

Universidad de las Ciencias Informáticas
Facultad # 6



Sistema Informático de Genética Médica alasMEDIGEN.
Reportes de discapacidad.

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

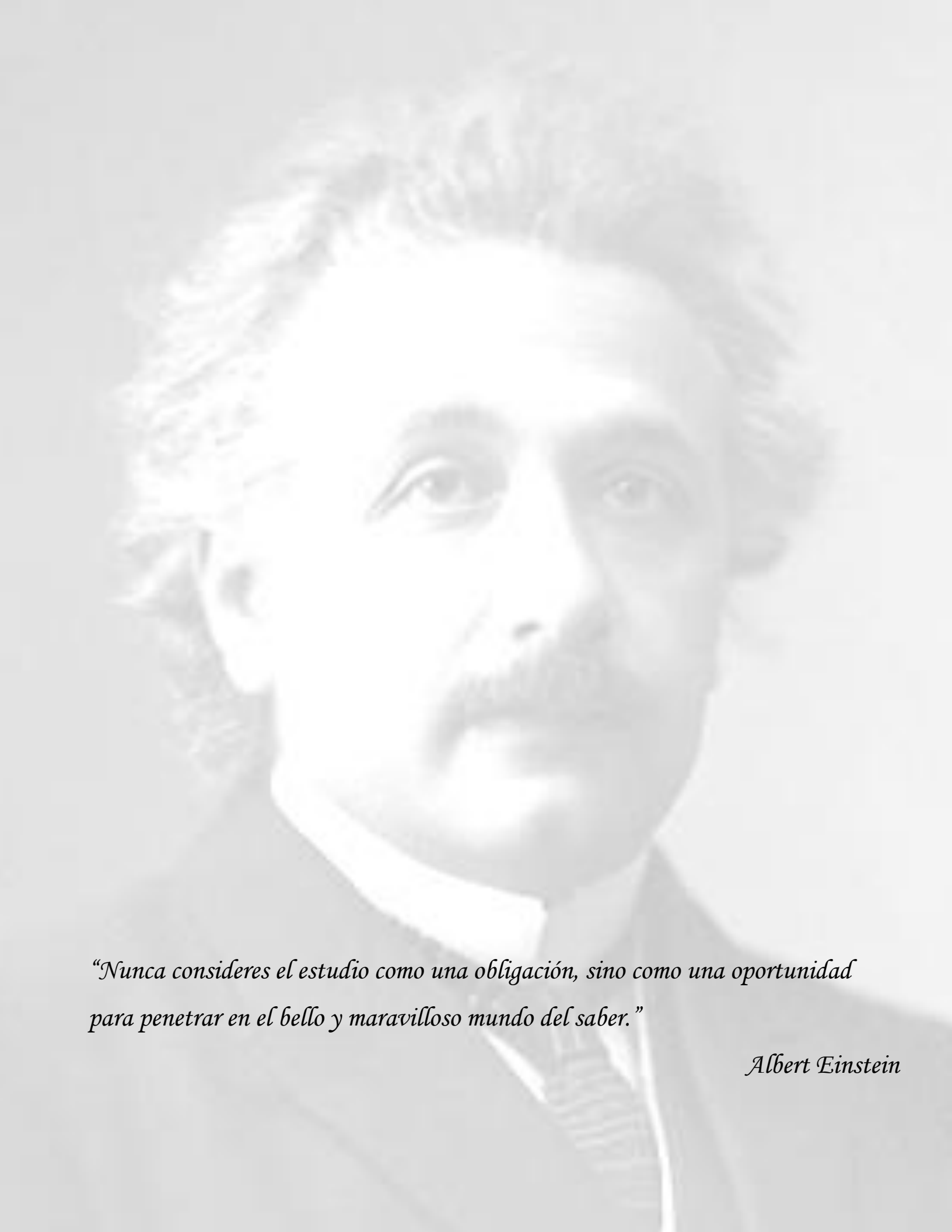
Autores:

Yissel de La Caridad Falcón Sánchez
Erick Martín Guerra Rodríguez

Tutores:

Ing. Yudiel La Rosa González
Ing. Dayana Joseph Smarth
Msc. Elvismary Molina de Armas

La Habana, junio de 2013
“Año 55 de la Revolución”



“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein

Declaración de Autoría

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

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

Yissel de la Caridad Falcón Sánchez

Firma del Autor

Erick Martín Guerra Rodríguez

Firma del Tutor

Ing. Dayana Joseph Smarth

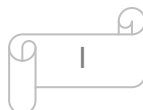
Firma del Autor

Ing. Yudiel La Rosa González

Firma del Tutor

Msc. Elvismary Molina de Armas

Firma del Co-Tutor



Datos de Contacto

Autora: Yissel de La Caridad Falcón Sánchez

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: yfalcon@estudiantes.uci.cu

Autor: Erick Martín Guerra Rodríguez

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: emguerra@estudiantes.uci.cu

Tutores:

Ing. Dayana Joseph Smarth.

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: djoseph@uci.cu

Ing. Yudiel La Rosa González

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: yglarosag@uci.cu

Co-Tutor:

MsC. Elvismary Molina De Armas

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: emolina@uci.cu



Agradecimientos

De Yissel:

Un sueño se hace realidad cuando se lucha día a día por él, cuando se pone todo el empeño y dedicación para alcanzarlo. Hoy se hizo realidad mi sueño, hoy después de muchos años de sacrificio he alcanzado la meta que una vez me propuse y por eso quiero agradecer a:

Dios que le debo todo lo soy, quien ha escuchado todas mis oraciones y peticiones, quien ha estado espiritualmente en todos los momentos más difíciles de mi vida y me ha demostrado que: “Todo lo puedo en Cristo que me fortalece”. Filipenses 4-13.

Mi mamá Nilda que cariñosamente todos la conocemos y le decimos “**Mimi**”. Por darme el tesoro más valioso que puede dársele a una hija: amor, por el sacrificio realizado para ayudarme a alcanzar mi sueño con la confianza plena y apoyo que me ha brindado en todo momento. A quien sin escatimar esfuerzo alguno, han sacrificado la mayor parte de su vida para formarme y educarme.

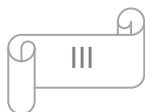
Mi padre Joaquin por sus cuidados y atenciones en el transcurso de mi vida, por siempre confiar en mi capacidad de poder llegar hasta el lugar en el que hoy me encuentro, por amarme y hacerme una persona de bien para con la sociedad.

Mi hermanito Yunio, por darme todo el amor y cariño que se le pueda dar a una hermana, por brindarme su apoyo, por contribuir para que cada día fuera una mejor persona, por darme la fuerza y el valor para seguir adelante. Yunito siempre vas a tener un lugar guardado en mi corazón.

Fr Jose Mari: Por ser el abuelito que nunca tuve, por estar siempre pendiente a mí, por su amor, sus oraciones. Padrecito quiero que sepas que te quiero mucho y que aunque pasen muchos años siempre serás parte de mí y que nunca te olvidaré.

Mi novio Adrian por apoyarme en todas las decisiones que he tomado, por todos los momentos lindos que hemos vivido juntos y por estar siempre pendiente a mi trabajo.

Landy: Por darme su mano amiga y desinteresada en uno de los momentos más difíciles en mi vida y por demostrarme que el valor de las personas nace con ellas.



Dayana, Elvismary y a Yudiel por ser mis tutores, por su ayuda y dedicación para cumplir mis sueños, por ayudarme con sus críticas y consejos.

A todos los compañeros de estudio que he conocido desde mi primer año en esta universidad, de manera muy particular a mi dúo de Tesis Erick y todos los profesores que han contribuido de alguna manera a mi formación profesional a través de sus enseñanzas.

De Erick:

A mis padres Martín y Lupe por darme la dicha de pertenecer a este mundo y ser mis guías siempre, a mi padrastro Beimis que se ha convertido en un papá más. A mi tata Glenda que la quiero con la vida, por depositar su confianza en mí. A mi abuelito Rigo que aunque no está físicamente no hay un minuto de mi existencia en el que no esté conmigo en mi mente y en mi corazón. A mi abuelita Mina por ser tan dulce y especial conmigo. A mi tía Ibis, mi segunda mamá y su esposo Félix, mi primita que la adoro Jessy y su novio Alejandro que son los culpables de tantos momentos de felicidad. A mi amigo Ariel que aunque estamos en facultades diferentes siempre estuvo ahí en todo momento brindándome su mano cuando daba algún tropezón, mi amigo Pedro Pablo que a pesar de no estar en la universidad siempre estuvo ahí con sus consejos y su apoyo en los momentos malos y buenos, a mi cuñado Maikel que más que mi cuñado se ha convertido en un gran amigo con el cual he podido contar siempre, a mi profe Niurka que más que mi profesora se convirtió en una gran amiga. A todos mis compañeros de aula que son muy especiales, y demás conocidos de la universidad. A mi compañera de tesis Yissel que sin su apoyo no hubiese sido posible salir adelante. A mis tutoras Dayana y Elvismary por su colaboración. A Patricia, mi churry, mi gran amor, por brindarme apoyo, por llegar a mi vida en el momento que más lo necesitaba, por estar siempre a mi lado en estos meses de continua desesperación. Por todas estas cosas, le agradezco con todo el amor del mundo.

Dedicatoria

De Yissel:

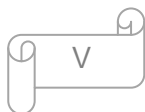
Quiero dedicar este Trabajo de Diploma a la persona más especial que tengo en la vida, ella es:

La persona que más quiero en este mundo, "Mimi", la principal causante que este momento se haya hecho realidad, por lograr este sueño que también es de ella, por ser la luz que siempre guió cada uno de mis pasos, por brindarme siempre su amor incondicional, por sus enseñanzas, sus años de sacrificio, su confianza, sus consejos y apoyo constante, por ser la mejor mamá de este mundo, por confiar tanto en mí y estar seguro que no lo defraudaría. Del 1 al 100 Te quiero 101.

Mimita he llegado al final de este camino y en mi han quedado marcadas, huellas profundas de éste recorrido, quiero que sientas que el objetivo logrado también es tuyo y que la fuerza que me ayudó a conseguirlo fue tu apoyo.

De Erick:

Dedico este trabajo especialmente a mis padres quienes me infundieron la ética y el rigor que guían mi transitar por la vida, a mi hermana Glenda que la quiero mucho, a mi tía Ibis por su apoyo y por depositar su confianza en mí, a mis abuelos que han sido muy especiales conmigo, a mi prima Jessy y a mi novia que significa mucho en mi vida.



Resumen

En el Centro Nacional de Genética Médica se han desarrollado estudios a pacientes que pudieran presentar determinadas enfermedades o padecimientos genéticos. Estos estudios generan gran volumen de información de las nuevas afecciones que van apareciendo, además de otros datos de interés referentes al paciente que es sujeto a la investigación. Los mismos son almacenados y gestionados desde el Sistema Informático de Genética Médica alasMEDIGEN.

Actualmente los módulos Registro Cubano de Discapacitados Intelectual y Registro Cubano de Discapacidad cuentan con tres y seis reportes respectivamente, los cuales representan un bajo por ciento de la información demandada por los genetistas e impide realizar un estudio más detallado de los datos recogidos. Además estos módulos presentan dificultades para visualizar los resultados de los reportes utilizados por los genetistas debido a que no poseen una representación gráfica de la información que manejan.

Debido a lo anterior, se decide realizar nuevos reportes que se incorporarán a los dos módulos. Estos reportes permitirán una mejor atención a las personas que presentan alguna discapacidad y posibilitará a los genetistas realizar estudios a las personas que presenten alguna enfermedad genética en aras de elevar la calidad de vida de la población cubana.

Palabras claves: Registro Cubano de Discapacitados Intelectual, Registro Cubano de Discapacidad, reporte, alasMEDIGEN; discapacidad, genetista.

Índice

DATOS DE CONTACTO II

AGRADECIMIENTOS..... III

DEDICATORIA V

RESUMEN..... VI

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 5

 1.1 Desarrollo de los estudios genéticos en Cuba..... 5

 1.2 alasMEDIGEN: Sistema Informático de Genética Médica 6

 1.3 Tecnologías y Herramientas..... 10

 1.3.1 Reportes en el ámbito de la informática 10

 1.3.2 Sistemas Generadores de Reportes..... 11

 1.3.3 ORM Propel 13

 1.3.4 Técnica de visualización de datos mediante el gráfico de barras 14

 1.3.5 Metodología de desarrollo 14

 1.3.6 Herramienta de Modelado CASE 15

 1.3.7 Lenguaje de programación..... 16

 1.3.8 Framework de desarrollo..... 17

 1.3.9 Servidor Web 18

 1.3.10 Sistema Gestor de Base de Datos(SGDB) 19

 1.3.11 Entorno de desarrollo 20

 1.3.12 Bibliotecas de Clases 21

 1.3.13 Patrones de arquitectura 22

 Conclusiones..... 23

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA 24

 2.1 Modelo de dominio 24

 2.2 Especificación de los requisitos del sistema 25

 2.3 Definición de los casos de uso del sistema 29

 Conclusiones..... 35

CAPÍTULO 3: DISEÑO DEL SISTEMA	36
3.1 Vista lógica de la arquitectura	36
3.2 Diagrama de clases del diseño.....	38
3.3 Patrones.....	40
3.4 Diagrama de Interacción	45
3.5 Diagrama de despliegue.....	47
Conclusiones.....	48
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	49
4.1 Estructura del sistema.....	49
4.2 Diagrama de componente	50
4.3 Prueba de software	52
4.4 Evidencia de la ejecución de las pruebas.....	59
Conclusiones.....	61
CONCLUSIONES	62
RECOMENDACIONES	63
TRABAJOS CITADOS	64
BIBLIOGRAFÍA	66

Índice de Figuras

Figura # 1: Diagrama de clases del modelo de dominio	25
Figura # 2: Diagrama de caso de uso del sistema.....	30
Figura # 3: Interfaz del reporte de Discapacidad Intelectual a nivel nacional.....	34
Figura # 4: Interfaz del reporte de Discapacidad Intelectual a nivel provincial.....	34
Figura # 5: Vista lógica del sistema.....	36
Figura # 6: Diagrama de clases del diseño. CU Generar reportes de discapacidad por tipo.	38
Figura # 7: Diagrama de clases del diseño. Paquete modelo del CU reporte por tipo de discapacidad.	40
Figura # 9: Diagrama de la vista de despliegue.	48
Figura # 10: Estructura del sistema.....	50
Figura # 11: Diagrama de componentes. Generar reporte de tipo de discapacidad.	51
Figura # 12: Representación del código del método executeReporteConsumo().	54
Figura # 13: Grafo del flujo asociado al método executeReporteConsumo().	55
Figura # 14: Reporte por cantidad de consumo.	60
Figura # 15: Clasificación de consumo de bebidas alcohólicas.	60
Figura # 18: Reporte por cantidad de consumo.	61

Índice de Tablas

Tabla # 1: Descripción de las clases del modelo de dominio.	24
Tabla # 2: Actores del sistema.	30
Tabla # 3: Descripción del caso de uso del sistema. Generar reporte de discapacidad.	32
Tabla # 4: Descripción del caso de uso del sistema. Generar reporte de discapacidad intelectual.	34
Tabla # 5: Descripción del caso de uso de graficar reporte.....	35
Tabla # 6: Caminos básicos del flujo de la prueba de caja blanca.	55
Tabla # 7: Caso de prueba. Generar reporte discapacidad.....	57
Tabla # 8: No conformidades del sistema	59

Introducción

La humanidad se ha visto favorecida notablemente con el desarrollo y avance de la ciencia. En épocas anteriores se hacía prácticamente imposible realizar cualquier tipo de adelanto en la medicina debido al pobre desarrollo existente en la tecnología. Con el decursar de los años, el país en aras de elevar la calidad de vida de las personas con ciertas enfermedades y trastornos como las discapacidades, ha desarrollado programas de Genética Médica que han favorecido a la equidad de las personas con este tipo de trastornos con respecto al resto de la población.

La Genética Médica juega un rol muy importante en la evolución de la especie y en la erradicación de las enfermedades genéticas. Es el campo de las ciencias biológicas que trata de comprender cómo la herencia es transmitida de una generación a la siguiente y cómo se efectúa el desarrollo de las características que controlan este proceso. (1)

Esta rama debe su progreso a la continua y ascendente proyección de la informática, debido al desarrollo e implementación de herramientas informáticas que han permitido el acceso, manejo y procesamiento de todas las investigaciones de la genética. La fusión de estas dos ciencias ha posibilitado el desarrollo de aplicaciones médicas en beneficio de la sociedad, al permitir al sector de la salud, no sólo contar con métodos novedosos, sencillos y eficaces, sino también disponer de complejas aplicaciones que reducen la posibilidad de error en el diagnóstico de las enfermedades al ganar un tiempo vital para el paciente. Ejemplo de ello, son los estudios de la genética poblacional que analizan las características de las personas con determinadas enfermedades hereditarias y que contienen un gran volumen de información, permitiendo inferir datos valiosos para desarrollar nuevas formas de curar o prevenir las enfermedades que afligen a los pacientes.

Con el fin de llevar un mejor control y asesoramiento de los problemas genéticos, surge el 5 de agosto del año 2003 el Centro Nacional de Genética Médica (CNGM). Este centro constituye la sede de referencia nacional para el Programa Cubano de Diagnóstico, Manejo, Prevención de Enfermedades Genéticas y Defectos Congénitos, dirige la red de genética médica del país, cumpliendo tareas asistenciales, docentes e investigativas en el campo de la genética médica y la inmunología. (2)

En el año 2008 la Universidad de las Ciencias Informática (UCI) conjuntamente con el CNGM decidieron

desarrollar el Sistema Informático de Genética Médica (alasMEDIGEN), una herramienta para la gestión de la información asociada a una consulta de genética médica y a los estudios que el centro realiza.

El sistema está constituido por 9 módulos, los cuales permiten la gestión de toda la información que se genera a partir de la atención a personas con problemas genéticos. Los módulos del sistema son:

- Módulo Registro Cubano de Historias Clínicas (RECUHCL).
- Módulo Registro Cubano de Enfermedades Genéticas (RECUEGEN).
- Módulo Registro Cubano de Discapacitados (RECUDIS).
- Módulo Registro Cubano con Discapacidad Intelectual (RECUDI).
- Módulo Registro Cubano de Malformaciones Congénitas (RECUMAC).
- Módulo Registro Cubano de Gemelos (RECUGEM).
- Módulo de Teleconsulta Genética.
- Módulo Registro Cubano de Anomalías Cromosómicas (RECUAC).
- Módulo Registro Cubano de Enfermedades Comunes (RECUEC).

Los módulos Registro Cubano de Discapacitados y Registro Cubano de Discapacidad Intelectual permiten que los especialistas puedan gestionar la información de las personas discapacitadas a nivel nacional y provincial de forma rápida y segura. A pesar de ello, no es posible obtener toda la información que el genetista precisa de las personas que tienen algún tipo de discapacidad ya que solamente existen nueve reportes diseñados para estos registros, los cuales representan un bajo por ciento de la información demandada por los genetistas. Un ejemplo de los reportes que no se pueden obtener del sistema alasMEDIGEN son los tipos de discapacidad, si tienen amparo filial, la ocupación actual, el nivel de discapacidad, las madres que consumen alcohol o tabaco, el estado de la vivienda, si tienen capacidad laboral, entre otros. Esta situación impide realizar un estudio más detallado de los datos, así como realizar análisis estadísticos de la incidencia de enfermedades genéticas al impedir a los genetistas poder llevar un control adecuado de toda la información de las personas con discapacidades existentes. Además periódicamente se necesita obtener información de estos pacientes para evaluar el estado y el resultado de los diferentes programas que el país lleva a cabo para la atención a los discapacitados y poder elevar la calidad de vida de estas personas. Otras de las dificultades que poseen estos dos módulos es a la hora de visualizar los resultados de los reportes, los cuales al manejar muchos datos, impiden que los

genetistas puedan trabajar de forma rápida con toda la información, debido a que no cuentan con una representación gráfica de los discapacitados.

Luego de analizadas las dificultades existentes se define el **siguiente problema a resolver**: ¿Cómo contribuir a la obtención de información en el sistema alasMEDIGEN para facilitar el análisis de los estudios de discapacidad? Se establece como **objeto de estudio** el proceso de generación de reportes y constituye como **campo de acción** el proceso de generación de reportes de discapacidad en el sistema alasMEDIGEN.

En búsqueda de la solución al problema planteado se establece como **objetivo general**: Desarrollar nuevos reportes en los módulos Registro Cubano de Discapacitados y Registro Cubano con Discapacidad Intelectual del sistema alasMEDIGEN que permitan obtener información para un análisis favorable sobre los estudios de discapacidad, del cual se derivan los siguientes **objetivos específicos**:

- Seleccionar las herramientas y tecnologías para los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Analizar los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Diseñar los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Implementar los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Verificar el funcionamiento de los nuevos reportes de discapacidad del sistema alasMEDIGEN.

Para solucionar el problema planteado y lograr con éxito el cumplimiento de los objetivos, se definieron las siguientes **tareas de la investigación**:

- Estudio del estado del arte de las herramientas y tecnologías a utilizar para la generación de los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Análisis de la arquitectura del sistema alasMEDIGEN.
- Especificación de los requisitos funcionales de los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Especificación de los casos de uso de los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Especificación de los diagrama de clases del diseño de los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Especificación de los diagramas de componentes de los nuevos reportes de discapacidad del

sistema alasMEDIGEN.

- Implementación de los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Diseño de los casos de prueba de caja negra y caja blanca correspondientes a los nuevos reportes de discapacidad del sistema alasMEDIGEN.
- Ejecución de las pruebas de caja negra y caja blanca a los nuevos reportes de discapacidad del sistema alasMEDIGEN.

Este trabajo consta de cuatro capítulos:

Capítulo 1: Fundamentación Teórica

Constituye la base teórica de la investigación realizada, en la cual son expuestos los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión. Se describen los principales aspectos de las herramientas, tecnologías y metodologías a emplear durante el desarrollo de la misma.

Capítulo 2: Características del sistema

En este capítulo se aborda acerca del negocio en que está enmarcada la investigación. Se identifican los casos de usos, actores y trabajadores del negocio. Se muestran los diagramas de dominio y se realiza un levantamiento de los requisitos funcionales y no funcionales del sistema con el fin de lograr los objetivos planteados.

Capítulo 3: Diseño del sistema

En este capítulo se expone todo lo referente a esta fase de desarrollo, mostrándose todos los detalles que intervienen en el diseño de la solución: diagramas de clases, diagramas de interacción y despliegue. También se tocan aspectos como la arquitectura y patrones utilizados para el desarrollo.

Capítulo 4: Implementación y prueba

Se describe todo lo relacionado con los flujos de trabajo de implementación y prueba. Se ofrece una breve explicación de las pruebas realizadas y se ilustrarán los principales resultados obtenidos.

Capítulo 1: Fundamentación Teórica

Introducción

En este capítulo se brinda una investigación sobre los generadores de reportes en cuanto a sus características, importancia y ventajas. Se analiza la técnica de visualización de datos mediante el gráfico de barras. Además se hace necesario el estudio de la metodología de desarrollo RUP y las herramientas a utilizar para los nuevos reportes del Registro Cubano de Discapacitados y el Registro Cubano con Discapacidad Intelectual. Se efectuó un análisis de las diferentes bibliotecas de clases para graficar reportes en aras de visualizar la información con el propósito de coadyugar la toma de decisiones de los genetistas.

1.1 Desarrollo de los estudios genéticos en Cuba

Cuba ha producido una verdadera revolución en la extensión de los servicios de genética a la comunidad y a propuesta de Fidel Castro Ruz, el Centro Nacional de Genética Médica (CNGM) llevó a cabo un estudio social de todas las personas con discapacidades en todo el país, donde un grupo de médicos especialistas visitaron casi un 100% de la población cubana afectada. Entre los datos que se recopilan en un estudio determinado se destacan los datos primarios y relativos de la vida social del paciente, conjuntamente con los datos médicos específicos, necesarios para la definición del diagnóstico, tratamiento y seguimiento. Dicho cúmulo de información es almacenado y gestionado en expedientes no digitalizados, lo que constituye un problema a la hora de poder encontrar y/o actualizar alguna planilla en particular. Estas planillas son almacenadas en cada uno de los municipios y provincias, lo cual obliga a que si un paciente sea remitido a una consulta al nivel provincial o nacional, el mismo debe portar su historia clínica con los riesgos que ello implica. Todo esto trae consigo la imposibilidad de consultar casos a distancia y la falta de intercambio de la información que se genera en cada uno de los estudios, por parte de la comunidad genética médica y científica nacional.

Partiendo de las dificultades presentadas anteriormente el CNGM en colaboración con la UCI desarrollaron alasMEDIGEN para la gestión digital de toda la información recopilada por los estudios genéticos desarrollados en el país. También en aras de seguir ayudando a estos pacientes se continuaron

realizando estudios pero esta vez a las personas con malformaciones congénitas, a gemelos. Luego fue tomada esta experiencia adquirida en Cuba y fue aplicada en países de Latinoamérica como Venezuela, Ecuador, Nicaragua y Bolivia aunque en estos estudios la recogida de información no es igual a la de alasMEDIGEN.

1.2 alasMEDIGEN: Sistema Informático de Genética Médica

El sistema fue desarrollado sobre tecnología web utilizando el lenguaje de programación PHP 5.2 y el framework Symfony en su versión 1.0. El servidor web que se utiliza para la publicación del sistema es Apache 2.2.16 y la base de datos está soportada sobre MySQL 5.0. Dentro de las utilidades que brinda Symfony 1.0 se seleccionó Propel como sistema para mapear la base de datos en objetos relacionales. El desarrollo de la aplicación fue llevado a cabo utilizando la metodología de desarrollo Proceso Unificado de Rational (RUP) y se utilizó Lenguaje Unificado de Modelado (UML) para el modelado del sistema.

AlasMEDIGEN se encuentra integrado con el Registro Informatizado para la Salud (RIS). Este sistema creado en el marco de la informatización del Sistema de Salud Cubana, tiene como objetivo fundamental digitalizar y concentrar toda la información común que utilizan las diferentes entidades del Ministerio de Salud Pública de la República de Cuba (MINSAP). Consta de un grupo de registros importantes entre los que se encuentran el Registro del Ciudadano (RC), el Registro de Unidades de Salud (RUS), el Registro de Ubicación Geográfica (RU), el Registro de Localidad (RL), el Registro Personal de la Salud (RPS) y el Sistema de Autenticación, Autorización y Auditoría (SAAA), a los cuales se integra el sistema alasMEDIGEN a través del consumo de los servicios web que estos brindan y de esta manera se obtiene la información estandarizada.

A continuación se describen las funcionalidades que brindan los registros del RIS utilizadas por el sistema:

- El RC permite buscar los pacientes a través de diferentes criterios de búsqueda, si el paciente no se encuentra registrado, a través de alasMEDIGEN se pueden insertar los datos primarios del paciente en el RC. También posibilita obtener la cantidad de pacientes de un municipio o de una provincia.
- El RUS permite obtener los hospitales a través de diferentes criterios de búsqueda ya sea a nivel municipal o provincial, también las unidades y las áreas de salud.
- El RU permite listar los municipios y las provincias del país, obtener la provincia dado un municipio,

buscar las localidades de un municipio o de una provincia, buscar las calles dado una localidad, una manzana, un municipio o una provincia y listar los países.

- El RL permite buscar los Consejos Populares dado una localidad, un municipio o una provincia.
- El RPS permite buscar a un trabajador de la salud a través de diferentes criterios de búsqueda.
- El componente de seguridad SAAA permite controlar los niveles de acceso al sistema. A través del servicio Autenticar dado un usuario y una contraseña, el sistema recibe un certificado digital, el identificador del usuario y el nivel de acceso del mismo. Para cada petición que se haga a un componente del RIS, debe enviársele el certificado digital, el cual contiene la autorización para recibir el servicio.

AlasMEDIGEN está desarrollado sobre el patrón arquitectónico Modelo Vista Controlador (MVC) debido que el framework que se utilizó para el desarrollo del sistema implementa este patrón. Es una solución novedosa para Cuba que integra en un solo sistema un grupo de registros y estudios que hoy realiza el CNGM. Contiene la información genética de los individuos del país, lo que significa una invaluable fuente de información para la investigación y la toma de decisiones en función del beneficio de la sociedad cubana. A partir de las conclusiones que brinda el sistema, su información aporta una mejor prevención y tratamiento de las enfermedades genéticas, que son gestionadas a través de los reportes existentes.

Los reportes están asociados a las anomalías cromosómicas, enfermedades comunes, malformaciones congénitas, discapacitados, gemelos, historias clínicas, enfermedades genéticas, discapacidad intelectual, los cuales se encuentran distribuidos por módulos según correspondan. El sistema cuenta con nueve módulos donde cada uno de ellos permite gestionar los datos correspondientes a los diferentes estudios que se realizan en el CNGM, permitiendo insertar y modificar la información relacionada con cada uno de los estudios, además de realizar determinados reportes. Los principales beneficios que brinda son: una mayor agilidad en la gestión de los estudios de genética y una mayor facilidad para el proceso de monitoreo de la información que se maneja. Los módulos del sistema son: Módulo Registro Cubano de Historias Clínicas (RECUHCL), Módulo Registro Cubano de Enfermedades Genéticas (RECUEGEN), Módulo Registro Cubano de Malformaciones Congénitas (RECUMAC), Módulo Registro Cubano de Gemelos (RECUGEM), Módulo de Teleconsulta Genética, Módulo Registro Cubano de Anomalías Cromosómicas (RECUAC), Módulo Registro Cubano de Enfermedades Comunes (RECUEC), Módulo Registro Cubano de Discapacitados (RECUDIS) y el Módulo Registro Cubano con Discapacidad Intelectual (RECUDI).

Estos dos últimos módulos constituirán un importante aporte al desarrollo de la genética en Cuba al permitir a los especialistas poder gestionar la información de los pacientes que presentan algún tipo de discapacidad mediante los siguientes reportes:

RECUDIS:

- Reporte por Sexo: Muestra la cantidad de personas discapacitadas según su sexo y lista los datos de estas personas.
- Reporte por Tipo de Discapacidad: Muestra la cantidad de personas discapacitadas según su tipo de discapacidad y lista los datos de estas personas.
- Reporte por Amparo: Muestra la cantidad de personas discapacitadas según su amparo filial y los nombres de estas personas.
- Reporte por Capacidad Laboral: Muestra la cantidad de personas discapacitadas según su capacidad laboral y lista los datos de estas personas.
- Reporte por Ocupación: Muestra la cantidad de personas discapacitadas según su ocupación y lista los datos de estas personas.
- Reporte por Vínculo laboral: Muestra la cantidad de personas discapacitadas según su vínculo laboral y lista los datos de estas personas.

RECUDI:

- Cantidad de personas según el diagnóstico de retraso mental: Determina la cantidad de personas con retraso mental que existen por provincia según el diagnóstico de su causa de retraso mental.
- Cantidad de personas según la atención en algún servicio de genética: Determina la cantidad de personas con retraso mental que existen por provincias según la atención en algún servicio de genética.
- Visitas a los centros: Muestra las visitas a los centros según el nombre de la institución y el total de visitas realizadas al mismo.

El flujo de información del sistema comienza cuando el genetista desea registrar los datos del paciente. El sistema es el encargado de buscarlo y una vez seleccionado, se puede proceder a la inserción. Posteriormente los datos de los pacientes que son guardados en el sistema pueden ser modificados. Para poder obtener algún reporte, el especialista selecciona el criterio y el nivel deseado, al mostrar por parte

del sistema, el resultado que puede ser numérico o un listado en correspondencia del reporte seleccionado.

AlasMEDIGEN ha contribuido al desarrollo de las investigaciones de los genetistas, pero presenta como limitante que el Módulo Registro Cubano de Discapacitados (RECUDIS) tiene seis reportes y el Módulo Registro Cubano con Discapacidad Intelectual (RECUDI) tiene tres reportes. Estos reportes son un listado de las personas con determinadas características que permite realizar inferencias, asociaciones entre factores y enfermedades.

En estos dos módulos los reportes que existen son insuficiente ya que en Cuba hay 226 375 personas que presentan problemas genéticos, de las cuales se recogen alrededor de 200 características que constituyen posibles criterios de evaluación para el genetista. Los módulos RECUDI como RECUDIS presentan solamente nueve reportes, que muestran solo 15 características, imposibilitando obtener todos los datos necesarios para poder hacer una investigación científica que ayude a reducir el impacto de las enfermedades genéticas sobre la salud de la población o tomar cualquier decisión en aras de elevar la calidad de vida del paciente.

En estos dos módulos los reportes que existen son insuficientes ya que en Cuba hay 226 375 personas que presentan problemas genéticos, de las cuales se recogen alrededor de 200 características que constituyen posibles criterios de evaluación para el genetista. Los módulos RECUDI como RECUDIS presentan solamente a nueve reportes, que muestran solo 15 características, imposibilitando obtener todos los datos necesarios para poder hacer una investigación científica que ayude a reducir el impacto de las enfermedades genéticas sobre la salud de la población o tomar cualquier decisión en aras de elevar la calidad de vida del paciente.

Para dar solución a este problema se decide integrar al sistema 24 nuevos reportes, los cuales fueron distribuidos de la siguiente forma: 16 para el módulo RECUDI y 7 en RECUDIS. El flujo de estos módulos empezaría cuando el genetista solicita algún reporte. El sistema muestra el reporte escogido según el nivel del usuario. Si su nivel es nacional, puede visualizar todas las provincias del país, pero si es provincial solamente va a poder tener acceso a los municipios de la provincia del genetista. Luego tiene la opción de graficar en forma de barra el reporte, además de poder exportar la gráfica a pdf. Estos reportes

permitirán una mejor atención y tratamiento a los pacientes que presenten alguna discapacidad y mejorará la calidad de vida de todos los discapacitados del país.

1.3 Tecnologías y Herramientas

Las tecnologías de información y las herramientas que soportan el desarrollo de una aplicación, juegan un papel clave en su evolución. Representan el apoyo a la administración del proyecto, adaptándose a las características y objetivos propios del equipo de desarrollo. Seguido de la necesidad de seguir contribuyendo al desarrollo del sistema alasMEDIGEN, se realizó un análisis de las tecnologías y herramientas que dan solución al problema actual.

1.3.1 Reportes en el ámbito de la informática

Existen varios tipos de definiciones sobre lo que es un reporte. Una de las más citadas es:

Un reporte es un informe o una noticia. Este tipo de documento puede ser impreso o audiovisual, pretende transmitir una información, aunque puede tener diversos objetivos (3). Específicamente en el ámbito de la informática un reporte es “Un Documento generado por el Sistema, que nos presenta de manera Estructurada y/o Resumida, datos relevantes guardados o generados por la misma aplicación de tal manera que se vuelvan útiles para los fines estipulados” (4). Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios. Los reportes tienen diversos niveles de complejidad, desde una lista o enumeración hasta gráficos mucho más desarrollados. Según la herramienta informática y la base de datos en cuestión, los reportes permiten la creación de etiquetas y la elaboración de facturas, entre otras tareas.

Para lograr obtener un reporte con calidad y según las necesidades del cliente, se hace necesario el estudio de algunos generadores de reportes. Estos no son más que una herramienta informática que mediante la extracción de los datos almacenados en los sistemas de gestión de la información permiten crear reportes de cualquier proyecto en aras de facilitar la toma de decisiones en diversas instituciones.

1.3.2 Sistemas Generadores de Reportes

Crystal Reports

Crystal Reports es un producto de alta tecnología para la creación e integración de reportes con datos provenientes de múltiples fuentes de datos (bases de datos). Logra transformar rápidamente cualquier fuente de datos en contenido interactivo al brindar la posibilidad a los usuarios de acceder e interactuar con los reportes a través de portales Web, dispositivos móviles y documentos de Microsoft Office. Una de las características que lo distingue del resto de generadores de reportes es la posibilidad de combinar datos y gráficos dentro del mismo informe de una forma natural.

Permite crear contenido interactivo con calidad de presentación en la plataforma .NET (Framework (Marco de trabajo) de Microsoft) brindando un rápido desarrollo de aplicaciones, debido a que es la herramienta de elaboración de informes estándar para Visual Studio .NET. En Crystal Reports la optimización de consultas SQL permite que el tiempo de generación de los informes sea mínimo, al proporcionar mayor velocidad a la hora de obtener los datos e indicadores deseados.

No obstante aunque es una potente herramienta, presenta como limitante que es privativo y de uso restringido mediante el pago de patente, por cuanto, no se hace efectiva su utilización. (5)

ActiveReports

ActiveReports cuenta con la habilidad de diseñar, crear y desplegar aplicaciones de reportes de forma fácil y rápida. Admite exportaciones de datos a todos los formatos de archivo habituales, como PDF, Excel, entre otros e incluye un diseñador de informes Visual Studio .NET y una potente API (Interfaz de programación de aplicaciones). Resulta un diseñador de reportes orientado a usuarios finales, al permitir su inclusión en las aplicaciones, con la finalidad de que los propios usuarios diseñen y modifiquen sus reportes. Incluye un control de gráficos que admite tipos comunes de gráficos en 2D y 3D y proporciona prestaciones gráficas avanzadas, además de exportación nativa a diversos formatos de imagen.

Tiene soporte multilingüe y es una plataforma de informes que está basada 100 % en la Web. Incluye un control de visor de informes que admite zoom, vista previa de informes, múltiples fichas para visualización de hipervínculos, vistas divididas de múltiples páginas, un panel de índice de contenidos, miniaturas, búsquedas de texto, anotaciones y personalización de barra de herramientas. Además está escrito en C #

y administrado por Visual Studio para que los programadores puedan aprovechar su conocimiento de C # o Visual Basic.NET al programar con ActiveReports.

Presenta como impedimento que se requiere tener instalado la plataforma Windows y la facultad 6 de la UCI tiene como política regulatoria tener instalado en todas las estaciones de trabajo el Sistema Operativo Linux. Se distribuye bajo licencia privativa, por lo que resulta imposible poder usarlo en el proyecto. (6)

JasperReports

JasperReports es una librería de clases de Java de código abierto y su propósito principal es crear reportes de una manera simple, flexible y dinámica. Es un potente motor para la generación de reportes y tiene la habilidad de crear informes para ser visualizados en la web o enviados a la impresora. Los reportes pueden ser exportados a una multitud de formatos como PDF, HTML, XML, entre otros. Permite incluir en los reportes imágenes y gráficos, además esta herramienta posee un amplio y expandible grupo de posibles fuentes de datos.

Algunas de las características más significativas de JasperReports es que puede ser utilizado en cualquier entorno o sistema operativo siempre que exista una implementación de la máquina virtual de Java para dicho entorno. Permite generar textos o imágenes de fondo para utilizarlo como marcas de agua con el propósito de identificar el reporte o simplemente por motivos de seguridad. Es libre y distribuido mundialmente bajo los términos de la Licencia Pública para Librerías GNU (*GNU Library Public License*) y está respaldado por una gran comunidad internacional de desarrollo, el proyecto JasperForge.org y la empresa JasperSoft Corporation. Permite que los desarrolladores provean datos en varias formas, lo cual significa que los desarrolladores pueden pasar datos a los reportes a través de parámetros. Además los reportes son capaces de presentar los datos de manera textual o a través de gráficos.

JasperReports presenta como inconveniente para la solución, que al ser una librería implementada en java en cualquier ordenador donde se ejecute la aplicación debe estar instalado la máquina virtual de java la cual no es necesaria debido a que se está implementando una aplicación web por lo que se decidió seguir investigando otros reporteadores.

Generador Dinámico de Reportes (GDR)

El Generador Dinámico de Reportes es un sistema desarrollado en la Universidad de las Ciencias Informáticas, en el centro de desarrollo de DATEC, es una aplicación multiplataforma, que permite la creación y edición de reportes. Permite a sus clientes consultar las bases de datos de sus organizaciones al poder generar reportes con la información que estos manejan, está basado en tecnología web, conserva una interfaz gráfica amigable y está orientado al usuario final.

Brinda funcionalidades avanzadas al hacer posible: crear gráficos de pastel, barra, puntos, curva y lineales en los reportes. Se encuentra en constante perfeccionamiento con el objetivo de que cada día, logre satisfacer a los clientes. Además proporciona soporte a todo el ciclo de vida de los reportes, provee accesibilidad a la información emitiendo reportes a través de un navegador Web u otros formatos estándares. Por otra parte permite personalizar al detalle la salida de los reportes y la forma en que la información será visualizada, igualmente presenta una arquitectura basada en componentes que distribuye sus funcionalidades por módulos. Dichas partes se complementan entre sí, haciendo más fácil la generación de reportes, puesto que aumentan la reutilización y reducen la redundancia de información.

El principal problema del GDR es que no permite tomar los resultados que arrojan las consultas, además de que no permite la realización de consultas múltiples y todos los reportes necesarios son de este tipo.

1.3.3 ORM Propel

Propel aunque no es un generador de reporte, es un mapeo de objetos –relacional (ORM, de sus siglas en inglés *Object-Relational Mapping*), es de código abierto para PHP5. Soporta MySQL, PostgreSQL, SQLite, MSSQL y Oracle. Ofrece a los desarrolladores de aplicaciones web las herramientas para trabajar con bases de datos de la misma manera que se trabaja con otras clases y objetos en PHP. La principal ventaja que posee es que está completamente integrado con Symfony y permite acceder a su base de datos mediante un conjunto de objetos, proporcionando una API sencilla para almacenar y recuperar datos. Está completamente documentado, con el respaldo de muchos tutoriales en toda la web. Además el código generado por Propel está bien comentado y fácil de usar.

Finalmente se decidió utilizar el ORM de Symfony que es Propel para la realización de los reportes demandado por los genetistas teniendo en cuenta que ninguno de los generadores de reportes antes

expuestos permiten la realización de los reportes necesarios a través de ellos. Además Propel es el ORM por defecto que utiliza Symfony hasta su versión estable 1.4 y en la solución se utiliza Symfony en su versión 1.0.22, esta es la razón por la cual se utiliza Propel y no otro ORM como por ejemplo Doctrine.

1.3.4 Técnica de visualización de datos mediante el gráfico de barras

Al concluir los reportes pertinentes, se utilizó la técnica de visualización de datos mediante los gráficos de barra, los cuales consisten en mostrar los grupos de valores y series como conjuntos de barras horizontales agrupadas por categoría. Los valores están representados por la longitud de las barras con relación al eje X y las etiquetas de las categorías se presentan en el eje Y. (7) Estos gráficos permiten reunir los datos en categorías (por ejemplo en diferentes tipos de discapacidad) y posibilita al genetista poder comparar los valores de cada categoría rápidamente, para realizar un análisis más detallado de los pacientes que padecen estas anomalías, además permiten obtener mejores resultados en las investigaciones de los especialistas.

1.3.5 Metodología de desarrollo

La metodología de desarrollo de software es un punto importante para cualquier proyecto que busca minimizar riesgos e incrementar las posibilidades de éxito en el desarrollo de productos informáticos. Muestra el camino a seguir para la obtención de un producto predecible y eficiente, que posea calidad a pesar de su tamaño y complejidad. Especifica quién debe hacer qué, cuándo y cómo, a través de la estructuración, planificación y control del proceso de desarrollo. En la actualidad no existe una metodología universal que se le pueda aplicar a todos los proyectos y su selección depende de las características que posea cada proyecto. Existen varias metodologías, pero a continuación solo se describe RUP, dada que fue la seleccionada para el desarrollo de la investigación.

Proceso Unificado de Desarrollo de Software (RUP)

El Proceso Unificado Racional o de Desarrollo tradicionalmente abreviado como RUP constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Utiliza el Lenguaje de Modelado Unificado (UML) para describir un sistema. El flujo de desarrollo de RUP está dividido en cuatro fases: inicio, elaboración, construcción y transición.

El ciclo de vida de RUP se caracteriza por:

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo representan la realización de los casos de uso.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo y describe los elementos del modelo que son más importantes para su construcción.
- Iterativo e incremental: Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo y los incrementos, al crecimiento del producto. (8)

Se definió el uso de RUP como metodología porque presenta la cualidad de ser altamente configurable. Facilita el desarrollo del proyecto al dividir el trabajo en partes más pequeñas o mini proyectos. Provee al equipo de la documentación detallada de todo el proceso de desarrollo, además es la metodología que se ha utilizado desde el inicio del proyecto.

1.3.6 Herramienta de Modelado CASE

Las herramientas CASE (*Computer Aided Software Engineering*, del español Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo del software para reducir el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida del desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto y cálculo de costos. Proveen mejoras en la calidad y productividad del diseño. (9) La herramienta CASE que se escogió fue Visual Paradigm en su versión 6.4.

Visual Paradigm 6.4

Visual Paradigm es una de las herramientas UML CASE que soporta el ciclo de vida completo del desarrollo del software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Reduce la duración del ciclo de desarrollo al brindar ayuda tanto a arquitectos, analistas, diseñadores y desarrolladores. Está diseñado para incorporar soporte para trabajo en equipo al permitir que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus

compañeros.

Comparte y genera documentación de diagramas y diseños en formatos PDF, HTML, Microsoft Work. Es multiplataforma y permite la generación de código e ingeniería inversa para los lenguajes: Java, C, C++, PHP, entre otros. También provee el modelado de procesos de negocios, además de un generador de mapeo de objetos-relacionales para los lenguajes de programación Java .NET y PHP. Presenta disponibilidad de múltiples versiones, siendo fácil de instalar y actualizar. Intercambia diagramas UML y modelos con otras herramientas. Soporta la Importación y Exportación a formatos XMI y XML y archivos Excel.

Por lo antes analizado en el trabajo de diploma se hace uso de Visual Paradigm en su versión 6.4 ya que es una herramienta CASE que permite construir diagramas UML, como son los flujos de eventos del sistema, las clases, todo lo que es documentación tanto de desarrollo como procesos de negocio.

1.3.7 Lenguaje de programación

El lenguaje de programación es el idioma artificial utilizado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo otras circunstancias. Está compuesto por un conjunto de símbolos, reglas sintácticas, semánticas que definen su estructura y el significado de sus elementos. Son utilizados para controlar el comportamiento físico y lógico de una máquina a través de la creación de programas, para lograr expresar algoritmos con precisión, o como modo de comunicación humana. Se determina que el lenguaje de programación seleccionado sea PHP5. (10)

PHP5

PHP es el acrónimo de *Hypertext Preprocessor*, es un lenguaje de programación interpretado que fue diseñado originalmente para la creación de páginas web dinámicas. Es multiplataforma y tiene la capacidad de interactuar con la mayoría de los motores de base de datos que se utilizan, tales como MySQL, PostgreSQL, Oracle, Informix, entre otras. Permite las técnicas de Programación Orientada a Objetos y está completamente escrito en C, por lo que a la hora de ejecutarse utiliza poca memoria. El código fuente es invisible al navegador y al cliente. Es el servidor el que se encarga de ejecutar el código y

enviar su resultado HTML al navegador, para lograr que la programación en PHP sea segura y confiable.

PHP tiene la capacidad de expandir su potencial al utilizar la enorme cantidad de módulos (llamados ext's o extensiones), a su vez posee una amplia documentación en su página oficial, en la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Brinda licencia de software libre y es posible ampliarlo con nuevas funcionalidades. Cuenta con una biblioteca de clases nativa de funciones sumamente amplia y ofrece la integración con varias bibliotecas externas, las cuales permiten que el desarrollador haga casi cualquier cosa, desde generar documentos en formato PDF hasta analizar código XML. (11)

Se decide hacer uso como lenguaje de programación PHP5 porque alasMEDIGEN está hecho con este lenguaje. También posee una gran variedad de funciones que pueden ser utilizadas para mejorar el rendimiento de los programas y se evidencia que es un lenguaje con una sintaxis muy simple y fácil de aprender, además el marco regulatorio del Grupo de Integración de Soluciones (GIS) establece como lenguaje de programación PHP para las aplicaciones de salud.

1.3.8 Framework de desarrollo

Un framework o marco de trabajo 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. Incluye soporte de programas, bibliotecas y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Es utilizado para simplificar y agilizar el proceso de desarrollo de aplicaciones. Permite reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones de diseño.

Symfony 1.0.22

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes al permitir al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Es compatible con la mayoría de gestores de bases de datos como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Proporciona varias herramientas y clases

encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Presenta algunas características como:

- Fácil de instalar y configurar en la mayoría de plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Está preparado para aplicaciones que son adaptable a las políticas y arquitecturas propias de cada proyecto, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros. (12)

Se decide usar Symfony en su versión 1.0.22 como marco de desarrollo porque el sistema alasMEDIGEN en todas sus versiones se ha desarrollado sobre este marco de trabajo, permite un mantenimiento muy sencillo, el código es fácil de leer y está publicado bajo una licencia de software libre.

1.3.9 Servidor Web

Un servidor web es un programa que se ejecuta continuamente en una computadora, manteniéndose en espera de solicitudes de ordenadores clientes a través de navegadores web y dándole repuestas a estas mediante páginas web que se muestran en dicho navegador.

Apache 2.0

Apache es un servidor web HTTP para plataformas Unix (BSD, GNU/Linux), Windows, Macintosh. Al ser una tecnología gratuita de código fuente abierto le da una transparencia inmediata, de forma que si quisieras ver qué se está instalando en el servidor, se puede saber, sin ningún secreto. Es un servidor altamente configurable de diseño modular, siendo muy sencillo ampliar las capacidades del servidor. Se caracteriza por su alta capacidad de configuración en la creación y gestión de registros, permitiendo la creación de ficheros de registros a medida del administrador, obteniendo así un mayor control sobre lo que sucede en su servidor.

Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto. Trabaja con gran

cantidad de lenguajes de script como Perl, PHP, entre otros. La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia permite hacer lo que desee con el código fuente siempre que sea reconocido su trabajo.

Se decidió utilizar el Servidor Web Apache 2.0 porque es un servidor web flexible, rápido, eficiente, de código abierto, multiplataforma, continuamente actualizado y adaptado a los nuevos protocolos. Asimismo tiene la particularidad de ser un servidor web muy popular lo cual permite cierta facilidad a la hora de conseguir ayuda y soporte para el mismo.

1.3.10 Sistema Gestor de Base de Datos(SGDB)

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Tiene como propósito el manejo sencillo de los datos y las entidades que los organizan con el fin de llegar más fácilmente a la información contenida dentro de la base de datos. Con la utilización del gestor de base de datos se logra: independencia de los datos y los programas de aplicación, minimización de la redundancia, integración y sincronización de las bases de datos, integridad, seguridad y protección de los datos, facilidad de manipulación de la información y control centralizado

MySQL 5.1

MySQL es un sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran cantidad de consultas de forma muy eficiente. Es probablemente el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Posee infinidad de bibliotecas de clases y herramientas que permiten su uso a través de gran cantidad de lenguajes de programación. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP). Tiene gran portabilidad entre sistemas y puede mezclar tablas de distintas bases de datos en la misma consulta. Es multiplataforma, de fuente abierta (fuente abierta significa que es posible, para cualquier persona, usarlo, modificarlo, incluso poder descargar el código fuente en internet y usarlo sin pagar). Tiene un sistema de privilegios y contraseñas que es muy flexible, seguro y que permite la verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor. (13)

Se decide utilizar como gestor de Base de Datos MySQL en su versión 5.1 por tener una probabilidad muy

reducida de corromper los datos, hasta en los casos en que los errores no se produzcan en el propio gestor, sino en el sistema que lo utiliza. Incluso su bajo consumo de recursos lo tornan apto para ser ejecutado en una máquina con escasos recursos además el Sistema Informático de Genética Médica se rige por el marco regulatorio del Grupo de Integración de Soluciones (GIS) de Softel, es que se decide hacer uso de este gestor en su versión más estable, la 5.1.

1.3.11 Entorno de desarrollo

Un Entorno de Desarrollo Integrado (IDE) es un entorno de programación creado como un programa de aplicación con un conjunto de herramientas para el programador: un editor de código, un compilador, un depurador y constructor de interfaz gráfica de usuario. Pueden ser aplicaciones independientes o pueden formar parte de aplicaciones existentes. Proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación, es multiplataforma, soporta múltiples idiomas, posee manual de usuarios y ayuda. El IDE escogido para la investigación fue el Netbeans 7.1.

NetBeans IDE 7.1

NetBeans es un entorno de desarrollo Integrado, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso. Además proporciona soporte de primera clase completo para las últimas tecnologías Java y las mejoras más recientes de Java antes de otros IDE.

NetBeans IDE 7.1 es la última versión estable del IDE que ofrece características para ayudar en la construcción, mantenimiento de aplicaciones web y móviles. Brinda un rendimiento superior para C, C++ y los desarrolladores de PHP, al proporcionar editores, tecnologías relacionadas y herramientas integrales para sus marcos como son XML, HTML, PHP, JavaScript, entre otras. Está bajo la Licencia de Distribución y Desarrollo Común (CDDL) y la Licencia Pública General (GPL) de GNU versión 2. (14)

El IDE de desarrollo que se escogió fue el NetBeans con su versión 7.1 al tener en cuenta que el framework utilizado es Symfony y el IDE antes mencionado presenta módulos de soporte para dicho framework. Además ofrece completamiento de código para las variables de Symfony.

1.3.12 Bibliotecas de Clases

Una biblioteca de clases es un conjunto de subprogramas o fragmentos de códigos que contienen funciones que son frecuentemente utilizadas por los programadores, ubicados en un solo paquete. Entre las bibliotecas de clases para graficar se encuentran:

JpGraph

JpGraph es una librería que incluye una serie de clases de código orientado a objetos que sirven para crear imágenes con todo tipo de gráficas dinámicamente desde páginas PHP. Algunas de sus características es que reduce peso en bytes de las imágenes que utiliza, habitualmente unos pocos Kilobytes (KB). Posee diversos tipos de gráficas 2D o 3D, como de puntos, líneas, tartas, barras, cajas, entre otros.

JpGraph es configurable con distintos tipos de colores, leyendas, tipografías, imágenes de fondo. Contiene soporte para generar gráficas con varios juegos de valores a la vez. Presenta escalas flexibles tanto en el eje X como el eje Y, ajustándose al juego de datos que se quiera representar. Es una biblioteca de clases de fácil manejo y simple de instalar. Incluye una completa documentación con varios ejemplos de los distintos gráficos que se pueden generar y tutoriales para aprender a manejarlo. (15).

Aunque es una herramienta muy potente en la elaboración de gráficos para las páginas web, los mensajes de error son escasos y la mayoría de las veces cuando algo falla no se obtiene ninguna explicación. Otro impedimento es que no es gratuita para fines comerciales, por ello no es factible utilizarlo para la solución.

CanvasXpress

CanvasXpress es una biblioteca de clases para JavaScript e implementada en HTML5. Brinda gráficos de barras, líneas, área, circulares, entre otros. Está disponible bajo los términos de la licencia de código abierto, Licencia Pública General Menor (LGPL) en su versión 3.0. Permite generar gráficos tridimensionales, que se puede rotar libremente para una mejor perspectiva.

CanvasXpress está bastante bien optimizado y tiene un uso sencillo para permitir personalizar absolutamente todos los aspectos tales como los colores, el texto o los elementos. Funciona en los principales navegadores como Firefox, Safari, Chrome, Opera e Internet Explorer (IE). Tiene como

desventaja que IE debe tener incluido una biblioteca de clases adicional para poder interpretar sus gráficas que es ExplorerCanvas, por este motivo se decidió no hacer uso de la misma.

Highcharts 2.2.2

Highcharts es una biblioteca de gráficos escritos solo en JavaScript, es ideal para crear aplicaciones web en cualquier lenguaje (PHP, ASP.NET). Dispone de funcionalidades como por ejemplo Zoom o tooltips, lo cual le brindan una gran ventaja competitiva frente a otras soluciones de gráficos en JavaScript. Con el módulo de exportación habilitado, los usuarios pueden exportar el gráfico PNG, JPG, PDF, SVG o imprimirlo directamente desde la página web.

Tiene una configuración sencilla y permite cargar datos externos para los gráficos. Además a través de su API se pueden agregar, eliminar, modificar la serie, ejes o puntos en cualquier momento después de crear la gráfica. Incluye una gran colección de gráficos para aplicaciones web enriquecidas como: gráficos de torta, de dispersión, simples, dinámicos, animaciones o animaciones en gráficos, entre otros.

Por las ventajas antes mencionadas se decide utilizar Highcharts en su versión 2.2.2

1.3.13 Patrones de arquitectura

Los patrones arquitectónicos ayudan a especificar la estructura fundamental de una aplicación. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y pautas para la organización de las relaciones entre ellos. Facilitan la documentación de diseños arquitectónicos y son de gran ayuda para controlar la complejidad de un diseño.

Dentro de los diversos patrones de arquitectura existentes, cabe destacar uno de los más significativos: el patrón Modelo Vista Controlador (MVC).

Patrón modelo vista controlador (MVC)

El patrón MVC divide la lógica de la aplicación en tres componentes: modelo, vista y controlador:

El Modelo: Es la representación específica de la información con la cual el sistema opera. Maneja los datos y controla todas sus transformaciones. Envía a la vista aquella parte de la información que en cada

momento se le solicita para que sea mostrada. Además las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

Vista: Recibe los datos del modelo y genera una representación visual para que el usuario pueda interactuar con los datos.

Controlador: Es la capa que controla todo lo que puede realizar la aplicación. Responde a eventos, usualmente acciones del usuario. Invoca peticiones al modelo cuando se hace alguna solicitud sobre la información. También puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta de modelo.

Conclusiones

En este capítulo se adentró en los principales conceptos vinculados al tema de investigación lo cual permitió una mayor comprensión del mismo. Se realizó el análisis de varios generadores de reportes que aunque no se utilizaron en la solución sirvieron para conocer las características que poseen cada uno y las opciones que brindan cuando diseñan y crean un reporte. Se seleccionaron las herramientas y metodologías para el desarrollo de los reportes del módulo de discapacitados alasMEDIGEN. Estos fueron: RUP como metodología de desarrollo, Visual Paradigm 6.4 para el modelado de los datos, PHP 5 como lenguaje de programación, Symfony 1.0.22 como marco de desarrollo, Apache 2.2 para el servidor de aplicaciones, MySQL 5.1 para la gestión de la base de datos, NetBeans 7.1 como entorno de desarrollo. También se seleccionó Highcharts 2.2.2 como biblioteca de clases para graficar lo que permite realizar la solución en menor tiempo y más robusta. Además se definió el patrón arquitectónico MVC para lograr un diseño y una implementación con la calidad requerida.

Capítulo 2: Características del Sistema

Introducción

En el presente capítulo se abordan los temas referentes a las reglas del negocio, a través de un modelo de dominio con su descripción pertinente. También se detallan los requisitos funcionales y no funcionales. Se especifican los actores del sistema, casos de usos del sistema, diagramas de casos de usos del sistema, así como la descripción textual y ampliada de los casos de usos del sistema.

2.1 Modelo de dominio

Un modelo del dominio captura los tipos de objetos más importantes que existen, o los eventos que suceden en el entorno donde tendrá incidencia el sistema, comprende elementos conceptuales, se definen dichos elementos y se unen o relacionan en un diagrama de clases UML. El objetivo fundamental de este modelo es comprender y describir los conceptos más importantes dentro del contexto del sistema. Es una representación visual del entorno real del proyecto y contribuye a la comprensión de los requisitos del sistema que se desprenden de este contexto.

Descripción de las clases del modelo del dominio.

Clases	Descripción
Genetista	Especialista que interactúa con el sistema solicitando información a través de un reporte.
alasMEDIGEN	Sistema Informático que contiene diferentes módulos para gestionar la información. Se encarga de generar el reporte solicitado por el genetista.
Reporte	Muestra la información solicitada por el genetista.
Gestor BD	Gestor de base de datos que recibe la solicitud de una consulta para la posterior exhibición de los resultados.

Tabla # 1: Descripción de las clases del modelo de dominio.

Diagrama de clases del modelo del dominio

El genetista interactúa directamente con el sistema alasMEDIGEN seleccionando el reporte que desea analizar. Se realiza una consulta a la base de datos y luego le son mostrados los resultados al genetista para que realice las acciones estimadas.

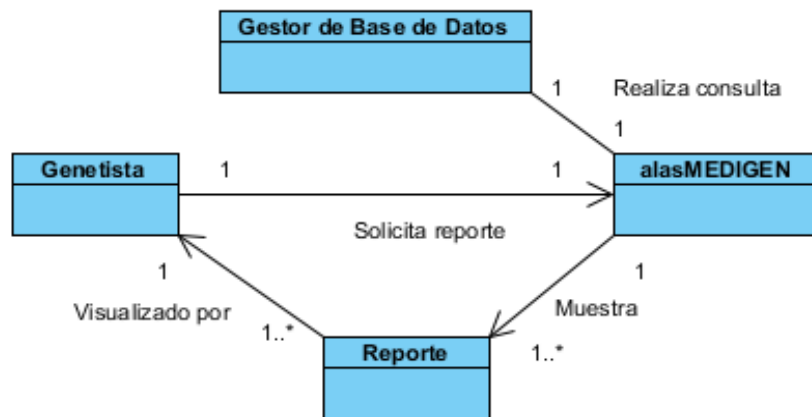


Figura # 1: Diagrama de clases del modelo de dominio

2.2 Especificación de los requisitos del sistema

Los requisitos no son más que la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. El propósito de esta sección es identificar y documentar lo que en realidad se necesita, en una forma clara y entendible.

Requisito funcional

Los requisitos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir. Además definen las funciones que el sistema será capaz de realizar.

- RF1 Generar Reporte de Discapacidad por Sexo
- RF2 Generar Reporte de Discapacidad por Tipo de Discapacidad
- RF3 Generar Reporte de Discapacidad por Reporte por Amparo Filial
- RF4 Generar Reporte de Discapacidad por Rango de Edades

RF5 Generar Reporte de Discapacidad por Evaluación Funcional
RF6 Generar Reporte de Discapacidad por Cantidad de Consumo
RF7 Generar Reporte de Discapacidad por Capacidad Laboral
RF8 Generar Reporte de Discapacidad por Ocupación
RF9 Generar Reporte de Discapacidad por Vínculo Laboral
RF10 Generar Reporte de Discapacidad por Factor de Discapacidad
RF11 Generar Reporte de Discapacidad por Nivel de Discapacidad
RF12 Generar Reporte de Discapacidad Atendidos por Instituciones del MINED
RF13 Generar Reporte de Discapacidad por Estado de la Vivienda
RF14 Generar Reporte de Discapacidad por Madres que consumen alcohol o tabaco
RF15 Generar Reporte de Discapacidad por Asociación de Discapacitados
RF16 Generar Reporte de Discapacidad por Miembro paralizado
RF17 Generar Reporte de Discapacidad Intelectual por Diagnóstico
RF18 Generar Reporte de Discapacidad Intelectual por Atención
RF19 Generar Reporte de Discapacidad Intelectual por Rango de Edades
RF20 Generar Reporte de Discapacidad Intelectual por Total de Discapacitados
RF21 Generar Reporte de Discapacidad Intelectual por Tipos de Etiología Genética
RF22 Generar Reporte de Discapacidad Intelectual por Eventos Perinatales
RF23 Generar Reporte de Discapacidad Intelectual por Malformaciones Internas
RF24 Generar Reporte de Discapacidad Intelectual por Malformaciones Externas
RF25 Graficar Reporte
RF26 Exportar gráfica a PDF

Requisitos no funcionales

Los requisitos no funcionales son las propiedades emergentes que va a poseer el sistema y que harán del mismo un producto seguro, confiable y de respuesta rápida.

RNF1 Requisitos de usabilidad

El sistema es entendible y fácil de usar debido a que presenta al usuario una interfaz atractiva e interactiva, con un menú general que lo guía en la búsqueda de la información que necesita. Además

todos los módulos aparecerán en el menú izquierdo para que el usuario pueda ir rápidamente. Son auto desplegable, por sección y satisface las necesidades de los usuarios.

RNF2 Requisito de fiabilidad

El sistema estará disponible en Infomed las 24 horas del día y los siete días de la semana. En caso de fallo, pudiera estar fuera de servicio por un período de 72 horas máximo. La precisión y exactitud de las salidas del sistema se corresponden con la calidad y exactitud de la información contenida en las base de datos desde donde se extraen los datos de los reportes. La información manejada por el sistema es protegida contra la corrupción. Se permite la creación de copias de respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información. El sistema lleva a cabo la validación de los formularios para lograr una mayor confianza sobre los datos con que opera. El funcionamiento básico de este mecanismo consiste en que si el usuario introduce datos no válidos y envía el formulario, la próxima página que se muestra contiene los datos con los mensajes de error, posibilitando así al usuario arreglarlos antes de volver a enviar el formulario.

RNF3 Requisito de soporte

Se asegura 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. Además se hace un mantenimiento del sistema anual.

RNF4 Requisito de hardware

La infraestructura de hardware permite aumentar la cantidad de servidores o adicionar componentes de hardware en función de disminuir la carga, sin que sea necesario realizar modificaciones al software. Para el desarrollo y ejecución de la aplicación propuesta se necesita como mínimo:

Para el servidor de aplicación:

- Microprocesador Pentium IV a 2.5 GHz o superior.
- 1Gb de RAM o superior.
- Mínimo 5Gb de espacio libre para la base de datos en un inicio (después dependerá del nivel de crecimiento de la base de datos).

Para el Cliente:

- Microprocesador a 800 MHz o superior
- 512 MB de RAM.
- Tarjeta de red (MODEM o red con TCP-IP para conexión al servidor.)

RNF5 Requisito de software

Se requiere disponer para el funcionamiento del sistema de un servidor que cuente con Sistema Operativo Linux y servidores Apache 2.0 o superior y MySQL 5.0 o superior. Los usuarios del sistema deberán contar con un navegador capaz de interpretar código JavaScript, como: Internet Explorer 5.5 o Mozilla Firefox 2.0 o superiores, para poder acceder a las opciones que brinda el sistema. Además en la PC cliente se deberá disponer del Sistema Operativo Linux o Windows 98 o una versión superior.

RNF6 Interfaz

Interfaces de usuario:

Las ventanas del sistema contienen claro y bien estructurados los datos, además permiten la interpretación correcta de la información. La entrada de datos incorrecta es detectada e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español. El diseño de la interfaz del sistema responde a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso. La interfaz es sencilla, amigable, intuitiva y de fácil navegación por el usuario.

Interfaces de comunicación

Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo TCP/IP, HTTP.

RNF7 Requisitos legales, de derecho de autor y otros

La UCI posee todos los derechos sobre la aplicación. Las herramientas y las tecnologías en que está basado el proyecto deberán cumplir con la licencia de GNU/GPL.

RNF8 Requisitos de apariencia o interfaz externa

- El sistema tiene una interfaz fácil de usar y amigable para que pueda ser utilizada sin mucho entrenamiento por el usuario.
- Empleo de imágenes y colores identificados con el negocio donde se implantará el sistema.
- Estará diseñado para resolución de 800x600, aunque deberá verse en cualquier resolución superior a ésta.

RNF9 Requisitos de seguridad

El sistema tiene un mecanismo para gestionar la seguridad a través de niveles de acceso a la información. Estos vienen dados por el SAAA y el Sfguard Plugin de Symfony. El primero es un componente de seguridad implementado por la dirección de informatización de salud, donde cada usuario desempeña un papel de visualizador o editor nacional, provincial o municipal. Sus siglas significan seguridad, autenticación, autorización y auditoria en el sistema. El segundo ofrece funciones de autenticación y autorización permitiendo delimitar el acceso a la aplicación y asegurándola para diferentes tipos de usuarios.

RNF10 Requisito de 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.

RNF11 Rendimiento

Los tiempos de respuestas y velocidad de procesamiento de la información son rápidos, no mayores de 10 segundos teniendo en cuenta las diferentes consultas que se ejecutan para los reportes.

2.3 Definición de los casos de uso del sistema

El modelo de casos de uso del sistema representa las relaciones existentes entre actores y casos de uso. Representa un esquema que recoge las funcionalidades principales del sistema y determina el uso que tendrá desde el punto de vista del usuario, ya que su construcción es en base a las necesidades del

mismo. También muestra los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

Además cada caso de uso tiene una descripción que especifica la funcionalidad que se incorporará al sistema propuesto y los actores son terceros fuera del sistema que interactúan con él (puede ser cualquier persona, individuo, grupo, entidad, organización, máquina o sistema de información externo; con los que el sistema interactúa).

Diagrama de casos de uso del sistema

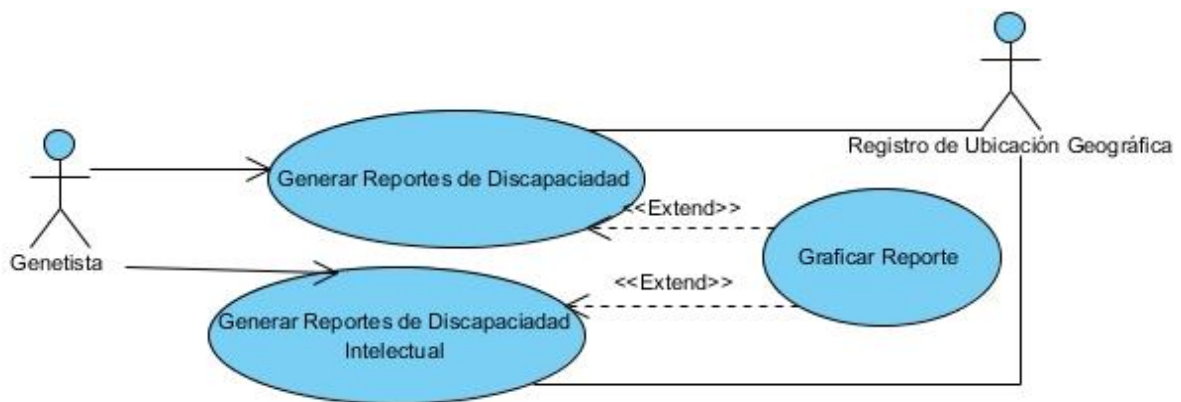


Figura # 2: Diagrama de caso de uso del sistema.

Actores del sistema

Actor	Descripción
Genetista	Representa a los genetistas a nivel nacional y provincial que es el encargado de visualizar los reportes.
Registro de Ubicación Geográfica	Representa al componente del RIS para obtener el listado de las provincias y los municipios.

Tabla # 2: Actores del sistema.

Descripción de casos de uso

Caso de Uso:	Generar Reportes de Discapacidad	
Actores:	Genetista (Inicia el caso de uso).	
Resumen:	El caso de uso inicia cuando el genetista solicita algún reporte de discapacidad, luego se muestra el reporte deseado de acuerdo al criterio escogido, finalizando así el caso de uso.	
Precondiciones:	El Genetista debe estar autenticado para acceder al sistema.	
Referencias	RF1, RF2, RF3, RF4, RF5, RF6, RF7, RF8, CU Graficar Reporte (Extendido).	
Prioridad	Crítica	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El Genetista selecciona cualquiera de las siguientes opciones:</p> <ul style="list-style-type: none"> • Generar Reporte de Discapacidad por Sexo. • Generar Reporte de Discapacidad por Tipo de Discapacidad. • Generar Reporte de Discapacidad por Reporte por Amparo Filial. • Generar Reporte de Discapacidad por Rango de Edades. • Generar Reporte de Discapacidad por Evaluación Funcional. • Generar Reporte de Discapacidad por Cantidad de Consumo. • Generar Reporte de Discapacidad por Capacidad Laboral. • Generar Reporte de Discapacidad por Ocupación. • Generar Reporte de Discapacidad por Vínculo Laboral. • Generar Reporte de Discapacidad por Factor de Discapacidad. • Generar Reporte de Discapacidad por Nivel de Discapacidad. 	<p>2. El sistema muestra el reporte correspondiente según el nivel del usuario, finalizando así el caso de uso.</p>	

<ul style="list-style-type: none"> • Generar Reporte de Discapacidad Atendidos por Instituciones del MINED. • Generar Reporte de Discapacidad por Estado de la Vivienda. • Generar Reporte de Discapacidad por Madres que consumen alcohol o tabaco. • Generar Reporte de Discapacidad por Asociación de Discapacitados. • Generar Reporte de Discapacidad por Miembro paralizado. 	
---	--

Tabla # 3: Descripción del caso de uso del sistema. Generar reporte de discapacidad.

Caso de Uso:	Generar Reportes de Discapacidad Intelectual	
Actores:	Genetista (Inicia el caso de uso).	
Resumen:	El caso de uso inicia cuando el genetista solicita algún reporte de discapacidad intelectual, luego se muestra el reporte deseado de acuerdo al criterio escogido, finalizando así el caso de uso.	
Precondiciones:	El Genetista debe estar autenticado para acceder al sistema.	
Referencias	RF9, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20, RF21 RF22, RF23, RF24, CU Graficar Reporte (Extendido).	
Prioridad	Crítica	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
1.	<p>El Genetista selecciona cualquiera de las siguientes opciones:</p> <ul style="list-style-type: none"> • Generar Reporte de Discapacidad Intelectual por Diagnóstico. • Generar Reporte de Discapacidad Intelectual por Atención. • Generar Reporte de Discapacidad Intelectual por Rango de Edades. 	2. El sistema muestra el reporte correspondiente según el nivel del usuario, finalizando así el caso de uso.

- Generar Reporte de Discapacidad Intelectual por Total de Discapacitados.
- Generar Reporte de Discapacidad Intelectual por Tipos de Etiología Genética.
- Generar Reporte de Discapacidad Intelectual por Eventos Perinatales.
- Generar Reporte de Discapacidad Intelectual por Malformaciones Internas.
- Generar Reporte de Discapacidad Intelectual por Malformaciones Externas.

Interfaz del Reporte de Discapacidad Intelectual a nivel nacional

The screenshot shows the ALAS MEDIGEN web interface. The header includes the logo, the name 'ALAS MEDIGEN SISTEMA INFORMÁTICO DE GENÉTICA MÉDICA', and user information: 'Bienvenido Marcia Leticia Izaquirre Lorenzo al Sistema Informático de Genética Médica.' The user is identified as 'rev_nac_gm' with 'Derechos: Revisor' and 'Nivel: Nacional'. The main content area displays a report titled 'Reporte por diagnostico' with a table showing data for various provinces.

Reporte por diagnostico			
Provincia	Tiene diagnóstico de DI	No tiene diagnóstico de DI	Total
Pinar del Rio	3	0	3
Artemisa	2	0	2
La Habana	1	1	2
Mayabeque	0	0	0
Matanzas	0	1	1
Cienfuegos	0	0	0
Villa Clara	0	0	0
Sancti Spiritus	0	0	0
Ciego de Avila	0	1	1
Camagüey	0	0	0
Las Tunas	0	0	0
Holguín	0	0	0
Granma	1	0	1
Santiago de Cuba	0	0	0
Guantánamo	0	0	0
Isla de la Juventud	0	0	0
Total	7	3	10

Figura # 3: Interfaz del reporte de Discapacidad Intelectual a nivel nacional.

Interfaz del Reporte de Discapacidad Intelectual a nivel provincial

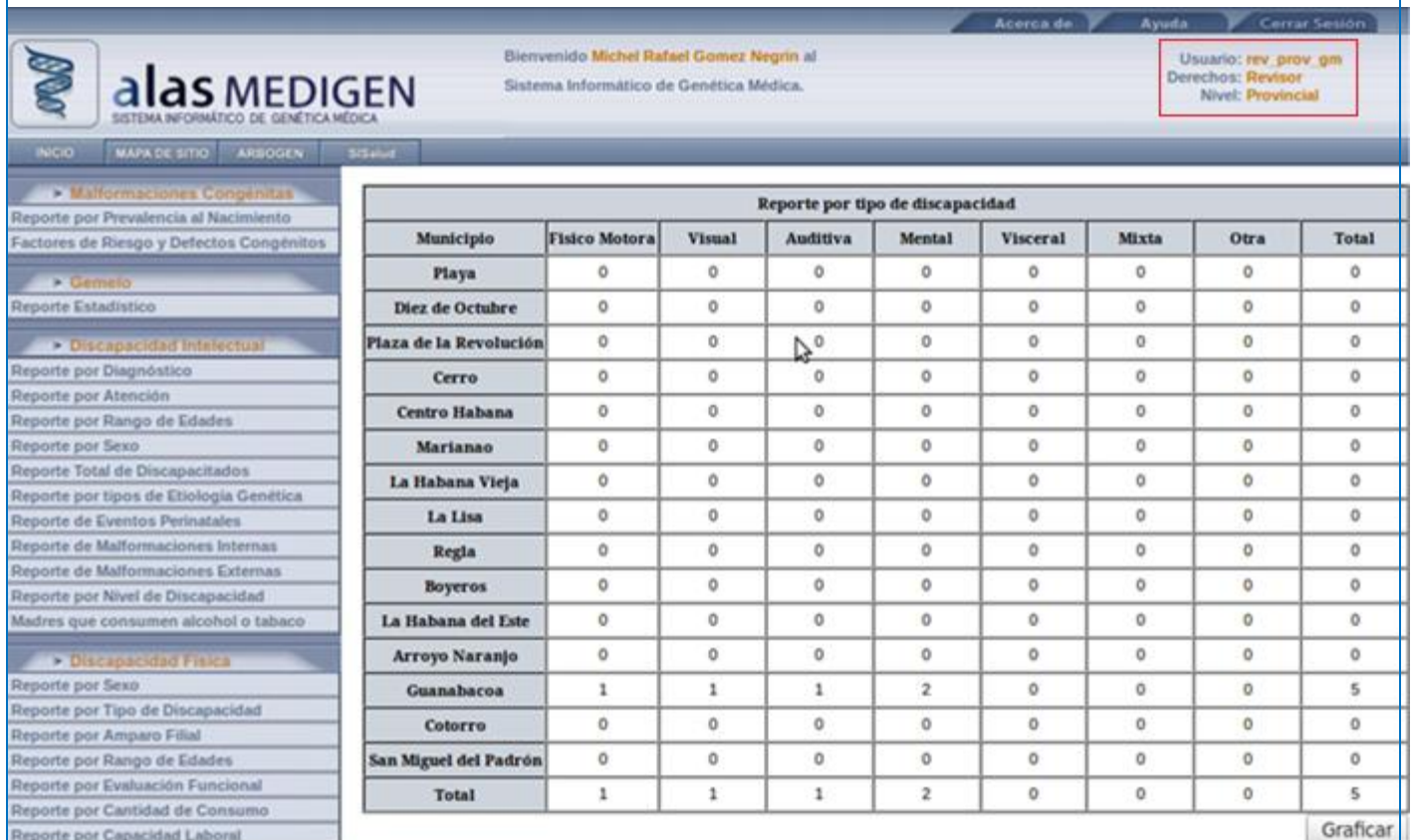


Figura # 4: Interfaz del reporte de Discapacidad Intelectual a nivel provincial.

Tabla # 4: Descripción del caso de uso del sistema. Generar reporte de discapacidad intelectual.

Caso de Uso:	Graficar Reporte
Actores:	Genetista (Inicia el caso de uso).
Resumen:	El caso de uso inicia cuando el genetista accede a graficar según el reporte seleccionado. El sistema muestra el reporte graficado en forma de barra. Luego el genetista escoge la opción de exportar a PDF la gráfica y selecciona el directorio donde quiere guardarlo. El sistema guarda el PDF en el directorio seleccionado, finalizando así el caso de uso.

Precondiciones:	Debe haberse ejecutado los casos de uso generar reportes de discapacidad y generar reportes de discapacidad intelectual	
Referencias	RF25, RF26	
Prioridad	Secundaria	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El genetista accede a la opción de graficar el reporte.	2. El sistema muestra el reporte graficado en forma de barra.	
3. El genetista escoge la opción de exportar a PDF la gráfica.	4. El sistema muestra una interfaz para que el genetista elija el directorio.	
5. El genetista selecciona el directorio en donde será guardado el PDF.	6. El sistema guarda el PDF en el directorio elegido, finalizando así el caso de uso.	

Tabla # 5: Descripción del caso de uso de graficar reporte.

Conclusiones

En este capítulo se abordaron los temas referentes a la elaboración del diagrama de clases del modelo de dominio, con una descripción detallada de las clases que intervienen para mejorar el entendimiento de la lógica de los procesos. Se definió 26 requisitos funcionales y 11 requisitos no funcionales que permitieron identificar las funcionalidades con las que contará el sistema, las cuales darán respuesta a las necesidades del problema. Además se identificó los actores, los casos de usos y la relación existente entre ellos reflejada en el diagrama de casos de uso del sistema, también se describieron detalladamente los casos de uso del sistema logrando así un mayor acercamiento a lo que el sistema deberá cumplir y un mayor entendimiento de los requisitos funcionales previamente establecidos.

Capítulo 3: Diseño del Sistema

Introducción

En el siguiente capítulo se muestra lo referente a la fase de diseño de los módulos RECUDI y RECUDIS. Se evidenciará el uso de los patrones aplicados por el framework de desarrollo Symfony así como lo referente a la arquitectura Modelo-Vista-Controlador (MVC) implementada. Además se describen los diagramas de diseño y los diagramas de secuencia, así como el modelo de despliegue.

3.1 Vista lógica de la arquitectura

La vista lógica representa los elementos de diseño más importantes para la arquitectura del sistema. Este permite observar cómo está diseñada la funcionalidad en el interior del sistema y lo describe mostrando las clases más importantes, su organización en paquetes y subsistemas.

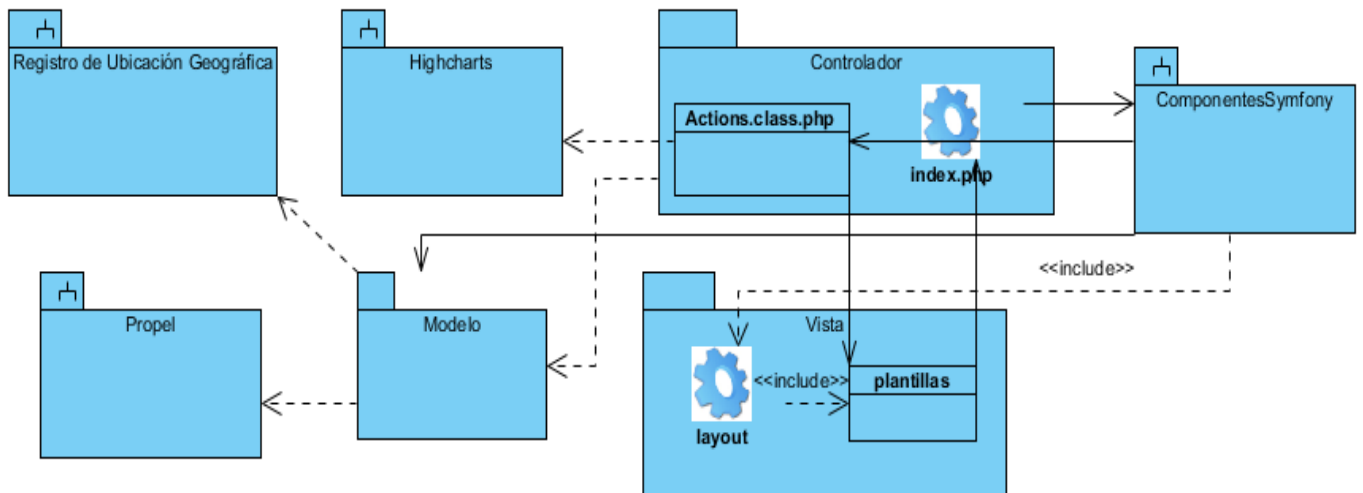


Figura # 5: Vista lógica del sistema

A continuación se expone la descripción de los paquetes y subsistemas del diseño que conforman la vista lógica.

Paquete del modelo: Representa la información con la que trabaja la aplicación, el cual 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 utilizan 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 Base de Datos, solamente es necesario actualizar la capa de abstracción de la base de datos.

Paquete vista: Obtiene los datos del modelo para generar una página web que permite al usuario interactuar con ella. Además está separada en un *layout* y en una *plantilla*. Normalmente, el *layout* contiene los elementos que se muestran de forma idéntica para las páginas web a lo largo de toda la aplicación y la *plantilla* sólo se encarga de visualizar las variables definidas en el paquete del controlador.

Paquete del controlador: Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y en la vista. Se encarga además de procesar las peticiones, las sesiones de los usuarios, la autenticación, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Además está dividida en varios componentes: en acciones y en un controlador frontal. Este último es el único punto de entrada a la aplicación, carga la configuración y determina la acción a ejecutarse. Las acciones son las que contienen la lógica de la aplicación, verifica la integridad de las peticiones y preparan los datos requeridos por la capa de presentación, utilizando los componentes de Symfony.

Subsistema Propel: Es un ORM cuya función es gestionar el modelo del sistema. Implica que el acceso y la modificación de los datos almacenados en la base de datos se realicen mediante objetos, nunca de forma explícita, permitiendo un alto nivel de abstracción y fácil portabilidad.

Subsistema Registro de Ubicación Geográfica: Permite listar los municipios y las provincias del país, obtener la provincia dado un municipio, buscar las localidades de un municipio o de una provincia, buscar las calles dado una localidad, una manzana, un municipio o una provincia y listar los países.

Subsistema Highcharts: Incluye una gran colección de gráficos para aplicaciones web como: gráficos de barra, de pastel, entre otros.

Subsistema Symfony: Representa todas las clases del framework Symfony que serán utilizadas durante el funcionamiento del sistema, tales como validadores de formularios, sistema de enrutamiento, componentes de seguridad y configuración, entre otros.

3.2 Diagrama de clases del diseño

Un diagrama de clases del diseño es una representación concreta de lo que se debe implementar. Estos diagramas representan la parte estática del sistema a través de la representación de las clases y sus relaciones. A continuación en la siguiente figura se muestra el diagrama clases del diseño del CU Generar reportes de discapacidad por tipo.

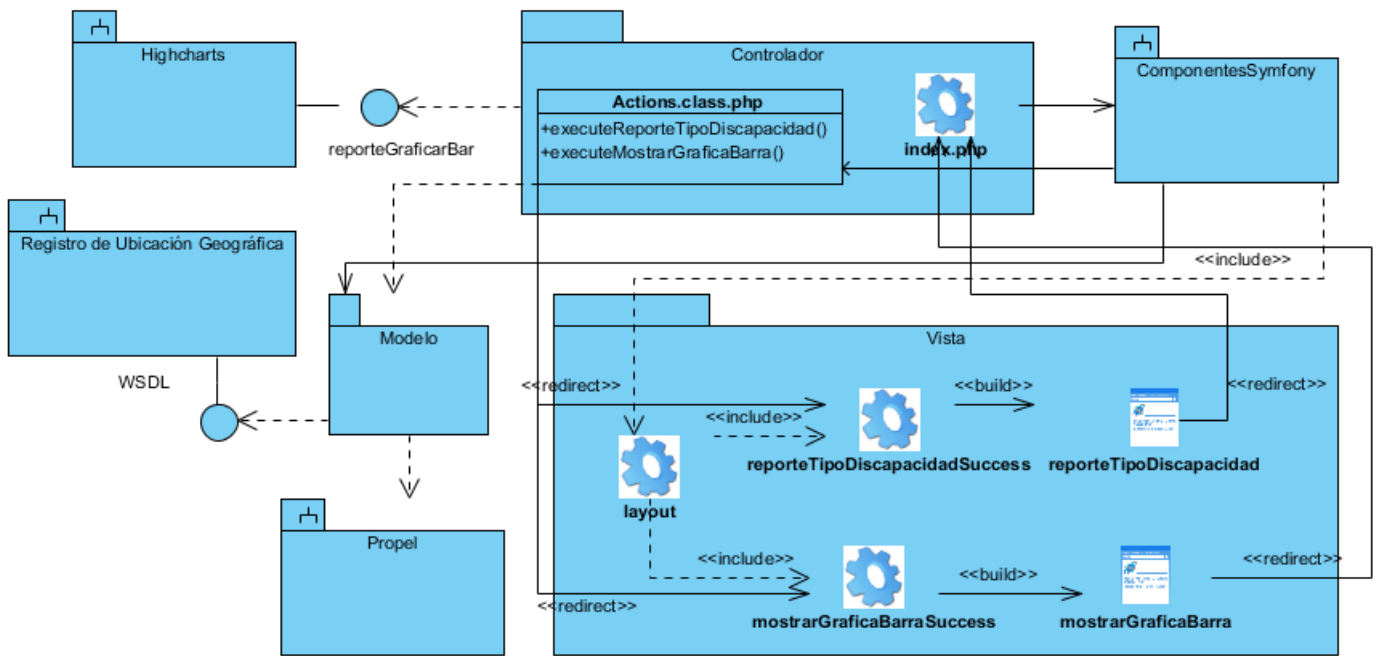


Figura # 6: Diagrama de clases del diseño. CU Generar reportes de discapacidad por tipo.

En este diagrama clases del diseño cuando el usuario selecciona la opción Reporte por Tipo de Discapacidad del Módulo Registro Cubano de Discapacidad se llama al controlador frontal *index.php*. Esta clase redirecciona la petición a la acción *executeReporteTipoDiscapacidad()* de la clase controladora *actions.class.php* a través de los componentes de Symfony. En esta acción se accede a la capa del modelo para buscar los datos del reporte, donde las clases del modelo acceden a los datos de la base de datos utilizando el ORM *Propel* que viene integrado en Symfony. Esta capa del modelo accede a los

Registros de Ubicación a través de la interfaz WSDL para poder obtener el listado de las provincias y de los municipios. Luego la clase *actions* se comunica con *reporteTipoDiscapacidadSuccess.php* de la clase vista para construir la página web *reporteTipoDiscapacidad* y esta última envía el resultado para el *index.php*.

Al seleccionar la opción de graficar el reporte en forma de barra el controlador frontal se encarga de redireccionar la petición a la acción *executeMostrarGraficaBarra()* de la clase controladora *actions.class.php*. La clase controladora se comunica con la interfaz *reporteGraficarBarr* de la biblioteca Highcharts, que permite graficar el reporte en forma de barra. Luego el resultado es enviado a la clase *mostrarGraficaBarraSuccess.php* que es el encargado de construir la vista *mostrarGraficaBarra* y enviar la información a la clase *Index.php* para mostrar el reporte graficado.

Además los restantes diagramas de diseño de los demás casos de usos siguen el mismo flujo y para consultarlos junto con sus paquetes de modelo pueden remitirse al expediente de proyecto.

Modelo

En las aplicaciones que son desarrolladas utilizando Symfony, el acceso y la modificación de los datos almacenados en la base de datos se realizan mediante objetos, para no acceder de forma explícita a la misma. Se utiliza Propel como interfaz para el mapeo objeto-relacional (ORM por sus siglas en inglés) que posibilita acceder a los datos mediante objetos. Esto permite la reutilización de los métodos de los objetos en cualquier parte de la aplicación y encapsula la lógica de los datos. Para traducir el modelo relacional de la base de datos a un modelo de objetos de datos, el ORM necesita la descripción del modelo relacional que se guarda en el archivo *schema.yml*. Es necesario ejecutar el comando *propel- build-model*, que analiza el esquema, genera las clases bases del modelo y las guarda en el directorio *lib/model/om* de la aplicación. Además genera las clases que se denominan propias del modelo en el directorio *lib/model/*. Estas clases propias heredan de las clases bases. Por cada tabla del modelo relacional se generan dos clases bases y dos clases propias del modelo. Cada vez que se modifique la base de datos se modifica el archivo *schema.yml*, teniendo que regenerarse las clases bases del modelo, debido a que son las que se generan directamente a partir del esquema. Por esto es que las funcionalidades se deben agregar en las clases propias y no en las clases bases.

A continuación se representan las clases del modelo utilizadas. Por cada tabla de la Base de datos se generan cuatro clases. De las doce clases generadas seis son propias del modelo y las otras seis son las clases bases. Las clases propias tienen las funcionalidades de acceso a los datos con las cuales interactúa la clase actions.

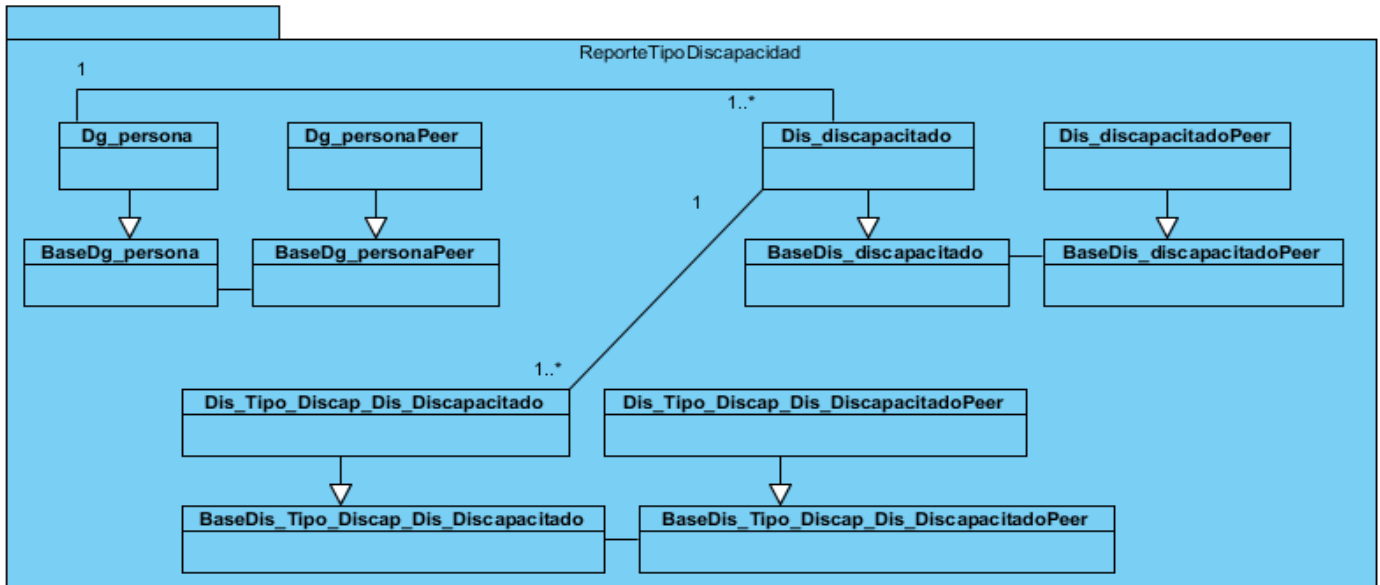


Figura # 7: Diagrama de clases del diseño. Paquete modelo del CU reporte por tipo de discapacidad.

3.3 Patrones

En el transcurso de los años han surgido varias definiciones de patrón y la más utilizada es la creada por Frank Buschmann que describe a un patrón como:

"Un problema de diseño recurrente, que surge en contextos específicos de diseño y presenta un esquema genérico probado para la solución de este. El esquema de la solución describe un conjunto de componentes, responsabilidades, relaciones entre éstos y formas en que dichos componentes colaboran entre sí." (16)

Es decir, un patrón es una solución experta a un problema determinado, de tal forma que la solución pueda ser reutilizada en proyectos distintos.

Luego de conocer sobre la definición de patrón, se hace necesario su análisis para el desarrollo de la solución. Para ello se encuentran agrupados en dos categorías: Patrones de diseño y Patrones arquitectónico, como se describe a continuación.

Patrones de Diseño

Una de las personas que ha influenciado el actual pensar de la Ingeniería de Software, o del Diseño en general, es el arquitecto Christopher Alexander que ha propuesto teorías originales aplicables al Diseño y a la Arquitectura. Estas teorías han sido utilizadas en el diseño de lenguajes de programación, en la identificación, uso de Patrones de Diseño y han sentado las bases para que surgiera el libro "Design Patterns, Elements of Reusable Object-Oriented Software" (Diseño de patrones, elementos de reutilización de aplicaciones orientadas a objetos) escrito por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, donde ofrecen la siguiente definición sobre los patrones de diseño:

“Un Patrón de diseño nombra, abstrae e identifica los aspectos claves de una estructura de diseño común, lo que los hace útiles para crear un diseño orientado a objetos reutilizable. El patrón de diseño identifica las clases e instancias participantes, sus roles, colaboraciones y la distribución de responsabilidades. Cada patrón de diseño se centra en un problema concreto, describiendo cuando aplicarlo y si tiene sentido hacerlo teniendo otras restricciones de diseño, así como las consecuencias y las ventajas e inconvenientes de su uso.” (17)

Con otras palabras los patrones de diseño son un conjunto de soluciones a problemas que generalmente encuentras en el diseño de un programa, que pueden ser reusadas en la construcción de aplicaciones. Además guían el camino para resolver problemas comunes en la programación y permiten que el diseño sea reutilizable y flexible.

Ventajas:

- Proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo.
- Los patrones favorecen la reutilización de clases ya existentes y la programación de clases reutilizables.
- La propia estructura del patrón es reutilizada cada vez que se aplica
- Facilitan el aprendizaje al diseñador inexperto.

- La reutilización de patrones de diseño ayuda a prevenir fallos inesperados y aumenta la legibilidad del código para implementadores y arquitectos familiares con los patrones. (18)

Existen varios patrones de diseño utilizados en el proceso de desarrollo del software. Estos son agrupados fundamentalmente por dos grandes grupos conocidos como Patrones de Software para la asignación General de Responsabilidades (General Responsibility Assignment Software Patterns - GRASP) y “La Banda de los cuatro” (Gang of Four - GOF).

Patrones de asignación de responsabilidades

Los Patrones Generales de Software para Asignar Responsabilidades (GRASP) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. A continuación se explican los patrones utilizados:

Experto: Es el encargado de asignar responsabilidad a una clase que disponen de la información necesaria para llevar a cabo una tarea. Si esta asignación de responsabilidades se hace adecuadamente, los sistemas tienden a ser más fáciles de entender, mantener y ampliar.

En la arquitectura de Symfony, específicamente en el modelo, existen dos tipos de clases fundamentales:

- Las de acceso a datos: Responsables de interactuar con las clases de abstracción de datos y devuelven los objetos que necesitan los controladores en su forma original.
- Las clases encargadas de la abstracción de datos: Responsables de realizar todas las operaciones con la Base de Datos.

Symfony genera 4 clases por cada tabla de la Base de Datos y las que trabajan directamente con la Base de Datos, son las terminadas en Peer (las clases de abstracción de datos), las encargadas de hacer las consultas a la Base de Datos utilizando Propel. Como las clases de abstracción de datos son las que tienen los atributos necesarios para realizar dicha función, por tanto tienen la responsabilidad de realizar las acciones directamente con la Base de Datos y aquí es donde se aplica el patrón Experto.

Creador: Este patrón es el encargado de crear las instancias de los objetos para acceder a las entidades que brindan la información necesaria para realizar cualquier acción

En la clase actions es la que contiene las acciones definidas para el sistema y es la encargada de ejecutarlas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase actions es “creador” de dichas entidades. Un ejemplo de su uso es en la función *executeReporteTipoDiscapacidad()* donde se crea una instancia de la clase Criteria convirtiendo a actions en creadora de dicha entidad.

Alta Cohesión: Es la clase que tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. Además una clase con alta cohesión son útiles porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. También mejoran la claridad y la facilidad con que se entiende el diseño.

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 de ello es la clase actions, la cual es la encargada de definir las diferentes acciones a realizar, asignando a cada una de ellas un objetivo específico con el fin que las tareas realizadas sean con mayor eficiencia.

Bajo Acoplamiento: Se encarga de asignar una responsabilidad de manera que el acoplamiento permanezca bajo, es decir, para mantener un acoplamiento bajo las clases tienen que estar lo menos ligadas entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases.

Este patrón permite que no existan dependencias innecesarias entre algunas partes del sistema que no tengan que compartir o utilizar información que no necesiten. En la implementación MVC de Symfony, esto se evidencia en el modelo donde se manifiesta el uso de este patrón, ya que las clases de abstracción a la base de datos solo 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 (la vista o el controlador), lo que proporciona que la dependencia en este caso sea baja dándole un alto grado de reutilización.

Controlador: Es el encargado de asignar la responsabilidad del manejo de uno de los eventos del sistema, a una clase facultada de atender determinada funcionalidad. Donde un evento del sistema es un

evento generado por un actor externo que se asocia a operaciones del sistema; las que emite en respuesta a los eventos del sistema.

Todas las peticiones Web son manipuladas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en la clase *actions* y en *index.php*.

Patrones de diseño GOF

Los patrones de diseño GOF se ha popularizado por el libro "DesignPatterns", que fue escrito por los que comúnmente se conoce como GoF (Gang of Four, "Pandilla de los Cuatro"), formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Ellos recopilaron y documentaron 23 patrones de diseño que fueron agrupados en tres categorías: de creación, de estructura y de comportamiento. A continuación se listan estos patrones, con una breve descripción del propósito de cada uno de ellos:

Patrón Decorator (Envoltorio): Añade funcionalidad a una clase, dinámicamente. Por ejemplo, en alasMEDIGEN hay un archivo nombrado *layout.php*. Este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, 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 procedimiento es una implementación del patrón Decorator.

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.

Front Controller (Controlador Frontal): 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 actions 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 actions.

3.4 Diagrama de Interacción

Se entiende como la representación de secuencias que componen un sistema, para facilitar el entendimiento de información y la relación que mantiene entre las diferentes partes. Modelan el comportamiento dinámico del sistema; el flujo de control en una operación y describe la interacción entre objeto a través de mensajes para cumplir ciertas tarea. Existen dos tipos de diagramas de interacción: Diagramas de Secuencia y Diagramas de Colaboración.

Diagrama de secuencia

Un diagrama de secuencia muestra la interacción entre usuarios, sistemas, subsistemas y hace hincapié en la ordenación del tiempo de los mensajes. Contiene detalles de implementación del escenario, incluyendo los objetos, clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos, es decir representan el flujo de las acciones que debe realizar el sistema. A continuación en la figura 6 se muestra el diagrama de secuencia correspondiente al caso de uso Generar Reportes por Discapacidad. Reporte por Tipo.

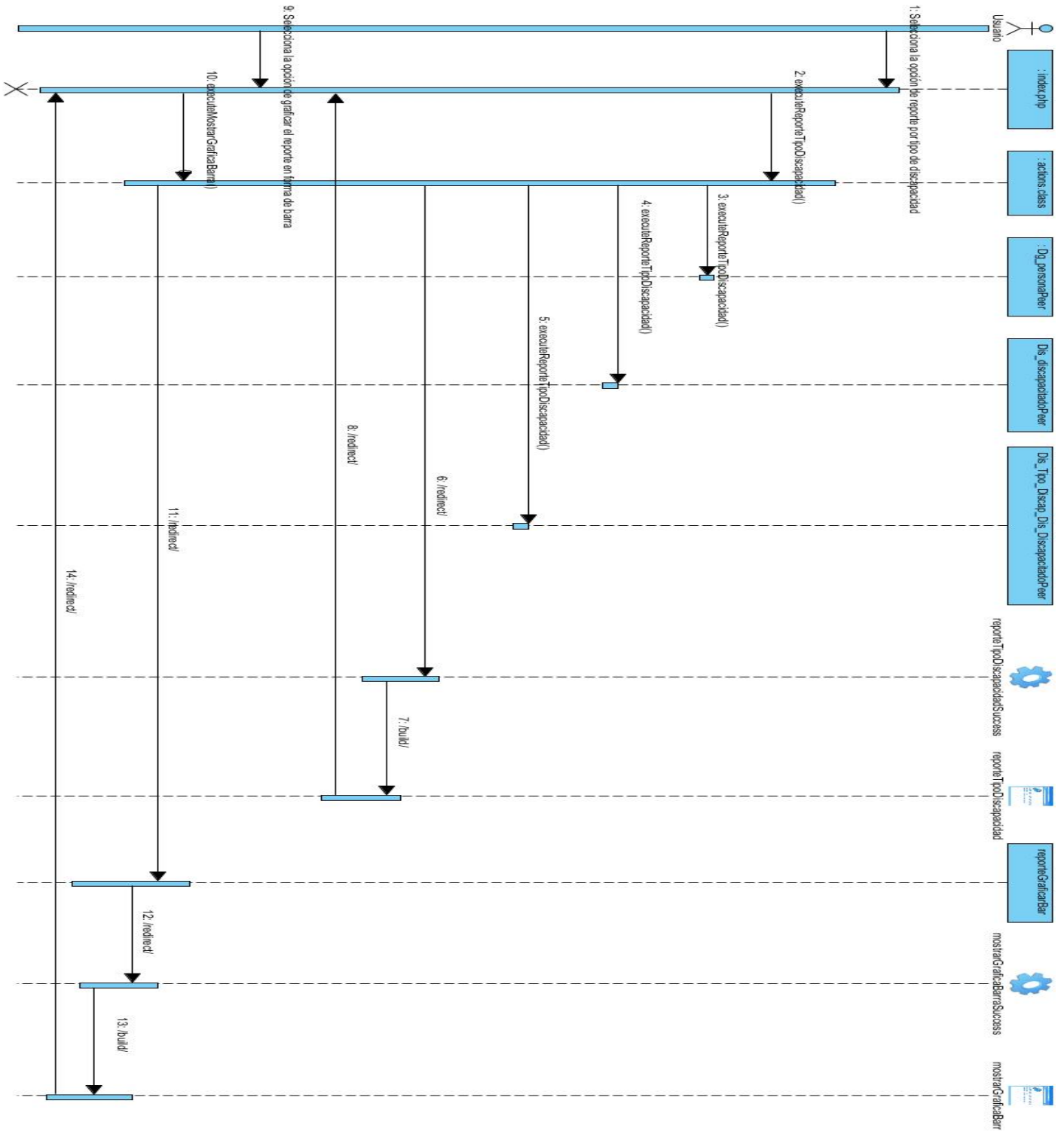


Figura # 8: Diagrama de secuencias. Generar reporte por tipo de discapacidad.

En este diagrama de secuencia cuando el usuario (genetista) selecciona la opción *Reporte por Tipo de Discapacidad* se hace un enlace al controlador frontal *index.php*. Este redirecciona la petición a la acción *executeReporteTipoDiscapacidad()* de la clase controladora *actions*. En esta acción se accede a la capa del modelo para buscar los datos del reporte y al Registro de Ubicación. Luego la clase *actions* se comunica con *reporteTipoDiscapacidadSuccess.php* de la clase vista para construir la página web *reporteTipoDiscapacidad* y esta última envía el resultado para el *index.php*.

Al seleccionar la opción de graficarlo en forma de barra, se hace un enlace al *index.php* que se encarga de redireccionar la petición a la acción *executeMostrarGraficaBarr()* de la clase controladora *actions.class.php*, donde se comunica con la biblioteca Highcharts, que permite graficar el reporte en forma de barra. Luego el resultado es enviado a la clase *mostrarGraficaBarrSuccess.php* que es el encargado de construir la vista *mostrarGraficatBarr*, donde esta vista envía la información a la clase *index.php* para mostrar el reporte graficado.

3.5 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema. Se utilizan para capturar los elementos de configuración del procesamiento, las conexiones entre esos elementos y para visualizar la distribución de los componentes en los nodos físicos. Además cada nodo representa un recurso computacional, generalmente con memoria y con capacidad de procesamiento o puede ser un dispositivo sobre el que se pueden desplegar los componentes; estos tienen conexiones bidireccionales que pueden ser HTTP, USB, TCP/IP, entre otros.

La siguiente figura muestra el diagrama de despliegue correspondiente a los módulos Registro Cubano de Discapacitados y Registro Cubano con Discapacidad Intelectual del sistema alasMEDIGEN.

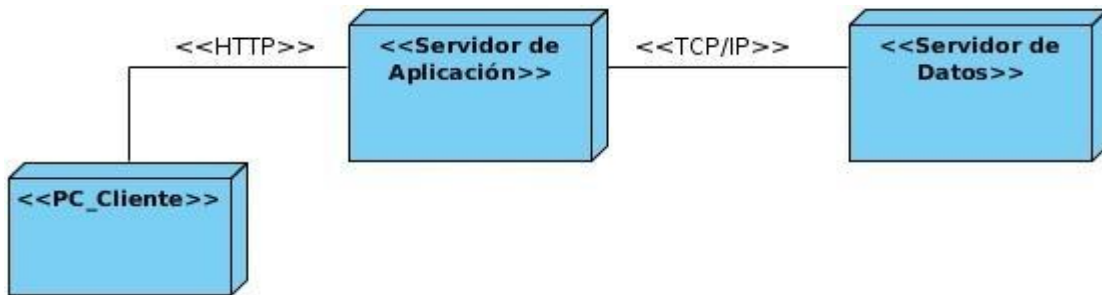


Figura # 9: Diagrama de la vista de despliegue.

El **nodo PC-Cliente** es por el cual se accede al sistema por medio de un navegador y se comunica con el servidor de aplicaciones donde se encuentra el sistema alasMEDIGEN a través del protocolo HTTP.

El **nodo Servidor de Datos** es donde se encuentra almacenada la base de datos, que mediante el protocolo TCP/IP se conectará al servidor de aplicaciones.

El **nodo Servidor de Aplicación** se procesa todas las órdenes que vengan del nodo PC-Cliente a través del protocolo HTTP y se comunica con el servidor de datos mediante el protocolo TCP/IP.

Conclusiones

En este capítulo al desarrollar la vista lógica se tuvieron en cuenta las clases, paquetes y subsistemas de diseño más significativos para el sistema, permitiendo observar la forma en están diseñadas las funcionalidades en el interior del mismo. Posteriormente se realizó la realización del diagrama de clases del diseño, del cual se obtuvo una mejor comprensión estructural y funcional del sistema para su posterior implementación, apoyado en los patrones de diseño GRASP y GOF. Además se generaron los diagramas de secuencia, así como los diagrama de despliegue donde se representaron los nodos y conexiones que intervienen en el funcionamiento del sistema, mostrando la configuración física sobre la que será desplegado.

Capítulo 4: Implementación y Prueba

Introducción

En el siguiente capítulo se realiza una evaluación del funcionamiento del framework Symfony donde se muestra su estructura. Se describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Además se realizaron varias pruebas para conocer la calidad del producto y para detectar y corregir los errores que pueda presentar la aplicación, utilizando los métodos de prueba.

4.1 Estructura del sistema

Symfony establece una estructura de carpetas para sus proyectos. Dentro de un proyecto, las operaciones se agrupan de forma lógica en aplicaciones. Cada aplicación está formada por uno o más módulos. A continuación se describen las carpetas más importantes y seguido se muestra la figura 8 de la estructura física de alasMEDIGEN.

Los directorios más importantes son:

- **apps:** Contiene un directorio por cada aplicación del proyecto, en este caso genética.

Subdirectorios de apps:

- ✓ **config:** Contiene archivos de configuración adicionales con parámetros exclusivos.
- ✓ **lib:** Almacena las clases y librerías utilizadas exclusivamente por los módulos.
- ✓ **modules:** Almacena los módulos que definen las características de la aplicación
- ✓ **templates:** Contiene las plantillas correspondientes a las acciones del módulo.
- **config:** Almacena la configuración general del proyecto.
- **data:** En este directorio se almacenan los archivos relacionados con los datos, como por ejemplo el esquema de una base de datos
- **lib:** Guarda todo el código común a todas las aplicaciones del proyecto.
- **log:** Guarda todos los archivos de log generados por Symfony. También se utiliza para guardar los logs del servidor web, de la base de datos o de cualquier otro componente del proyecto.
- **web:** Constituye la raíz web del sistema. Contiene los únicos archivos accesibles desde Internet.

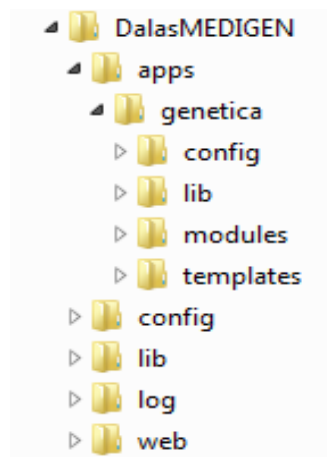


Figura # 10: Estructura del sistema.

4.2 Diagrama de componente

Es el encargado de modelar la vista estática de un sistema y describir los elementos físicos compuesto por archivos, librerías, módulos, ejecutables o paquetes. Pueden ser usados para modelar y documentar cualquier arquitectura del sistema. Los diagramas de componentes son empleados para estructurar el modelo de implementación en términos de subsistemas y mostrar las relaciones entre sus elementos.

En la figura 9 se muestra la vista del diagrama de componentes que corresponde al caso de uso Generar Reporte de Tipo de Discapacidad. Además es importante destacar que para el resto de los casos de uso se mantiene la misma estructura, solo cambia el nombre de los componentes correspondientes a la vista, que vienen dado según el caso de uso.

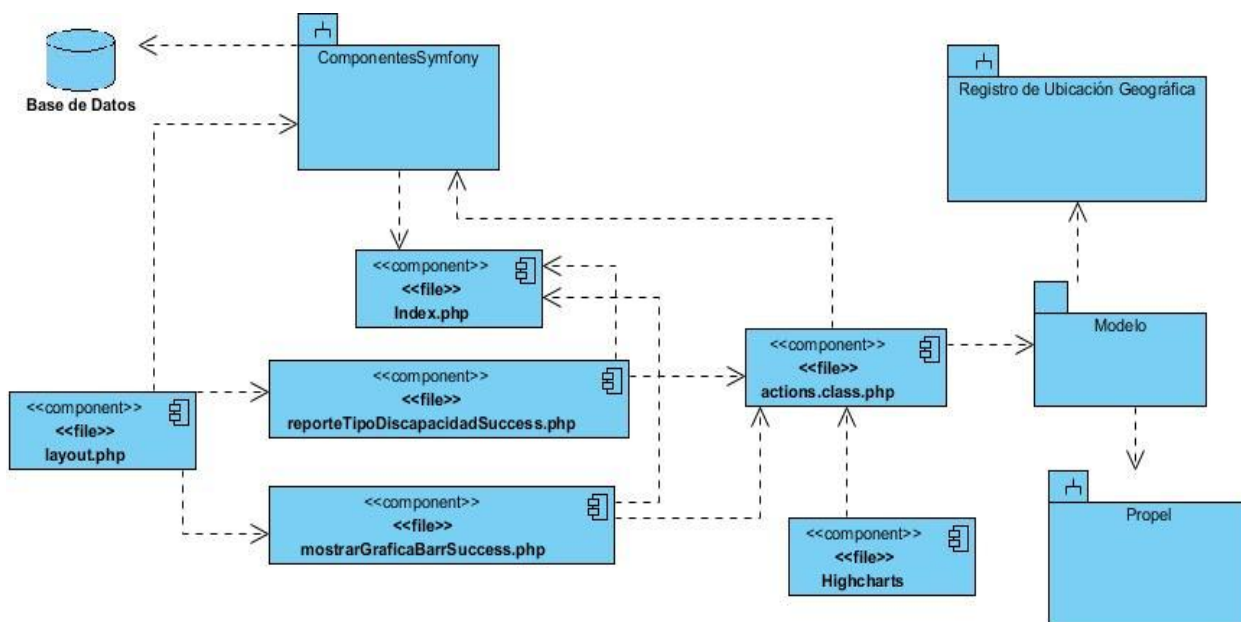


Figura # 11: Diagrama de componentes. Generar reporte de tipo de discapacidad.

A continuación se expone la descripción de los componentes y subsistemas de implementación que conforman el diagrama de componente.

index.php: Corresponde con el controlador frontal del sistema que es el único punto de entrada a la aplicación.

layout.php: Contiene los elementos que se muestran de forma idéntica a lo largo de toda la aplicación.

Base de Datos: Contiene todos los datos del sistema.

actions.class.php: Corresponde a la clase que almacena todas las acciones del módulo.

reporteTipoDiscapacidadSuccess.php: Se encarga de visualizar el Reporte por Tipo de Discapacidad.

mostrarGraficaBarraSuccess.php: Es el encargado de mostrar el reporte seleccionado por el usuario en forma de barra.

Highcharts: El componente Highcharts es una biblioteca que incluye gráficos para aplicaciones web como: gráficos de barra, de pastel, entre otros.

Subsistemas de implementación Symfony: Encapsula todos los componentes propios del framework.

Subsistemas de implementación Propel: El componente Propel es el ORM que utiliza Symfony.

Subsistemas de implementación Registro de Ubicación Geográfica: El componente Registro de Ubicación Geográfica permite obtener todas las provincias y los municipios del país.

Subsistema de implementación Modelo: El componente Modelo es el encargado del acceso a los datos almacenados en la Base de Datos

4.3 Prueba de software

Una prueba de software constituye un pilar indispensable para evaluar y determinar la calidad de un producto, al comprobar los distintos modelos que se van generando durante todo el proceso de desarrollo. Básicamente es una fase en el desarrollo del software que consiste en probar las aplicaciones construidas, con la intención de descubrir errores y como resultado permitir un mayor control e identificación temprana de los fallos, garantizando la calidad del desarrollo con una reducción notable de los costos necesarios para corregir los errores. Para verificar los Módulos Registro Cubano de Discapacitados y Registro Cubano con Discapacidad Intelectual se realizó los métodos de prueba de caja blanca y caja negra que se encuentran explicados a continuación:

Métodos de caja blanca

Las pruebas de Caja Blanca, también conocidas como pruebas de caja de cristal o pruebas estructurales se basan en evaluar minucioso los detalles del código, siendo necesario conocer la lógica del programa. Son consideradas como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado la disminución del número de errores existentes, aportando mayor calidad y confiabilidad al producto.

Se escogieron dos niveles de prueba:

Prueba de desarrollador: Está diseñada e implementada por el equipo de desarrollo y tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad.

Prueba de unidad: Enfocada a los elementos más pequeño del software buscando la detección de errores

de cada módulo por individual, verificando que estos funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca.

La técnica utilizada en este método es la prueba del camino básico, para ello es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática. Esta consiste en derivar casos de prueba a partir de un conjunto de caminos independientes por los cuales puede circular el flujo de control. Para ello es necesario construir el grafo de flujo asociado y calcular su complejidad ciclomática. Una vez identificados los caminos independientes se procede a preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico

La complejidad ciclomática se calcula mediante tres formas:

$$V(G) = (A - N) + 2$$

$$V(G) = P + 1$$

$$V(G) = R$$

Donde $V(G)$ es la complejidad ciclomática de un grafo G , "A" la cantidad total de aristas, "N" la cantidad total de nodos, "P" la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas) y "R" la cantidad total de regiones.

A continuación se describe un ejemplo que muestra cómo se realiza este tipo de pruebas.

Se analizó la clase `actions`, específicamente el procedimiento `executeReporteConsumo()`, luego se identificaron y se enumeró los bloques de ejecución como se muestra a continuación.


```
//reporte por cantidad de consumo pagina 46
public function executeReporteConsumo() {
1  $this->privilegiosReview();

2  if (sfConfig::get('nivel') != 'Nacional') {
3      $columnaX = proxyUbicacion::getAllMunicipios($this->getUser()->getToken(), $this->getUser()->getIdProvincia());
4  } else {
5      $columnaX = proxyUbicacion::getAllProvincias($this->getUser()->getToken());
6  }

6  $columnaY = array(
    "Social" => "Social", "De Riesgo" => "De Riesgo", "Abuso" => "Abuso"
    , "Dependencia" => "Dependencia");
6  $resultado = array();

6  $c = new Criteria();
6  $c->addJoin(DgPersonaPeer::ID_PERSONA, DisDiscapitadoPeer::ID_PERSONA);
6  $c->addJoin(DisDiscapitadoPeer::ID_DISCAPITADO, DisConsumoAlcoholNfPeer::ID_DISCAPITADO);
7  foreach ($columnaX as $cp => $vp) {
8      foreach ($columnaY as $cd => $vd) {
9          $c->add(DisDiscapitadoPeer::CONSUMO, $cd);
10         if (sfConfig::get('nivel') != 'Nacional') {
11             $c->add(DgPersonaPeer::ID_PROVVIVE, $this->getUser()->getIdProvincia());
11             $c->add(DgPersonaPeer::ID_MUNVIVE, $vp->getIdMunicipio());
12         } else {
13             $c->add(DgPersonaPeer::ID_PROVVIVE, $vp->getIdProvincia());
14         }
14         $resultado[$cp][$cd] = DgPersonaPeer::doCount($c);
15     }
16 }

17 $this->columnaX = $columnaX;
17 $this->columnaY = $columnaY;
17 $this->resultado = $resultado;
18 if ($this->getRequest()->getMethod() == sfRequest::POST) {
19     $this->grafica = 'Bar';
20 }
```

Figura # 12: Representación del código del método *executeReporteConsumo()*.

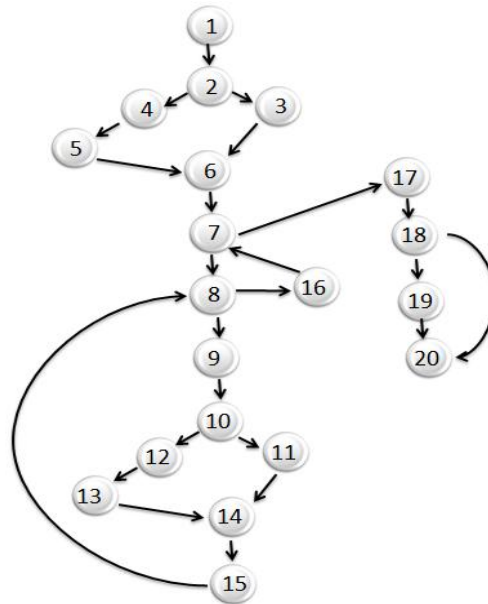


Figura # 13: Grafo del flujo asociado al método *executeReporteConsumo()*.

Se logró obtener 20 nodos, se dibujó el grafo de flujo asociado y se determinó el camino básico. Como se muestra en el grafo de flujo, las aristas indican los posibles caminos a seguir a partir del nodo correspondiente. Partiendo del camino básico determinado, fue aplicado uno de los tres métodos para calcular la complejidad ciclomática, específicamente el método $V(G) = A - N + 2$, aunque todos ellos arrojan al mismo resultado. Se obtuvieron 24 aristas y 20 nodos, quedando la fórmula de la siguiente forma: $V(G) = 24 - 20 + 2$. Por lo tanto la complejidad ciclomática tiene un valor de 6 lo cual significa que existen 6 posibles casos de ejecución para la función.

Número	Camino básico
1	1-2-3-6-7-8-9-10-11-14-15-8-16-7-17-18-19-20
2	1-2-4-5-6-7-8-9-10-12-13-14-15-8-16-7-17-18-20
3	1-2-3-6-7-8-16-7-17-18-19-20
4	1-2-4-5-6-7-8-16-7-17-18-20
5	1-2-3-6-7-17-18-19-20
6	1-2-4-5-6-17-18-20

Tabla # 6: Caminos básicos del flujo de la prueba de caja blanca.

Caso de prueba

Un caso de prueba específica una forma de probar el sistema, incluyendo las entradas con las que se prueba, los resultados esperados y las condiciones bajo las que ha de probarse. Ayuda a verificar que la aplicación desarrollada realice las funciones para las que ha sido creado en base a los requerimientos del usuario.

Caso de prueba para el primer camino básico:

La confección del caso de prueba sobre el método *executeReporteConsumo()*, se realizó con la utilización del grafo modelado anteriormente utilizando la técnica de camino básico . En este grafo se representaron 6 caminos de los que se obtiene los diferentes resultados a obtener en su ejecución. Para el caso de la ejecución del primer camino se pudo identificar que como resultado se obtenía datos que no correspondía con los esperados, debido que la variable tenía que contar las personas discapacitadas que consumen alcohol y no los familiares que consumen alcohol, como era el caso que sucedía en esta situación. Detectándose para el caso del nodo 11, la condición que definía el resultado era incorrecto por lo que se decidió a raíz del problema identificado proseguir a su solución. Para ello se esclarecieron las condiciones necesarias para que el camino realizado se ejecutara correctamente. Para los restantes caminos, el método se efectuó de forma satisfactoria al no detectarse ningún error en el código.

Método de Caja Negra

El método de caja negra, también conocida como pruebas de comportamiento, se centra en los requisitos funcionales. Tiene como objetivo demostrar que las funciones del software son operativas, que las entradas se acepten de forma adecuada y se produzca un resultado correcto, teniendo en cuenta que la integridad de la información externa se mantenga. Las pruebas se llevan a cabo sobre la interfaz del software, son completamente indiferentes al comportamiento interno y a la estructura del programa.

Este método de prueba se centra en lo que se espera del software, es decir, los casos de prueba pretenden demostrar que las funciones del sistema son operativas, que los valores de entradas se aceptan adecuadamente y que se produce una salida correcta, sin preocuparse de lo que pueda estar haciendo el software internamente. La prueba de caja negra no es una alternativa a las técnicas de prueba de la caja blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los

encontrados en los métodos de la caja blanca. Como nivel de prueba del método de caja negra que se escogió fue el desarrollador.

Los casos de prueba de caja negra constituyen un conjunto de condiciones de ejecución desarrollado para cumplir un objetivo en particular o una función esperada y demostrar el estado de operatividad en que se encuentra el software. Para la confección de los casos de prueba de caja negra se utilizó la técnica de Partición de Equivalencia, la cual representa una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que de otro modo requerirían la ejecución de muchos casos antes de detectar el error genérico.

A continuación se muestra el diseño de caso de prueba para el caso de uso Generar Reportes de Discapacidad, que incluye el siguiente escenario “Generar Reporte de Discapacidad por Tipo”.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC1 Generar Reporte por Tipo de Discapacidad	Cuando el genetista solicita algún reporte de discapacidad. El sistema muestra el reporte deseado de acuerdo al criterio escogido.	El sistema muestra el reporte de acuerdo al criterio escogido por el usuario.	<ol style="list-style-type: none">1. El Genetista selecciona la opción: Generar Reporte por Tipo de Discapacidad.2. El sistema muestra el reporte correspondiente según el nivel del usuario.

Tabla # 7: Caso de prueba. Generar reporte discapacidad.

Este proceso fue realizado para cada uno de los casos de uso del sistema obteniéndose tres casos de pruebas. Los resultados de las pruebas que no fueron satisfactorios pasaron a ser no conformidades. En la siguiente tabla se muestran algunas de las no conformidades encontradas.

Elemento	Número	No conformidad	Aspecto correspondiente	Etapa de detección	Clasificación	Estado NC	Respuesta de Equipo Desarrollo
Aplicación	1	En el Módulo Registro Cubano de Discapitados, cuando se muestra el Reporte por Tipo de Discapacidad se debe visualizar el campo "Otra".	Generar Reporte por Tipo de Discapacidad	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 2 de mayo de 2013 RA: 7 de mayo de 2013	Se realizaron las validaciones pertinentes.
Aplicación	2	En el Módulo Registro Cubano de Discapitados, cuando se muestra el Reporte por Cantidad de Consumo se debe mostrar solo la cantidad de discapacitados que consumen alcohol.	Generar Reporte por Cantidad de Consumo	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 2 de mayo de 2013 RA: 7 de mayo de 2013	Se realizaron las validaciones pertinentes.
Aplicación	3	En el Módulo Registro Cubano de Discapitados, el Reporte por Nivel de Discapacidad debe aparecer en el Módulo Registro	Generar Reporte por Nivel de Discapacidad	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 2 de mayo de 2013 RA: 7 de mayo de 2013	Se realizaron las validaciones pertinentes.

		Cubano de Discapacidad Intelectual.					
Aplicación	4	En el Módulo Registro Cubano de Discapacitados, el Reporte por Madres que Consumen Alcohol y Tabaco debe aparecer en el Módulo Registro Cubano de Discapacidad Intelectual.	Generar Reporte por Madres que Consumen Alcohol y Tabaco	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 2 de mayo de 2013 RA: 7 de mayo de 2013	Se realizaron las validaciones pertinentes.
<p><u>PD</u>: Prueba pendiente.</p> <p><u>RA</u>: Prueba resuelta.</p>							

Tabla # 8: No conformidades del sistema

4.4 Evidencia de la ejecución de las pruebas

En los módulos RECUDI Y RECUDIS se detectaron varias no conformidades, las cuales han estado afectando la efectividad del sistema, impidiendo que se cumplan los requisitos establecidos. Estas a pesar que representan errores en el sistema, te permiten identificar fallos que suceden en la interfaz de la aplicación. Estas son solucionadas por el desarrollador al final de cada iteración, mediante la prueba de regresión. En la siguiente figura se muestra la imagen referente a la tabla del reporte por cantidad de consumo, que inicialmente para la provincia de Pinar del Río no existen ninguna persona insertada en la base de datos para el criterio “Social”.

Reporte por cantidad de consumo					
Provincia	Social	De Riesgo	Abuso	Dependencia	Total
Pinar del Río	0	0	0	4	4
Artemisa	0	0	0	1	1
La Habana	0	0	0	1	1
Mayabeque	0	0	0	1	1
Matanzas	1	0	0	0	1
Cienfuegos	0	0	1	2	3
Villa Clara	0	0	0	1	1
Sancti Spiritus	0	0	0	0	0
Ciego de Avila	0	0	0	0	0
Camagüey	0	1	0	0	1
Las Tunas	0	0	0	2	2
Holguín	0	0	0	2	2
Granma	0	0	0	0	0
Santiago de Cuba	0	0	0	1	1
Guantánamo	0	0	0	1	1
Isla de la Juventud	0	0	0	2	2

Figura # 14: Reporte por cantidad de consumo.

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

Miembros del núcleo que consumen alcohol

Miembros del núcleo	Social	De Riesgo	Abuso	Dependencia
Madre	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Padre	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hermano	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Espos(a)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hijo(a)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Otros	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Discapacitado	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura # 15: Clasificación de consumo de bebidas alcohólicas.

En esta tabla se le asigna un criterio al discapacitado en dependencia de la cantidad de alcohol que consume en este caso el criterio es consumo de tipo "Social".

Reporte por cantidad de consumo					
Provincia	Social	De Riesgo	Abuso	Dependencia	Total
Pinar del Río	1	0	0	4	5
Artemisa	0	0	0	1	1
La Habana	0	0	0	1	1
Mayabeque	0	0	0	1	1
Matanzas	1	0	0	0	1
Cienfuegos	0	0	1	2	3
Villa Clara	0	0	0	1	1
Sancti Spiritus	0	0	0	0	0
Ciego de Avila	0	0	0	0	0
Camagüey	0	1	0	0	1
Las Tunas	0	0	0	2	2
Holguín	0	0	0	2	2
Granma	0	0	0	0	0
Santiago de Cuba	0	0	0	1	1
Guantánamo	0	0	0	1	1
Isla de la Juventud	0	0	0	2	2

Figura # 16: Reporte por cantidad de consumo.

Por último se muestra en la figura anterior el resultado de la cantidad de personas discapacitada por provincias que consumen alcohol según el criterio seleccionado anteriormente.

Conclusiones

En el presente capítulo se abordó sobre la estructura de Symfony, donde se identificaron a modules y templates como los subdirectorios básicos para la realización de los reportes en el Registro Cubano de Discapacitados y Registro Cubano de Discapacidad Intelectual. Se identificó la clase actions como la más importante dentro del diagrama de componentes por ser la encargada de almacenar todas las acciones de los módulos. Finalmente los reportes creados fueron evaluados mediante los métodos de prueba de caja blanca y caja negra, para garantizar el correcto funcionamiento de la aplicación. En la realización de caja negra se obtuvieron tres casos de pruebas, de los cuales se encontraron cuatro no conformidades que fueron resueltas satisfactoriamente. Mediante la ejecución del prueba de caja blanca fue aplicado al método `executeReporteConsumo()` obteniéndose seis caminos básicos, donde se corrigió el error detectado en el código.

Conclusiones

Con el desarrollo de los nuevos reportes en los módulos Registro Cubano de Discapacitados y Registro Cubano de Discapacidad Intelectual se ha arribado a las siguientes conclusiones:

- El estudio de la arquitectura del sistema alasMEDIGEN permitió determinar la selección de las herramientas y las tecnologías a utilizar para el desarrollo de los nuevos reportes en los módulos Registro Cubano de Discapacitados y Registro Cubano de Discapacidad Intelectual.
- El análisis de las funcionalidades del sistema y la utilización de los diferentes patrones de diseño y de arquitectura permitieron implementar una solución considerando las buenas prácticas de desarrollo del software, siendo comprobada mediante las pruebas que fueron aplicadas al sistema.
- La investigación arrojó la implementación de 24 nuevos reportes que se incorporaron al sistema alasMEDIGEN en los módulos Registro Cubano de Discapacitados y Registro Cubano de Discapacidad Intelectual, los cuales se pueden visualizar de forma tabular y gráfica, dándole al especialista facilidades para el análisis de un gran volumen de datos y la toma de decisiones en función del beneficio de la sociedad cubana.

Recomendaciones

A partir del estudio realizado se recomienda:

- Incorporar datos al instrumento de Discapacidad Intelectual en el sistema alasMEDIGEN que permitan generar nuevos reportes necesarios para facilitar la toma de decisiones a los genetistas.
- Exportar a formato de Microsoft Excel los reportes de RECUDI y RECUDIS.

Trabajos Citados

1. Glosario (Genética Médica). *Fundación Josep Carreras contra la leucemia*. [En línea] 17 de 1 de 2013. [Citado el: 17 de 1 de 2013.] www.fcarreras.org/es/.
2. **Marcheco, B.** “La Genética en la Salud Pública: el desafío del acceso de todos a los beneficios”. [Revista Electrónica] *Revista Cubana de Genética Comunitaria*, 2008.Vol 1, [Editorial] pp. 20-22., Disponible en: <http://bvs.sld.cu/revistas/rcgc/v1n1/gcoed107.pdf>
3. Reporte. *Definicion.De*. [En línea] [Citado el: 5 de 12 de 2012.] <http://definicion.de/reporte/>.
4. Reporte informático. *Sippec*. [En línea] [Citado el: 3 de 11 de 2012.] <http://sippec.sep.gob.mx/WebHelp/reportes/reporte.htm>.
5. Introducción a Crystal Reports. *msdn*. [En línea] [Citado el: 20 de 11 de 2012.] <http://msdn.microsoft.com/es-es/library/aa287933%28v=vs.71%29.aspx>.
6. Active Reports. *Active Reports*. [En línea] [Citado el: 16 de 11 de 2012.] <http://www.componentsource.com/products/activereports-6/index-es.html>.
7. Gráfico de barra. *msdn*. [En línea] [Citado el: 2012 de 12 de 9.] <http://msdn.microsoft.com/es-es/library/ms252077%28v=vs.80%29.aspx>.
8. Proceso Unificado de Desarrollo de Software (RUP). *eumed.net*. [En línea] [Citado el: 7 de 12 de 2012.] <http://www.eumed.net/libros-gratis/2009c/587/Proceso%20Unificado%20de%20Desarrollo%20de%20Software.htm>.
9. Herramienta CASE. *scribd*. [En línea] [Citado el: 12 de 12 de 12.] <http://es.scribd.com/doc/57334530/Herramienta-CASE>.
10. 01 programación Mónica Girón Giner. *pdf online*. [En línea] [Citado el: 9 de 1 de 2013.] <http://share.pdfonline.com/1185c3e644bf412ea7e65c1539cb3f34/01-Programaci%C3%B3n.htm>.
11. **Marley, Jimi.** Artículo(PHP). *Programacion*. [En línea] [Citado el: 10 de 12 de 2012.] http://www.programacion.com/articulo/por_que_elegir_php_143.
12. **Potencier, Fabien y Zaninotto, Francois.** *Symfony la guía definitiva*. 2008.
13. MySQL. *Scribd*. [En línea] [Citado el: 13 de 12 de 2012.] <http://www.scribd.com/doc/58274826/MySQL-Diapositivas-este-si>.
14. Netbeans. *Netbeans*. [En línea] [Citado el: 4 de 12 de 2012.] <http://netbeans.org/features/index.html>.
15. **Pablo González.** Creación de gráficas en PHP con JpGraph. *Desarrollo Web*. [En línea] [Citado el: 14 de 11 de 2012.] www.desarrolloweb.com/articulos/1987.php.

16. Patrones(Conceptos Generales). scribd. [En línea] [Citado el: 20 de 3 de 13.] <http://es.scribd.com/doc/137642292/75B63d01>.
17. **León Welicki**. Patrones y Antipatrones: una Introducción - Parte I . *msdn*. [En línea] [Citado el: 10 de 4 de 13.] <http://msdn.microsoft.com/es-es/library/bb972242.aspx>.
18. Manuales y guías prácticas sobre patrones de diseño. *scribd*. [En línea] [Citado el: 21 de 4 de 13.] <http://es.scribd.com/doc/110474646/patrones-de-diseno-en-uml>.

Bibliografía

1. **Fundación Josep Carreras.** Glosario (Genética Médica). *Fundación Josep Carreras contra la leucemia*. [En línea] 17 de 1 de 2013. [Citado el: 17 de 1 de 2013] www.fcarreras.org/es/.
2. **Marcheco, B.** “La Genética en la Salud Pública: el desafío del acceso de todos a los beneficios”. [Revista Electrónica] *Revista Cubana de Genética Comunitaria*, 2008.Vol 1, [Editorial] pp. 20-22., Disponible en: <http://bvs.sld.cu/revistas/rcgc/v1n1/gcoed107.pdf>.
3. Reporte. *Definicion.De*. [En línea] [Citado el: 5 de 12 de 2012.] <http://definicion.de/reporte/>.
4. Reporte informático. *Sipec*. [En línea] [Citado el: 3 de 11 de 2012] <http://sipec.sep.gob.mx/WebHelp/reportes/reporte.htm>.
5. Introducción a Crystal Reports. *msdn*. [En línea] [Citado el: 20 de 11 de 2012] <http://msdn.microsoft.com/es-es/library/aa287933%28v=vs.71%29.aspx>.
6. Active Reports. *Active Reports*. [En línea] [Citado el: 16 de 11 de 2012] <http://www.componentsource.com/products/activereports-6/index-es.html>.
7. Gráfico de barra. *msdn*. [En línea] [Citado el: 9 de 12 de 2012] <http://msdn.microsoft.com/es-es/library/ms252077%28v=vs.80%29.aspx>.
8. Proceso Unificado de Desarrollo de Software (RUP). *eumed.net*. [En línea] [Citado el: 7 de 12 de 2012] <http://www.eumed.net/libros-gratis/2009c/587/Proceso%20Unificado%20de%20Desarrollo%20de%20Software.htm>.
9. Herramienta CASE. *scribd*. [En línea] [Citado el: 12 de 12 de 12] <http://es.scribd.com/doc/57334530/Herramienta-CASE>.
10. 01 programación Mónica Girón Giner. *pdf online*. [En línea] [Citado el: 9 de 1 de 2013] <http://share.pdfonline.com/1185c3e644bf412ea7e65c1539cb3f34/01-Programaci%C3%B3n.htm>.
11. **Marley, Jimi.** Artículo(PHP). *Programacion*. [En línea] [Citado el: 10 de 12 de 2012] http://www.programacion.com/articulo/por_que_elegir_php_143.
12. **Potencier, Fabien y Zaninotto, Francois.** *Symfony la guía definitiva*. 2008.
13. MySQL. *Scribd*. [En línea] [Citado el: 13 de 12 de 2012] <http://www.scribd.com/doc/58274826/MySQL-Diapositivas-este-si>.
14. Netbeans. *Netbeans*. [En línea] [Citado el: 4 de 12 de 2012] <http://netbeans.org/features/index.html>.

15. Creación de gráficas en PHP con JpGraph. *Desarrollo Web*. [En línea] [Citado el: 14 de 11 de 2012.] www.desarrolloweb.com/articulos/1987.php.
16. Patrones(Conceptos Generales). scribd. [En línea] [Citado el: 20 de 3 de 13.] <http://es.scribd.com/doc/137642292/75B63d01>.
17. Patrones y Antipatrones: una Introducción - Parte I . *msdn*. [En línea] [Citado el: 10 de 4 de 2013] <http://msdn.microsoft.com/es-es/library/bb972242.aspx>.
18. Manuales y guías prácticas sobre patrones de diseño. *scribd*. [En línea] [Citado el: 21 de 4 de 2013] <http://es.scribd.com/doc/110474646/patrones-de-diseno-en-uml>.
29. Highcharts: Librería de gráficos para Javascript . *WebLatam*. [En línea] [Citado el: 9 de 12 de 2012.] <http://weblatam.com/wp/highcharts-libreria-de-graficos-para-javascript/>.
20. JasperReports Library. *jaspersoftCommunity*. [En línea] [Citado el: 20 de 5 de 2013] <http://community.jaspersoft.com/project/jasperreports-library>.
21. Una Introducción a Apache. *Linux ciberaula*. [En línea] [Citado el: 13 de 12 de 2012] http://linux.ciberaula.com/articulo/linux_apache_intro.
22. Sitio de descarga de software(VP). [En línea] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
23. **Daniel Sierra**. slideshare. *slideshare*. [En línea] [Citado el: 2 de 12 de 2012] <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
24. **Durán, A. R.** (2011). Impacto del Generador Dinámico de Reportes en los Sistemas de Gestión de Información. *Serie Científica de la Universidad de las Ciencias Informáticas*, 12.
25. **González, Y. L.** (2011, 07 15). alasMEDIGEN 1.2:Sistema Informático de Genética Médica. *Serie Científica de la Universidad de las Ciencias Informáticas*, 4(7), 10.
26. Modelo de Dominio. [En línea] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
27. Requisitos. [En línea] www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf.
28. **Colective, A.** (2003). Lives to Live (June 2003 ed.). La Habana: Casa Editora Abrir.
29. Requisitos Funcionales. [En línea] <http://es.scribd.com/doc/58715996/Requisitos-Funcionales>.
- itar 32. Caso de Uso. [En línea] www.2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf.
30. **Elena Mediavilla**. Definición de los Casos de Uso del Sistema. *ctr*. [En línea] [Citado el: 25 de 4 de 13] http://www.ctr.unican.es/asignaturas/MC_OO/Doc/Casos_de_uso.pdf.

31. **Miguel Vega**. Casos de Uso UML. Lsi-ugr. [En línea] [Citado el: 26 de 4 de 13] <http://lsi.ugr.es/~mvega/docis/casos%20de%20uso.pdf>
32. **DrC. Luis Carlos Silva**. Vista lógica de la arquitectura . *informatica2009*. [En línea] [Citado el: 3 de 5 de 2013] www.informatica2009.sld.cu/.
33. **Yupanqui García, Glen Jasper**. diagrama de clase del diseño. *Scribd*. [En línea] [Citado el: 10 de 3 de 2013] <http://es.scribd.com/doc/3930805/Patrones-de-Diseno>.
34. diagrama de clase del diseño1. *Scribd*. [En línea] [Citado el: 10 de 3 de 2013] <http://es.scribd.com/doc/23941988/Los-patrones-de-diseno>.
35. **Mario Solarte**. Introducción a los patrones de diseño. *slideshare*. [En línea] [Citado el: 10 de 3 de 2013] <http://www.slideshare.net/Enf2IST/introduccion-a-los-patrones-de-diseno>.
36. **Rodrigo Salinas**. patrones de diseño. *rodrigosalinas.cl*. [En línea] [Citado el: 13 de 3 de 2013] <http://www.rodrigosalinas.cl/alexander-arquitectura-y-agilidad-3/>.
37. GRASP. *practicadesoftware*. [En línea] [Citado el: 12 de 3 de 2013] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
38. GOF. *geektheplanet*. [En línea] [Citado el: 13 de 3 de 2013] <http://geektheplanet.net/5462/patrones-gof.xhtml>.
39. Diagrama de interacción. *inti.gob.ar*. [En línea] [Citado el: 16 de 3 de 2013] www.inti.gob.ar/cirsoc/pdf/ejemplos/interaccion2_ejemplos201.pdf.
40. Diagrama de interacción1. *slideshare*. [En línea] [Citado el: 16 de 3 de 2013] <http://www.slideshare.net/jent46/diagramas-de-interaccion-presentation-892165>.
41. diagrama de despliegue. *scribd*. [En línea] [Citado el: 5 de 4 de 2013] <http://es.scribd.com/doc/19611312/diagramas-de-despliegue-2222>.
42. **Alberto Zurita**. Diagrama de despliegue. *slideshare*. [En línea] [Citado el: 22 de 4 de 2013] <http://www.slideshare.net/albertozurita96/diagrama-de-despliegue-17071673>.
43. **Emilio Aviles Avila**. diagrama de despliegue. *slideshare*. [En línea] [Citado el: 3 de 4 de 2013] <http://www.slideshare.net/techmi/curso-uml-25-diagramas-de-implementacion>.
44. **Juan Pavón Mestras**. patrón mvc 1. *fdi.ucm*. [En línea] [Citado el: 22 de 3 de 2013] <http://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf>.
45. patrón mvc. *prestashop5estrellas*. [En línea] [Citado el: 2 de 4 de 2013] <http://prestashop5estrellas.wordpress.com/2010/03/29/el-patron-mvc-modelo-vista-controlador/>.

46. **Joshell**. diagrama de componente. *slideshare*. [En línea] [Citado el: 10 de 4 de 2013] www.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue.
47. diagrama de componente1. *msdn*. [En línea] [Citado el: 14 de 3 de 2013] msdn.microsoft.com/es-es/library/dd409390.aspx.
48. Caso de prueba. *ehowenespanol*. [En línea] [Citado el: 2 de 5 de 2013] http://www.ehowenespanol.com/escibir-caso-prueba-software-manual-paso-paso-como_180800/.
49. **Guillermo Lemus**. tipos de pruebas de software . *slideshare*. [En línea] [Citado el: 12 de 5 de 2013] <http://www.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software>.
50. **Ricardo Mansilla**. pruebas de software . *slideshare*. [En línea] [Citado el: 11 de 5 de 2013] <http://www.slideshare.net/cliceduca/pruebas-de-software-2420588>.
51. Técnicas de prueba . *indalog.ual*. [En línea] [Citado el: 10 de 5 de 2013] <http://indalog.ual.es/mtorres/LP/Prueba.pdf>.
52. Técnicas de Mantenimiento de Programas: Técnica dePrueba Caja Negra . *scribd*. [En línea] [Citado el: 10 de 5 de 2013] <http://es.scribd.com/doc/59836568/Tecnicas-de-caja-negra>.
53. **Consortio**. No conformidades. *slideshare*. [En línea] [Citado el: 10 de 5 de 2013] <http://www.slideshare.net/consorciolineauno/entrenamiento-de-control-de-no-conformidad-acciones-correctivas-preventivas-y-mejoria>.
54. no conformidades. *saitosoft*. [En línea] [Citado el: 9 de 5 de 2013] http://saitosoft.mx/?page_id=315el .