad de la información.* GIGA no.1/2002.
14. **Úbeda Luís.** *Vía a la sociedad del conocimiento.* (GIGA no. 4/2005.)
15. Softel. Documento sobre la arquitectura de software para los componentes a emplear por el sistema de información para la salud. Softel, 2006.
17. **Pompa Frank.** *Guía para el desarrollo de módulos para el Registro Informatizado de Salud (RIS).*2005
18. **Delgado Ariel, Vidal María.** *Informática en la Salud Pública Cubana.* 2005.
19. **Rodríguez, José A.** *Tutorial de PHP y MySQL.*

http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP (20/01/2007)

20. **Larman C.** *UML y Patrones*. 2000.

21. **Pressman, Roger.** *Ingeniería de software. Un enfoque práctico*. McGraw.Hill/Interamericana de España. 2002

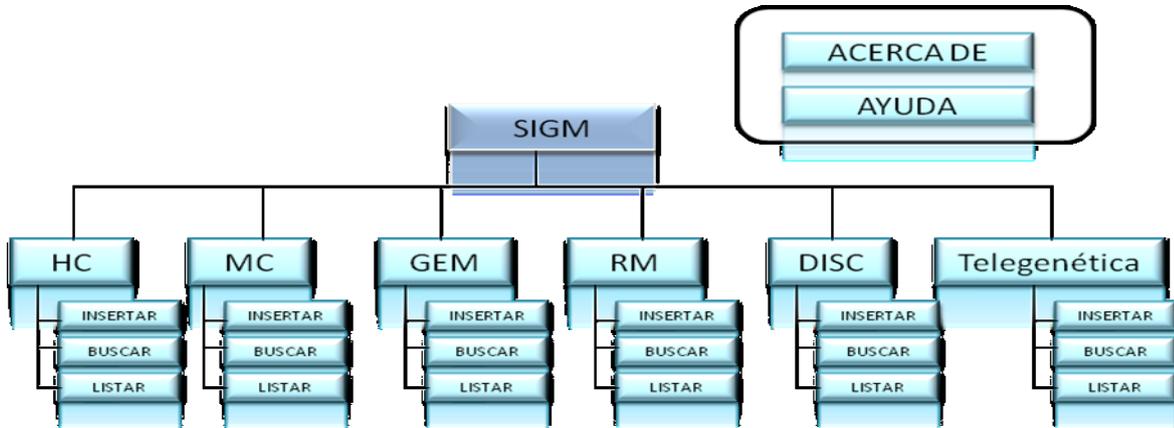
22. Collazo Avila Irina, Sistema Informático para la Gestión de Información del Registro Cubano de Discapacitados, 2007

23. Sánchez Perodín Yusdenis, Sistema de Información Genética Línea base de la arquitectura, 2007

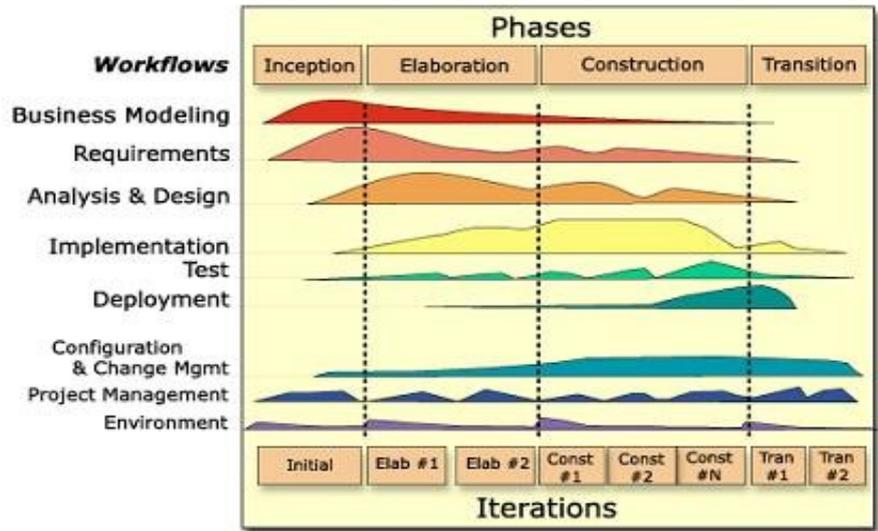
Índice de anexos

Anexo 1. Mapa del SIGM.....	68
Anexo 2. Fases y Flujo de Trabajo de RUP.	79
Anexo 3. Artefactos generados.	70
Anexo 4. Descripción de Casos de Uso	71
Anexo 5. Actores del Sistema	75
Anexo 6. Plantilla decorada por un layout	76
Anexo 7. El flujo de trabajo de Symfony	77
Anexo 8. Interfaces de error.....	78
Anexo 9. Instrumentos.	82

ANEXO 1. MAPA DEL SIGM.



ANEXO 2. FASES Y FLUJO DE TRABAJO DE RUP.



ANEXO 3. ARTEFACTOS GENERADOS.



ANEXO 4. DESCRIPCIÓN DE CASOS DE USO.

Caso de Uso:	Insertar Datos de Discapacitado	
Actores:	Defectólogo (Inicia el caso de uso)	
Resumen:	El caso de uso se inicia cuando el defectólogo desea insertar los datos del paciente en el sistema, el discapacitado ofrece los datos que se van a incluir en la aplicación y el defectólogo procede a la inserción de los mismos.	
Precondiciones:	El defectólogo tiene que estar registrado para poder acceder a esta parte del sistema. Debe estar disponible la información a la que tiene acceso el usuario (los consejos populares, la provincia, etc.).	
Referencias	RF1	
Prioridad	Crítica	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del sistema	
El Defectólogo selecciona la opción de insertar los datos complementarios de un Discapacitado.	Se llama al Caso de uso incluido “Gestionar Datos Primarios” para buscar a la persona a la cual se van a insertar los datos complementarios.	
El Defectólogo selecciona a la persona con discapacidad al cual le va a llenar los campos complementarios.	El sistema le brinda una interfaz que muestra los criterios para la inserción de los datos.	
El defectólogo registra los datos complementarios del discapacitado:	El sistema verifica que no se halla quedado ningún campo obligatorio vacío.	
	El sistema emite un mensaje diciendo que se ha insertado los datos correctamente.	
Prototipo de Interfaz		
Flujos Alternos		
Acción del Actor	Respuesta del sistema	

	6.1 Se emite un mensaje para que se llene los campos obligatorios.
Prototipo de Interfaz	
Poscondiciones	Quedará registrada toda la información de los discapacitados en la base de datos.

Modificar datos complementarios de un discapacitado.

Descripción de Casos de uso

Caso de Uso:	Modificar Datos de Discapacitado	
Actores:	Defectólogo (Inicia el caso de uso)	
Resumen:	El caso de uso inicia cuando los usuarios que pueden ser genetistas (provinciales o nacionales) o el defectólogo necesitan modificar los datos de un discapacitado. En el menú aparecerá la opción de modificar el discapacitado.	
Precondiciones:	El defectólogo debe estar registrado con los permisos pertinentes para poder acceder a esta parte del sistema. La información debe estar disponible.	
Referencias	RF2	
Prioridad	secundario	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Sistema	
El Defectólogo selecciona la opción de modificar los datos específicos de un Discapacitado.	2. Se llama al Caso de uso incluido “Gestionar Datos Primarios” para buscar a la persona a la cual se van a modificar los datos complementarios.	
El Defectólogo selecciona al discapacitado al cual le va a modificar los campos específicos.	El sistema le brinda una interfaz que muestra los datos específicos del discapacitado que pueden ser modificados	
4. El Defectólogo registra los nuevos datos	5. El sistema verifica que no se halla quedado	

específicos del criterio seleccionado y presiona el botón modificar.	ningún campo obligatorio vacío.
	6. El sistema modifica los datos del discapacitado que fueron cambiados.
	7. El sistema emite un mensaje diciendo que se han modificado los datos correctamente.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1 Se emite un mensaje para que se llene los campos obligatorios.
Prototipo de Interfaz	
Poscondiciones	Deben guardarse las modificaciones en los datos del discapacitado.

Mostrar Reportes Estadísticos

Descripción de Casos de Uso

Caso de Uso:	Obtener Reportes Estadísticos
Actores:	Usuario(Inicia el caso de uso)
Resumen:	El caso de uso inicia cuando el Usuario necesita conocer la cantidad de discapacitados que existen en dependencia de un criterio determinado, puede ser por sexo, por ocupación laboral, según la clasificación de su enfermedad, por la causa referida de su enfermedad, según el amparo filial, según el vinculo laboral, según la capacidad laboral, el control de esfínter, la evaluación funcional, el consumo de alcohol o la causa en el período post-natal, para ello selecciona en el menú el reporte y el sistema se encarga de mostrarle las estadísticas y los cálculos correspondientes en dependencia del nivel que posea el usuario, si es municipal solo su consejo popular, si es provincial toda su provincia con sus consejos populares y si es nacional todo el país con las provincias y los consejos populares.
Precondiciones:	Los usuarios deben estar registrados con los permisos pertinentes para poder acceder a esta parte del sistema. La información debe estar disponible.

Referencias	RF4, RF4.1, RF4.2, RF4.3, RF4.4, RF4.5.	
Prioridad	Secundaria	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Negocio	
El Genetista selecciona la opción Obtener Reportes Estadísticos.	El sistema verifica el nivel del Genetista. El sistema muestra la interfaz de obtener reporte estadístico.	
El Genetista selecciona el criterio de búsqueda que puede ser: sexo, ocupación laboral, clasificación de su enfermedad, causa referida de su enfermedad, amparo filial, vinculo laboral, según la capacidad laboral, control de esfínter, la evaluación funcional, consumo de alcohol o la causa en el período post-natal	El sistema busca y muestra las estadísticas y los cálculos correspondientes en dependencia del nivel que posea el Genetista	
Prototipo de Interfaz		
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
Prototipo de Interfaz		
Poscondiciones	Se muestra la información referente al reporte en dependencia del nivel del usuario que la solicita.	

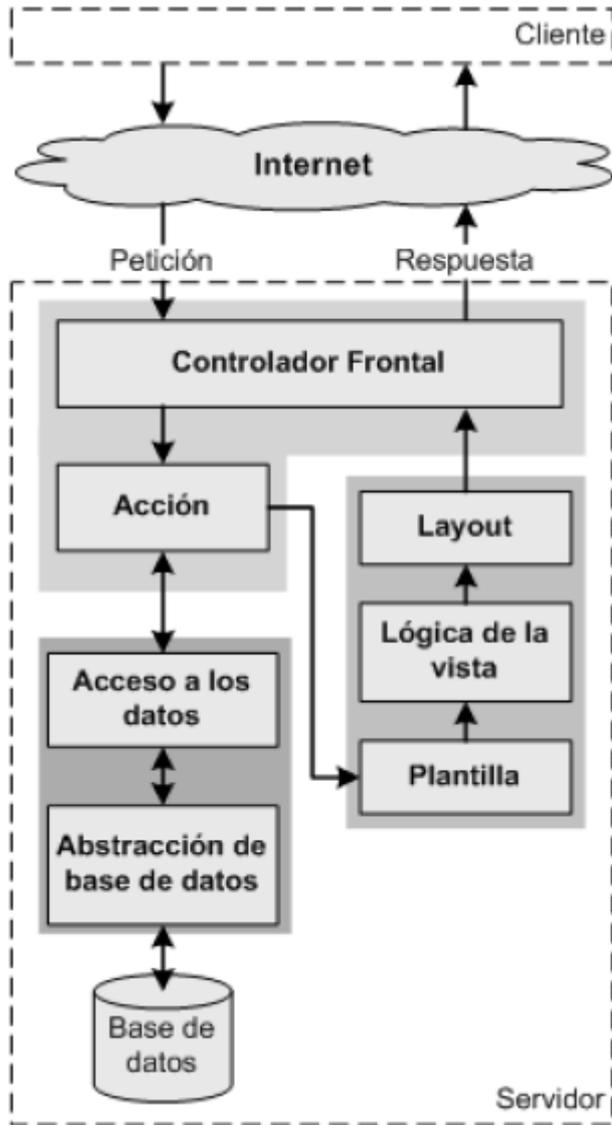
ANEXO 5. ACTORES DEL SISTEMA.

Actor	Descripción
Usuarios	Máster en Asesoramiento Genético que trabaja con el sistema y obtiene los reportes estadísticos del mismo, puede ser un genetista municipal, provincial o nacional.
Defectólogo	Es el especialista encargado de introducir los datos, modificarlos del discapacitado.

ANEXO 6. PLANTILLA DECORADA POR UN LAYOUT.



ANEXO 7. EL FLUJO DE TRABAJO DE SYMFONY



ANEXO 8. INTERFACES DE ERROR.

▼ Datos Primarios

<p>Nombre: <u>Javier</u></p> <p>Primer Apellido: <u>Mirabal</u></p> <p>Segundo Apellido: <u>Mirabal</u></p> <p>Estado Civil: <u>Casado</u></p> <p>Consejo Popular: <u>Indeterminado</u></p> <p>Area de salud: <u>jhdb</u></p> <p>Dirección: <u>Indeterminada</u></p>	<p>Fecha de nacimiento:</p> <p>Edad: <u>50</u></p> <p>Sexo: <u>Femenino</u></p> <p>No. C.I.: <u>62021514769</u></p> <p>Piel: <u>Amarilla</u></p> <p>Escolaridad: <<Seleccione>> ↓ Campo Obligatorio ↓ *</p>
---	--

▼ Tipo de discapacidad y Evaluación funcional

<p>Tipo de Discapacidad *</p> <p style="text-align: center; color: red;">↓ Campo Obligatorio ↓</p> <p>Físico motora <input type="checkbox"/></p> <p>Visual <input type="checkbox"/></p> <p>Auditiva <input type="checkbox"/></p> <p>Mental <input type="checkbox"/></p> <p>Visceral <input type="checkbox"/></p> <p>Mixta <input type="checkbox"/></p> <p>Otra <input type="checkbox"/></p>	<p>Evaluación Funcional *</p> <p style="text-align: center; color: red;">↓ Campo Obligatorio ↓</p> <p>Válido <input type="radio"/></p> <p>Semiválido <input type="radio"/></p> <p>No válido <input type="radio"/></p>
--	--

▼ Datos Particulares del Discapacitado

¿Está incorporado a alguna asociación de discapacitado? * Si No ↓ Campo Obligatorio ↓

▼ Condiciones de Vida

<p style="color: red;">↓ Campo Obligatorio ↓</p> <p>Condiciones de la Vivienda <<Seleccione>> * ↓ Campo Obligatorio ↓</p>	<p style="color: red;">↓ Campo Obligatorio ↓</p> <p>Higiene de la Vivienda <<Seleccione>> * ↓ Campo Obligatorio ↓</p>
---	---

Apoyo Familiar

↓ Campo Obligatorio ↓

Composición del núcleo familiar *

Padre

Madre

Hermanos

Hijos

Esposo(a)

Otros

↓ Campo Obligatorio ↓

Amparo Filial * Con Amparo Sin Amparo

↓ Campo Obligatorio ↓

Ingreso total del núcleo *

↓ Campo Obligatorio ↓

Ingreso Percápita *

↓ Campo Obligatorio ↓

No. de convivientes en el núcleo *

↓ Campo Obligatorio ↓

Condiciones de vida del discapacitado con respecto al resto de la familia *

Servicios de Atención Domiciliaria

	Si	No	
Necesita alimentación *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe alimentación *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita servicios de tintorería *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe el servicio de tintorería *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita el servicio peluquería y barbería *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe el servicio peluquería y barbería *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita el servicio de limpieza del hogar *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe el servicio de limpieza del hogar *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita la prioridad para la reparación de equipos *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe la prioridad para la reparación de equipos *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓

Situación Laboral

Estado de salud y necesidad de implementos, medicamentos y ayudas técnicas			
	Si	No	
Valoración del no válido(¿Se encuentra postrado?) *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Tiene relajación de esfínteres el discapacitado *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita cuidador permanente *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Tiene cuidador permanente *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesidad de material para el aseo *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe material para el aseo *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita Pato o Cuña *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe Pato o Cuña *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita tela antiséptica *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe tela antiséptica *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita Hule *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe Hule *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita medicamentos para su discapacidad *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe medicamentos para su discapacidad *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita silla de ruedas *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Tiene silla de ruedas *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita calzado ortopédico *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Tiene calzado ortopédico *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita Ortesis y bandajes *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe Ortesis y bandajes *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita Prótesis de miembros *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Tiene Prótesis de miembros *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita bastón o muletas *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Tiene bastón o muletas *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita Cama Especial *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓

<input checked="" type="checkbox"/> Necesidad de servicios de salud			
	Si	No	
Necesita ingreso en Hospital Psiquiátrico de Larga Estadía *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Se encuentra ingresado en Hospital Psiquiátrico de Larga Estadía *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita ingreso en Hogar de Ancianos *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Se encuentra ingresado en Hogar de Anciano *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita ingreso en Centro Diurno *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Se encuentra Seminternado en el Centro Diurno *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita atención especializada para su discapacidad *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Esta recibiendo atención especializada para su discapacidad *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita servicio de rehabilitación física *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Esta recibiendo el servicio de reavitiación física *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita servicio Estomatológicos *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Esta recibiendo los servicios Estomatológicos *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Ha necesitado el Servicio de atención de Urgencias *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Ha recibido el Servicio de Urgencia sin dificultad *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
<input checked="" type="checkbox"/> Necesidad Educativa			
	Si	No	
Necesita educación especial *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe la educación especial *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Necesita educación general *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
Recibe la educación general *	<input type="radio"/>	<input type="radio"/>	↓ Campo Obligatorio ↓
<input checked="" type="checkbox"/> Causa referida de la discapacidad			
↓ Campo Obligatorio ↓			
Seleccione la causa de la discapacidad * Prenatal <input type="radio"/> Perinatal <input type="radio"/> Post-Natal <input type="radio"/>			

ANEXO 9. INSTRUMENTOS.

Instrumento Psicosocial utilizado en el estudio para tomar los datos de las personas discapacitadas del país.

I.- DATOS GENERALES

1. Folio _____
2. Nombre(s) _____
3. 1er Apellido _____
4. 2do Apellido _____
5. Dirección _____
6. Consejo Popular _____
7. Área de Salud _____
8. CMF _____
9. Fecha de Nacimiento _____
10. Edad _____
11. Sexo
 1. Femenino _____
 2. Masculino _____
12. Documento de identificación
 1. No. de Carné de Identidad _____
 2. No. de Tarjeta de Menor _____
 3. No tiene _____
13. ¿Cuál es el color de piel?
 1. Blanco _____ (Sólo una marca)
 2. Negro _____
 3. Mestizo o mulato _____
 4. Amarillo _____
14. Escolaridad:
 1. ___ Sin escolarizar
 2. ___ Preescolar
 3. ___ Primaria sin terminar
 4. ___ Primaria
 5. ___ Secundaria
 6. ___ Pre Universitaria

7. ___ Universitaria

15. Estado Civil:

1. ___ Soltero
2. ___ Casado
3. ___ Viudo
4. ___ Concubinato
5. ___ Divorciado

16. ¿Se encuentra incorporado a alguna asociación de Discapacitados? :

1. ___ Sí
2. ___ No

17. Asociación a que pertenece:

1. ___ ANCI
2. ___ ANSOC
3. ___ ACLIFIM

18. Si no esta asociado:

1. ___ Desea
2. ___ No desea
3. ___ No procede

II.- CONDICIONES DE VIDA.

19. Condiciones de la vivienda:

1. ___ Buenas
2. ___ Regulares
3. ___ Malas
4. ___ Críticas

20. Higiene del hogar:

1. ___ Buena
2. ___ Regular
3. ___ Mala

21. Barreras arquitectónicas en la vivienda. (Solo para el físico-motor y ciegos)

1. ___ Sí
2. ___ No

22. Número de personas en el dormitorio del discapacitado.

- 1. ____ Una
- 2. ____ Dos
- 3. ____ Tres o cuatro
- 4. ____ Cinco o más

23. Accesibilidad a la transportación.

- 1. ____ Buena
- 2. ____ Regular
- 3. ____ Mala

III.- APOYO FAMILIAR.

Núcleo familiar	Composición del Núcleo Familiar	
	1. Si	2. No
24.0 Padre		
24.1 Madre		
24.2 Hermanos		
24.3 Hijos		
24.4 Esposo(a)		
24.5 Otro		

24.4 No. de convivientes en el núcleo _____

25 Amparo filial del discapacitado:

- 1. ____ Con amparo
- 2. ____ Sin amparo

26. Condiciones de vida con respecto al resto de la familia.

- 1. ____ Mejores
- 2. ____ Iguales
- 3. ____ Peores

27. Ingreso total del núcleo _____

28. Percápita _____

IV. SERVICIOS DE ATENCIÓN DOMICILIARIA.

	Si	No
29. Necesita alimentación		
30. Recibe alimentación		
31. Necesita el servicio de tintorería		
32. Recibe el servicio de tintorería		
33. Necesita el servicio peluquería y barbería		
34. Recibe el servicio peluquería y barbería		
35. Necesita el servicio de limpieza del hogar		
36. Recibe el servicio de limpieza del hogar		
37. Necesita la prioridad para la reparación de equipos		
38. Se le prioriza para la reparación de equipos electrodomésticos		

V.- SITUACIÓN LABORAL.

39. Ocupación:

1. _____ Ama de casa
2. _____ Trabajador
3. _____ Estudiante
4. _____ Jubilado
5. _____ Pensionado
6. _____ Desempleado
7. _____ Asistenciado

40. Capacidad Laboral:

1. _____ Apto
2. _____ No apto

41. Vínculo Laboral actual:

1. _____ Centro ordinario
2. _____ Taller protegido
3. _____ Por cuenta propia
4. _____ No tiene

42. Salario:

1. _____ No
2. _____ Sí

3. _____ ¿Cuánto?

42.1 Recibe ayuda económica 1. Sí _____ 2. No _____ 3. ¿Cuánto? _____

VI.- ESTADO DE SALUD Y NECESIDAD DE IMPLEMENTOS, MEDICAMENTOS Y AYUDAS TÉCNICAS.

43. Tipo de discapacidad:

- 1. _____ Físico motora
- 2. _____ Visual
- 3. _____ Auditiva
- 4. _____ Mental
- 5. _____ Visceral
- 6. _____ Mixta
- 6.1 _____ Otra
- 6.1.1 Especifique _____

44. Evaluación Funcional:

- 1. _____ Válido
- 2. _____ Semiválido
- 3. _____ No válido

	Si	No
45 Valoración del No Válido (¿Se encuentra postrado?)		
46 Tiene relajación de esfínteres el discapacitado		
47. Necesita cuidador permanente		
47.1 Tiene cuidador permanente		
48. Necesidad de material para el aseo		
48.1 Recibe material para el aseo		
49 Necesita Pato o Cuña		
49.1 Recibe Pato o Cuña		
50 Necesita tela antiséptica		
50.1 Recibe tela antiséptica		
51 Necesita Hule		
51.1 Recibe Hule		
52 Necesita medicamentos para su discapacidad		
52.1 Recibe medicamentos para su discapacidad		

53 Necesita silla de ruedas		
53.1 Tiene silla de ruedas		
54 Necesita calzado ortopédico		
54.1 Tiene calzado ortopédico		
55 Necesita Ortesis y bandajes		
55.1 Recibe Ortesis y bandajes		
56. Necesita Prótesis de miembros		
56.1 Tiene prótesis de miembros		
57 Necesita Bastones o muletas		
57.1 Tiene bastones o muletas		
58 Necesita Cama Fowler		
58.1 Tiene Cama Fowler		
59 Necesita Colchón		
59.1 Tiene Colchón		
60 Necesita Prótesis auditivas		
60.1 Tiene Prótesis auditivas		
61 Necesita Espejuelos		
61.1 Tiene Espejuelos		

VII.- NECESIDADES DE SERVICIOS DE SALUD.

	Si	No
62 Necesita ingreso en Hospital Psiquiátrico de Larga Estadía		
62.1 Se encuentra ingresado		
63 Necesita ingreso en Hogar de Ancianos		
63.1 Se encuentra ingresado		
64 Necesita ingreso en Centro Diurno		
64.1 Se encuentra seminternado		
65 Necesita atención especializada para su discapacidad		
65.1 Está recibiendo atención especializada para su discapacidad		
66 Necesita servicio de rehabilitación física		
66.1 Está recibiendo el servicio de rehabilitación física		
67 Necesita los servicios estomatológicos		

67.1 Está recibiendo el servicio estomatológico		
68 Ha necesitado el Servicio de atención de Urgencias		
68.1 Ha recibido el Servicio de Urgencia sin dificultad		
62 Necesita ingreso en Hospital Psiquiátrico de Larga Estadía		

VIII.- NECESIDAD EDUCACIONAL (Menores de 18 años)

	Si	No
69 Necesita educación especial		
69.1 Recibe la educación especial		
70 Necesita educación general		
70.1 Recibe la educación general		

IX. – CAUSA REFERIDA DE LA DISCAPACIDAD.

71. Prenatal:

1. ___ Enfermedad heredo familiar.
2. ___ Aberración cromosómica.
3. ___ Enfermedad por agente biológico, físico o químico.
4. ___ Factor mecánico
5. ___ Trastorno genético multifactorial
6. ___ Otra

71.1 Especifique _____

72. Perinatal:

1. ___ Trauma del parto
2. ___ Encefalopatía hipóxico-isquémica
3. ___ Conflicto Rh
4. ___ Infecciones congénitas
5. ___ Otras

72.1 Especifique _____

73 Post-Natal:

1. ___ Infecciones
2. ___ Accidentes del tránsito
3. ___ Accidentes en la escuela
4. ___ Accidentes en el trabajo
5. ___ Accidentes en el hogar

- 6. ____ Enfermedad sistemática
- 7. ____ Intoxicaciones
- 8. ____ Psicotraumas
- 9. ____ Desnutrición
- 10. ____ Sobredosis de medicamentos
- 11. ____ Otras

73.1 Especifique _____

74. ____ De causas no precisadas.

X.- CLASIFICACIÓN EN RELACIÓN CON EL CONSUMO DE BEBIDAS ALCOHÓLICAS DEL PACIENTE Y FAMILIARES.

Miembros del núcleo que consumen alcohol (Sólo una marca)

Miembros del Núcleo:	Clasificación			
	1. Social	2. De Riesgo	3. Abuso	4. Dependencia
75. Madre				
76. Padre				
77. Hermano				
78. Esposo(a)				
79. Hijo(a)				
80. Otros				
81. Discapacitado				

XI.- OBSERVACIONES:

(Datos significativos brindados por el individuo que respondió la encuesta, no incluidos en la misma, así como alguna observación relevante que el encuestador considere oportuno comunicar):

XII.- VALORACIÓN SOLO PARA USO DE LA COMISIÓN MUNICIPAL.

82. Incidencia: 1. ____ Municipal 2. ____ Relevante 3. ____ No procede.

Glosario de Términos.

En el glosario aparecen un grupo de conceptos básicos relacionados con la elaboración del sistema informático.

Técnicos

Expediente del proyecto: Conjunto de documentos y plantillas estructuradas siguiendo una jerarquía que constituyen referencia para la documentación de los proyectos de la Universidad de las Ciencias Informáticas.

Siglas

CNGM: Centro Nacional de Genética Médica

GM: Genética Médica. Nombre del proyecto.

INFOMED: Red de Salud de Cuba.

LMC: Lineamientos Mínimos de Calidad que deben cumplir los proyectos de la Universidad de las Ciencias Informáticas con el fin de obtener un producto final con calidad.

MINSAP: Ministerio de Salud Pública.

SIGM: Sistema Informático de Genética Médica. Producto del proyecto

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

Específicos

Con amparo filial: Tiene familia consanguínea o sustituta y se ocupa del discapacitado.

Discapacitado: Limitado para algunas actividades a causa de una deficiencia física o psíquica.

Discapacidad: Es toda restricción o ausencia, debida a una deficiencia, de la capacidad de realizar una actividad en la forma y dentro del margen que se considera normal para un ser humano en su contexto social. Puede ser transitoria o definitiva, reversible o irreversible, progresiva o regresiva.

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.

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

Sin amparo filial: No tiene familia o la tiene y no se ocupa del discapacitado y/o está sometido a maltrato.