

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 6**



Título: “Sistema Informático de Genética Médica alasMEDIGEN. Módulo de reportes.”

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

Autores:

Nilaisy Villamil Abad
Yoandy González Valdés

Tutores:

Ing. Yudiel La Rosa González
Ing. Aurelio Rodríguez Durán

Co-Tutor(a):

MsC. Elvismary Molina De Armas

La Habana, junio de 2012

“Año 54 de la Revolución”



"Para el logro del triunfo siempre ha sido indispensable pasar por la senda de los sacrificios."

Simón Bolívar

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 _____.

Nilaisy Villamil Abad

Firma del autor

Yoandy González Valdés

Firma del autor

Ing. Yudiel La Rosa González

Firma del tutor

Ing. Aurelio Rodríguez Durán

Firma del tutor

MsC. Elvismary Molina De Armas

Firma del Co-Tutor:

Tutores:

Ing. Ing. Yudiel La Rosa González

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: yglarosag@uci.cu

Ing. Aurelio Rodríguez Durán

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: arduran@uci.cu

Co-Tutor:

MsC. Elvismary Molina De Armas

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: emolina@uci.cu

AGRADECIMIENTOS

Agradecemos a nuestro comandante en jefe Fidel Castro y a la Revolución Cubana por darnos la oportunidad de estudiar en esta universidad de excelencia. A nuestros familiares que siempre están ahí cuando los necesitamos. A los tutores Elvismary, Yudiel y Aurelio que tanto nos apoyaron. A nuestros amigos por toda su comprensión en los momentos difíciles. A los compañeros del proyecto y a todos los que nos ayudaron en nuestra formación.

DEDICATORIA

De Nilaisy:

Dedico esta tesis a mis maravillosos y excepcionales padres, que ambos por igual me han brindado, en primer lugar, el derecho a la vida, y a partir de esta una excelente familia, confianza, educación, apoyo y mucha comprensión. Este trabajo también va dedicado a mi querido y adorable hermano, y a toda mi familia en general, que sin saberlo, en cada momento de mi vida, han sembrado en mí semillas de esperanzas y seguridad para alcanzar mis metas. Pero no quisiera dejar de mencionar a mis amigas Diana María y Yadisnelys, con las cuales he pasado uno de los mejores momentos de mi vida, la vida universitaria.

De Yoandy:

El fruto de 5 años de estudios se ven materializados en este trabajo diploma, el cual quisiera dedicárselo especialmente a quien durante toda mi vida me ha acogido como un hijo más, a esa persona que sin pedir nada a cambio me sabido dar lo mejor de sí, muchas gracias por todo MI PADRE ANDRÉS, llegue también esta dedicatoria a mi madre, mis hermanos y a mi abuela, que sin su apoyo no hubiese encontrado las fuerzas para terminar la carrera. A mis amigos, los que están y los que lamentablemente no se encuentran hoy pero que estoy seguro comparten mi alegría, a mi hermano aquí en la Habana esta va para ti también MI HERMA, a mi familia en general y a mis vecinos que tanto me han apoyado. Por último y no menos importante gracias por la comprensión a aquellos que deberían estar hoy aquí ocupando su lugar y por razones más fuertes no han podido estar, para ellos va dedicado también este trabajo.

RESUMEN

En el Centro Nacional de Genética Médica (CNGM) 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 al que se le realiza la investigación. Los mismos son almacenados y gestionados desde el Sistema Informático de Genética Médica alasMEDIGEN.

Actualmente el sistema cuenta con una serie de reportes que representan un bajo por ciento de la información que los genetistas necesitan. Esta insuficiencia de reportes impide realizar un estudio más detallado de los datos recogidos. Por otro lado, los especialistas no disponen de un mecanismo que les permita la obtención de reportes que faciliten un mejor análisis de los datos, es decir, el usuario apenas introduce criterios de búsqueda en la mayoría de los casos, y en el resto de ellos es muy poca información la que ofrece sus resultados, impidiendo una correcta toma de decisiones.

El nuevo módulo Reportes Genéticos de la aplicación informática alasMEDIGEN representará solo cinco módulos de los ocho anteriormente existentes en el sistema. Esta cantidad se debe a las prioridades definidas por el cliente. Este módulo permitirá una mejor visión de los distintos problemas genéticos que afronta la población cubana. Brindará además un sistema rico tanto en cantidad de reportes como en calidad de los resultados que estos pueden arrojar al usuario final.

Palabras Claves: alasMEDIGEN, reporte, reporte dinámico.

CONTENIDO

AGRADECIMIENTOS	I
DEDICATORIA.....	II
RESUMEN.....	III
Introducción	1
Capítulo 1: Fundamentos Teóricos.....	5
1.1 alasMEDIGEN: Sistema Informático de Genética Médica.....	5
1.2 Reporte	6
1.2.1 Generador de Reportes	6
1.2.2 Generación de Reportes	6
1.3 Reporteadores dinámicos.....	7
1.3.1 Crystal Report.....	7
1.3.2 Jasper Reports.....	7
1.3.3 Report Manager	8
1.3.4 InstantAtlas	8
1.3.5 Generador Dinámico de Reportes 1.7	9
1.4 Marco Regulatorio	10
1.5 Metodología de desarrollo	10
1.5.1 Proceso Unificado de Racional (RUP)	10
1.6 Herramienta de Modelado CASE.....	11
1.6.1 ArgoUML.....	11
1.6.2 Rational Rose	12
1.6.3 Visual Paradigm.....	13
1.7 Lenguaje de Programación.....	14
1.7.1 PHP	14
1.8 Marco de Desarrollo	15
1.8.1 Symfony.....	15
1.9 Servidor Web.....	15
1.9.1 Servidor Web Apache	16
1.10 Gestor de Base de Datos	16
1.10.1 MySQL.....	16
1.11 Controlador de Versiones	17
1.11.1 Subversion	17
1.12 Entorno de Desarrollo Integrado.....	18
1.12.1 Eclipse	18
1.12.2 NetBeans	19
1.15 Conclusiones Parciales	20
Capítulo 2: Análisis y diseño de la solución.....	21
2.1. Objeto de estudio	21
2.1.1. Objetivos estratégicos de la organización	21
2.1.2. Flujo actual de los procesos.....	21
2.1.3. Análisis crítico de la ejecución de los procesos	22
2.2. Objeto de automatización	22
2.3. Modelo de Dominio.....	22
2.3.1. Diagrama de clases modelo de dominio.....	23
2.3.2. Descripción de las clases del modelo de dominio	23
2.3.3. Reglas del negocio	24
2.4. Especificación de los requerimientos de la aplicación informática	24
2.4.1. Requerimientos funcionales	24
2.4.2. Requerimientos no funcionales	29

2.5 Definición de los Casos de Uso del Sistema.....	31
2.5.1 Actores del Sistema	31
2.5.2 Diagrama de Casos de Uso del Sistema.....	31
2.5.3 Descripción a Alto Nivel de los Casos de Uso del Sistema.....	32
2.5.3 Descripción del Caso de Uso Extendido Generar Reportes de Discapacidad Intelectual ...	34
1.13 Patrones de Diseño	38
1.13.1 Patrones GRASP	39
2.13.2 Patrones de diseño GOF.....	40
1.14 Patrones de Arquitectura	41
1.14.2 Patrón Modelo Vista Controlador (MVC)	41
2.6 Diagramas de clases del diseño	42
2.6.1 Vista Lógica de la Arquitectura	43
2.6.2 Diagrama de Clases de Diseño CU Generar Reportes de Discapacidad Intelectual	43
2.6.3 Aplicación de Patrones de Diseño.....	45
2.6.4 Integración del GDR con alasMEDIGEN	46
2.7 Diagramas de Interacción	46
2.8 Modelo de despliegue	48
2.9 Conclusiones Parciales	49
Capítulo 3: Implementación y pruebas de la solución.	50
3.1 Implementación	50
3.1.1 Diagrama de Componentes	50
3.2 Pruebas de Software	51
3.2.1 Métodos de pruebas de software	51
3.2.2 Tipo de pruebas de software	52
3.2.3 Diseño de Caso de Pruebas y Registro de No Conformidades	52
3.2.3 Resultado Obtenidos.....	56
3.3 Conclusiones Parciales	57
CONCLUSIONES	58
Recomendaciones	59
Referencias.....	60
Bibliografía.....	62
Glosario de Términos.....	65

Introducción

La Genética Humana es la ciencia que estudia la variación entre los seres humanos basada en las diferencias existentes en el material hereditario, mientras que la Genética Médica se ocupa de la aplicación de estos principios a la práctica médica y del estudio del papel de los genes en el origen de los rasgos humanos y de las enfermedades. (1) Este es el proceso en el cual un padre transmite ciertos genes a sus hijos, logrando como resultado la similitud en la apariencia de una persona, estatura, color del cabello, color de piel y de los ojos además de incluir la probabilidad de contraer ciertas enfermedades, capacidades mentales y talentos naturales. Su objeto de estudio son los patrones de herencia, es decir, el modo en que los rasgos y las características se transmiten de padres a hijos. (2)

La informática es la ciencia que tiene como objetivo estudiar el tratamiento automático de la información a través de la computadora. Esta definición, si bien es bastante general, se debe a que el concepto de informática también es amplio. (3)

Con el cursar de los años se ha comprobado que el estudio de la genética humana es una tarea muy tediosa y difícil sin las herramientas informáticas, puesto que el genoma contiene miles de genes. Los sistemas informáticos son útiles, pues muestran de forma electrónica, mediante combinaciones matemáticas, el proceso biológico que se produce en el organismo del ser humano. El uso de las tecnologías junto a las investigaciones de genética que han realizado los especialistas en la rama anteriormente mencionada, dan origen a una nueva y revolucionaria etapa en el desarrollo de la genética médica en el país, vinculado también a la aparición y uso de la Internet. Aunque parezca insólito, estas dos áreas son inseparables, hasta el punto de que sin las computadoras los científicos no habrían podido descifrar el genoma humano, el libro de la vida.

Con el objetivo 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 actúa como centro de referencia nacional para el Programa Cubano de Diagnóstico, Manejo y Prevención de Enfermedades Genéticas y Defectos Congénitos. Dirige la red de centros de genética médica del país, cumpliendo tareas asistenciales, docentes e investigativas en el campo de la genética médica y la inmunología. (4)

La Universidad de Ciencias Informáticas inicia su primer curso el 23 de septiembre del 2002. Este centro tiene como misión formar profesionales, comprometidos con su Patria, calificados en la rama de la Informática, a partir de un modelo pedagógico flexible, que vincula dinámica y coherentemente el

estudio con la producción y la investigación, acorde con las necesidades sociales del país y de otros pueblos hermanos.

En el año 2008 la UCI conjuntamente con el CNGM decidieron desarrollar el Sistema Informático de Genética Médica (alasMEDIGEN), una herramienta para la gestión de la información asociada a las consultas de genética médica y a los diferentes estudios que se realizan en el CNGM.

El gran volumen de información generado por los estudios de genética es almacenado y gestionado mediante el sistema informático alasMEDIGEN. Este cuenta con 8 registros:

- Registro Cubano de Historias Clínicas Genéticas (RECUHCL)
- Registro Cubano de Enfermedades Genéticas (RECUEGEN)
- Registro Cubano de Malformaciones Congénitas (RECUMAC)
- Registro Cubano de Discapacitados (RECU DIS)
- Registro Cubano de Discapacidad Intelectual (RECURM)
- Registro Cubano de Gemelos (RECUGEM)
- Registro Cubano de Anomalías Cromosómicas (RECUAC)
- Registro Cubano de Enfermedades Comunes (RECUEC)

El sistema dispone actualmente de 12 reportes de los diferentes módulos, los cuales representan un bajo por ciento de la información demandada por los genetistas. Esta situación impide realizar un análisis más detallado de los datos recogidos. Generalmente a la hora de realizar un nuevo estudio genético se requiere obtener nueva información que no está presente en los reportes existentes, ya que son preestablecidos en el sistema. Esta nueva información puede estar relacionada con el total de personas huérfanas que presentan discapacidad intelectual, total de discapacitados físicos asociados, conocer sus condiciones de vida, o saber la funcionalidad de la familia en gemelos, entre otros. Sin embargo, no se cuenta con un mecanismo que permita a los especialistas crear nuevos reportes dinámicos.

Luego de analizadas las dificultades existentes en el centro surge el siguiente **problema a resolver**: ¿Cómo contribuir a la obtención de información en el sistema alasMEDIGEN que faciliten un mejor análisis de las enfermedades genéticas y defectos congénitos?

Teniendo como **objeto de estudio** el proceso de generación de reportes, constituyendo como el **campo de acción** el proceso de generación de reportes en el sistema alasMEDIGEN.

Para satisfacer las necesidades del centro, el trabajo persigue un **objetivo general**: Desarrollar el módulo de reportes del sistema alasMEDIGEN para contribuir a la obtención de reportes que faciliten un mejor análisis de las enfermedades genéticas y defectos congénitos, del cual se derivan varios **objetivos específicos**:

1. Realizar el estudio relacionado con los sistemas generadores de reportes y la arquitectura del sistema alasMEDIGEN.
2. Realizar el análisis y diseño del módulo de reportes del sistema alasMEDIGEN.
3. Implementar el módulo de reportes del sistema alasMEDIGEN.
4. Validar el funcionamiento del módulo de reportes del sistema alasMEDIGEN.

En aras de realizar una correcta investigación y dar cumplimiento a los objetivos antes descritos se definen las siguientes **tareas de la investigación**:

- Análisis del marco conceptual relacionado con los sistemas generadores de reportes, el proceso de desarrollo de software y la arquitectura del sistema alasMEDIGEN.
- Selección de las herramientas y metodología a utilizar en el desarrollo del módulo de reportes del sistema alasMEDIGEN.
- Análisis del módulo de reportes del sistema alasMEDIGEN.
- Diseño del módulo de reportes del sistema alasMEDIGEN.
- Implementación de la interfaz gráfica del módulo de reportes del sistema alasMEDIGEN.
- Implementación de algoritmos internos y las funciones de procesamiento del módulo de reportes del sistema alasMEDIGEN.
- Diseño de los casos de prueba correspondientes al módulo de reportes del sistema alasMEDIGEN.
- Ejecución de las pruebas del módulo de reportes del sistema alasMEDIGEN.

Después de todo el proceso de investigación e implementación de la solución se espera como **posibles resultados**: que el módulo de reportes del sistema alasMEDIGEN contribuya a la obtención de reportes que faciliten un mejor análisis de las enfermedades genéticas y defectos congénitos en la Red Nacional de Genética Médica. Para ello la estructuración del trabajo de diploma se verá en tres capítulos:

Capítulo 1: Fundamentos Teóricos.

Este capítulo brinda una breve explicación de los reportadores dinámicos, alguna de sus características, así como la importancia y ventajas de su utilización para contribuir a la obtención de

reportes que faciliten un mejor análisis de los datos en el sistema alasMEDIGEN. También se abordan temas referentes los últimos avances, permitiendo una mejor visión del desarrollo de esta herramienta, así como la descripción de los conceptos, metodologías, herramientas, tecnologías y procedimientos tomados de la bibliografía. Además, aborda las tendencias y tecnologías de los sistemas reporteadores actuales para dar solución al problema propuesto.

Capítulo 2: Análisis y diseño de la solución.

Este capítulo se aborda lo referente al funcionamiento del negocio: las reglas, la descripción del mismo a través de un modelo de dominio con su descripción pertinente, y las mejoras que se proponen. Se establecen los requerimientos funcionales detectados a partir de las necesidades del cliente, así como los requerimientos no funcionales que de alguna manera mantendrán al usuario identificado y satisfecho con la aplicación. Se exponen también todos los diagramas que intervienen en el diseño de la solución: diagrama de casos de uso del sistema, diagrama de clases del diseño, diagrama de secuencia y diagrama de despliegue.

Capítulo 3: Implementación y pruebas de la solución.

En este capítulo se analiza el modelo de implementación y se describe el diagrama de componentes. Además, se dará una breve explicación de las pruebas realizadas para poder validar el correcto funcionamiento de la aplicación.

Capítulo 1: Fundamentos Teóricos

Introducción

Para lograr un producto de calidad se requiere el dominio de los aspectos más importantes que giran en torno a este. Por tal motivo en este capítulo se reflejará el estudio realizado sobre los reportadores dinámico: características, importancia y ventajas de su uso para contribuir a la creación de reportes dinámicos. También se hace necesario el análisis de los conceptos asociados a metodologías, herramientas, tecnologías y procedimientos con el objetivo de seleccionar las que más aporten al desarrollo del Módulo de Reportes del sistema alasMEDIGEN.

1.1 alasMEDIGEN: Sistema Informático de Genética Médica

El sistema alasMEDIGEN fue creado en el año 2008. Tiene como objetivo recoger los datos obtenidos por los diferentes estudios genéticos que se llevan a cabo en el Centro Nacional de Genética Médica. La orientación a servicios que presenta alasMEDIGEN agiliza el desarrollo de muchísimas investigaciones científicas, dado que estas necesitan datos brindados por el sistema como son: la cantidad de gemelos de una provincia o lugar determinado, las historias clínicas de ciertos pacientes, el listado de las enfermedades genéticas, entre otros. Este sistema fue realizado completamente en PHP5, usando el marco de desarrollo Symfony, como gestor de base de datos MySQL y como servidor de aplicaciones Apache. Constituye una base de datos de 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 esta información pueda arrojar se puede realizar una mejor prevención y tratamiento de las enfermedades genéticas, que son gestionadas a través de los reportes existentes en el sistema. Estos reportes están asociados a las anomalías cromosómicas, enfermedades comunes, malformaciones congénitas, discapacitados, gemelos, historias clínicas, enfermedades genéticas y discapacidad intelectual. Estos reportes se encuentran distribuidos por módulos. Se caracterizan por ser estáticos, tener muchas líneas de código, realizar consultas MySQL manualmente en el código del sistema, y además están definidos para devolver datos superficiales, de los cuales el usuario no introduce criterios de búsqueda en la mayoría de los casos, y en el resto de ellos es muy poca la información que ofrecen sus resultados.

1.2 Reporte

Un reporte es un documento, generado por el sistema, que 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. Los reportes generalmente agrupan los datos de acuerdo a un interés específico. (5)

En el ámbito de la informática, los reportes son informes que organizan y exhiben la información contenida en una base de datos a partir de una consulta realizada a la misma. 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.

1.2.1 Generador de Reportes

Para lograr obtener un reporte con calidad y según las necesidades del cliente, se hace necesario el empleo de un generador de reportes. Este no es más que un programa que prepara un informe o la exhibición de datos. Dicho programa permite seleccionar en una base de datos la información necesaria para construir el informe.

Con el uso de un generador de reportes se produce en forma automática diversos reportes.

En otras palabras, es un programa que ha sido diseñado por una empresa dedicada a la producción de software en respuesta a la demanda de este tipo de herramientas que optimizan la producción de información.

1.2.2 Generación de Reportes

La realización de esta función es esencial para un sistema de información, ella se encarga de producir la información requerida y trasmitirla a los puntos o centros de información que la soliciten. Esta transmisión de información se puede efectuar mediante el movimiento físico de los elementos de almacenamiento (papel, cintas magnéticas, disquete, y otros) o mediante la comunicación de señales eléctricas digitales o analógicas a dispositivos receptores (terminales, convertidores, estaciones remotas u otro computador).

En el sistema alasMEDIGEN la generación de reportes es una funcionalidad muy importante, principalmente para los especialistas a la hora de realizar algún estudio o diagnóstico. Esta acción les facilita la toma de decisiones con respecto a cualquier trastorno que pueda presentar un paciente determinado. (6)

1.3 Reporteadores dinámicos

Como parte de la investigación se realiza el análisis de varios reporteadores existentes con el fin de seleccionar el más apropiado para la solución.

1.3.1 Crystal Report

Crystal Reports es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos (bases de datos). Una de las características que distingue a Crystal Reports del resto de generadores de informes es la posibilidad de combinar datos y gráficos dentro del mismo informe de una forma natural. Asimismo, la optimización de consultas SQL permite que el tiempo de generación de los informes sea mínimo, proporcionando mayor velocidad a la hora de obtener los datos e indicadores deseados.

Varias aplicaciones, como Microsoft Visual Studio, incluyen una versión OEM (original equipment manufacturer, del español fabricante de equipamiento original) de Crystal Reports como una herramienta de propósito general de informes/reportes. Crystal Reports se convirtió en el escritor de informes estándar cuando Microsoft lo liberó con Visual Basic. Es la herramienta más popular en su categoría, y no solo entre quienes podrían considerarse sus usuarios puros (aquellos que necesitan obtener periódicamente información para la toma de decisiones a partir de los datos de la empresa), sino también entre los programadores, que lo han convertido en su herramienta favorita a la hora de embeber capacidades de generación de informes dentro de las aplicaciones a medida.

Tiene varias versiones que se han ido perfeccionando en su uso, la más actual de todas es la Crystal Reports XI (once), pero su mejor versión es el Crystal Reports 2008 que satisface el más amplio rango de requisitos de usuarios y aplicaciones. A pesar de sus admirables ventajas, presenta algunos aspectos que dificultan su empleo, algunos de ellos vienen dados por presentar un elevado coste adquisitivo, no ser multiplataforma, no estar orientada completamente al usuario final y usar licencia propietaria. (7)

1.3.2 Jasper Reports

Jasper Report es una de las herramientas más utilizadas de código libre en Java para generar reportes. Puede entregar ricas presentaciones o diseños en la pantalla, para la impresora o para archivos en formato PDF, HTML, RTF, XLS, CSV y XML.

Para su utilización se requiere tener instalado en el equipo el JDK 1.4 (SDK) o posterior. No basta con tener instalado el J2RE (Run Time Environment), las siguientes librerías junto con la de Jasper Reports deben incluirse en el proyecto en que se desee incluir esta herramienta para generar reportes:

- Driver JDBC 2.0 (Usualmente incluido en el SDK).
- PDF. Librería libre Java-PDF iText por Bruno Lowagie y Paulo Soares.
- XLS.

Está completamente escrita en Java y se puede utilizar en una gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web, para generar contenido dinámico. (8) A pesar de ser una potente herramienta el hecho de estar escrita en Java se torna un inconveniente para ser integrado a la solución, pues el marco regulatorio del Grupo de Integración de Soluciones (GIS) establece como lenguaje de programación PHP para las aplicaciones de salud.

1.3.3 Report Manager

Report Manager es una aplicación de generación de informes y un conjunto de componentes para Delphi, Builder y Kylix, también puede utilizarse desde otros entornos de desarrollo con el componente ActiveX incluido (Visual Basic, Visual FoxPro, cualquier lenguaje de Visual Studio.Net), se proporciona una librería dinámica estándar con funciones para su uso con cualquier lenguaje como GNU C. Funciona en Windows y Linux. Puede distribuir el diseñador de informes, con esto consigue la modificación de informes sin modificar (recompilar) su aplicación, el resultado puede ser guardado como un archivo con formato PDF.

Tiene como limitante que es un producto de código abierto bajo el modelo MPL (Mozilla Public License), se incluye permiso de uso en aplicaciones GPL (General Public License), por lo que puede usarlo en sus aplicaciones comerciales, pero cualquier mejora introducida en el motor de impresión debe ser publicada bajo esta licencia. A esto se le suma el hecho de ser una aplicación de escritorio implicando su instalación en cada máquina cliente. (9) Por esos motivos no se hace efectiva su utilización.

1.3.4 InstantAtlas

InstantAtlas es una poderosa y económica herramienta de visualización y presentación de datos que mejora el valor de estos al hacer que los patrones en la información sean fáciles de observar y

explorar. No se requiere experiencia en publicación de datos o en mapeo, aun así los resultados pueden ser increíbles y efectivos.

Consta de un conjunto de herramientas para diseñar, poblar y publicar reportes dinámicos (Templates, Publisher, Data Managers). Los reportes dinámicos de InstantAtlas se crean a partir de plantillas que están listas para usar, adquiriendo solo la que se necesita para desplegar los datos. No se requiere desarrollo. Los reportes de InstantAtlas son páginas web autónomas. No es necesario instalar software en el servidor web. Los reportes pueden ser publicados en un sitio web, en un sistema de manejo de contenido, pueden ser colocados para ser descargados, enviados vía email o distribuidos en un CD. (10) Pero para su uso es necesario solicitar una licencia de evolución de InstantAtlas, de aquí que tampoco es muy factible su utilización en el sistema.

1.3.5 Generador Dinámico de Reportes 1.7

El Generador Dinámico de Reportes (GDR) es un sistema que se está desarrollando en la UCI desde el año 2008, y en el año 2010 fue liberado la versión 1.7 del producto. Presenta una arquitectura basada en componentes, que permite distribuir 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. Es una aplicación multiplataforma, implementada en PHP5, y posee gran flexibilidad permitiendo al usuario final el diseño de reportes personalizados desde la Web. Cuenta con las funciones necesarias para diseñar los reportes de manera interactiva y aprovecha las ventajas de la web. Los reportes son basados en un fichero XML que contiene la definición del reporte. Dicha definición es el núcleo del sistema ya que comunica las diferentes aplicaciones y facilita que las mismas se integren conformando finalmente el reporte final. (11)

Después del estudio realizado de varios reporteadores, se determinó utilizar el reporteador dinámico GDR 1.7 para darle solución al problema a resolver. Además de las características mencionadas anteriormente, también se tuvo en cuenta que al tratarse de una tecnología propia del país y de la universidad se facilitan importantes aspectos como el mantenimiento continuo, costo de adquisición, personalización e integración de la misma con otros sistemas, este último gracias a la existencia de una API.

1.4 Marco Regulatorio

El proceso de informatización de la Salud Pública se lleva a cabo, como parte de la informatización de la sociedad, aplicando las Tecnologías de la Información y las Comunicaciones (TIC), con lo cual las instituciones de salud del país deben lograr un incremento de la calidad, efectividad y eficiencia de los servicios que prestan a la población y una mayor satisfacción del trabajo de salud.

En estos momentos se trabaja integradamente en el desarrollo de un grupo de aplicaciones básicas acordes a la nueva estrategia. En su desarrollo e implementación participan la Dirección Nacional de Informática del MINSAP (DI), INFOMED, las Direcciones Nacionales del Ministerio de Salud Pública, Softel, otras Empresas del Ministerio de la Informática y las Comunicaciones y la Universidad de las Ciencias Informáticas.

En esta nueva etapa se ha definido por la DI MINSAP las políticas y estrategias de desarrollo, especificaciones a cumplir, arquitectura, estándares, requerimientos y regulaciones; que garanticen la continuidad y sostenibilidad de los productos que se obtengan, los cuales se exponen en este documento. (12)

1.5 Metodología de desarrollo

Una metodología de desarrollo de software permite definir quién debe hacer qué, cuándo y cómo debe hacerlo. Una metodología es un proceso que puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología ayuda al desarrollo del software lo guía en la obtención de los distintos productos parciales y finales. (13)

1.5.1 Proceso Unificado de Racional (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

1. Inicio: el objetivo en esta etapa es determinar la visión del proyecto.
2. Elaboración: en esta etapa el objetivo es determinar la arquitectura óptima.
3. Construcción: en esta etapa el objetivo es llegar a obtener la capacidad operación al iniciar.
4. Transición: el objetivo es llegar a obtener el producto final.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en

función de la evaluación de las iteraciones precedentes.

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, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo y a que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
- Iterativo y de Incremento: RUP propone que cada fase se desarrolle en iteraciones. 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. (14)

Por tanto se asume utilizar para el desarrollo de la aplicación la metodología RUP, ya que hace exigente el uso de artefactos. Es una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software, y además es la metodología que se ha utilizado desde el inicio del proyecto.

1.6 Herramienta de Modelado CASE

La herramienta CASE (Computer Aided Software Engineering, del español Ingeniería de Software Asistida por Computadoras) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software. Algunas herramientas CASE conocidas son el ArgoUML, Rational Rose, Visual Paradigm, Xcase.

1.6.1 ArgoUML

ArgoUML fue concebido como un entorno y herramienta para usar en el análisis y diseño de sistemas de software orientados a objeto. Con ArgoUML se es capaz de crear nueve tipos de diagramas:

diagrama de clases, diagrama de estados, diagrama de actividad, diagrama de casos de uso, diagrama de colaboración, diagrama de despliegue, diagrama de secuencia.

Características:

- Es una aplicación Java pura 100%. Java fue concebido como un lenguaje interpretado.
- Compila código para su propio sistema, la Máquina Virtual Java. En caso de que se pregunte porque todos los lenguajes no son como este, es porque los lenguajes interpretados tienden a ser más lentos que los compilados. Mediante la elección de escribir ArgoUML en Java puro, se hace inmediatamente disponible para el mayor número de usuarios con la mínima cantidad de esfuerzo. Esto permite a ArgoUML funcionar en todas las plataformas.
- Es un proyecto de código abierto. Esto significa que cualquiera puede tener una copia gratis del código fuente, cambiarlo, usarlo para nuevos propósitos y cosas así. La única (gran) obligación es que usted pase su código de la misma forma a otros.

Requerimientos del Sistema:

- Puesto que ArgoUML está escrito en Java, debería funcionar en cualquier máquina con Java instalado. Es necesaria una versión 1.4 o posterior de Java. ArgoUML necesita una cantidad razonable de recursos.
- PC con 200MHz de procesador.
- 64Mb de RAM.
- 10Mb de espacio disponible en un disco duro. (15)

1.6.2 Rational Rose

Rational Rose Enterprise es una aplicación para desarrollo de arquitectura base del Software. Es una herramienta de desarrollo basada en modelos que se integra con las bases de datos y los IDE de las principales plataformas del sector. IBM Rational Rose Enterprise es uno de los productos más completos de la familia Rational Rose. Todos los productos de Rational Rose dan soporte a UML (Unified Modeling Language, del español Lenguaje Unificado de Modelado), pero no son compatibles con las mismas tecnologías de implementación. Al igual que todos los productos de Rational Rose, ofrece un lenguaje de modelado común que agiliza la creación del software.

Características:

- Característica de control por separado de componentes, modelo que permite una administración más granular y el uso de modelos.
 - Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
 - Soporte Enterprise Java Beans™ 2.0.
 - Capacidad de análisis de calidad de código.
 - Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
 - Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
 - Integración con otras herramientas de desarrollo de Rational.
 - Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational Clear Case.
 - Publicación web y generación de informes para optimizar la comunicación dentro del equipo.
- (16)

A pesar de las valiosas características que presenta esta herramienta, el hecho de ser un software privativo impide su utilización como parte de la solución propuesta.

1.6.3 Visual Paradigm

Visual Paradigm es una de las herramientas UML CASE del mercado profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Se caracteriza por:

- Soporta aplicaciones Web.
- Varios idiomas.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

- Diagramas de procesos de negocio - proceso, decisión, actor de negocio, documento.
- Modelado colaborativo con Subversion.
- Ingeniería de ida y vuelta.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Editor de figuras. (17)

Por lo antes analizado en el trabajo de diploma se hace uso de Visual Paradigm para UML en su versión 8.0 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. Soporta las últimas versiones del mismo. Provee el modelado de procesos de negocios, además de un generador de mapeo de objetos-relacionales para el lenguaje de programación PHP.

1.7 Lenguaje de Programación

1.7.1 PHP

El lenguaje Preprocesador de Hipertexto (Hypertext Preprocessor), es un lenguaje de código abierto interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

Entre algunas de sus características se destacan:

- El soporte de cookies HTTP de forma transparente. Las Cookies son un mecanismo por el cual se almacenan datos en el navegador remoto y así rastrear o identificar a usuarios que vuelven.
- El soporte de sesiones consiste en una manera de guardar ciertos datos a través de diferentes accesos web. Esto permite crear aplicaciones más personalizadas y mejorar las características del sitio web. Toda la información está en la sección.
- Manejo de Conexiones Internamente en PHP se mantiene un estado de la conexión que puede ser normal mientras se ejecute un script cualquiera, abortado referente a si un cliente remoto se desconecta, y tiempo de espera si el tiempo límite para la conexión expira.
- El código es ejecutado en el servidor, generando HTML y enviándolo al cliente.
- Ofrece soporte para una gran cantidad de bases de datos como por ejemplo: mSQL, MySQL,

PostgreSQL, Mysqli, siendo esta última una extensión mejorada de MySQL.

Ventajas

- Extremadamente simple para el principiante.
- Ofrece muchas características avanzadas para los programadores profesionales.
- Permite procesar la información de formularios.
- Generar páginas con contenidos dinámicos.
- Enviar y recibir cookies. (18)

1.8 Marco de Desarrollo

1.8.1 Symfony

Symfony es un framework o marco de trabajo que simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear un código más legible y más fácil de mantener. Divide la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Está desarrollado completamente con PHP5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux) como en plataformas Windows. (19) Es por todas las características antes mencionadas que se decide trabajar con este marco en su versión 1.0.22, que es la más estable actualmente de la versión 1.0, además que el mismo posee precedencia en el proyecto.

1.9 Servidor Web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. Además los servidores pueden disponer de un intérprete de otros lenguajes de programación que ejecutan código embebido dentro del código HTML de las páginas que contiene el sitio antes de enviar el resultado al cliente. Esto se conoce como programación de lado del servidor y utiliza lenguajes como ASP, PHP, Perl y Ajax. Las

ventajas de utilizar estos lenguajes radica en la potencia de los mismos ejecutando tareas más complejas como, por ejemplo acceder a bases de datos abstrayendo al cliente de toda la operación. (20)

1.9.1 Servidor Web Apache

Apache es el servidor web hecho por excelencia, su nivel de configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es una tecnología gratuita de código fuente abierta. Es un servidor altamente configurable de diseño modular, siendo muy sencillo ampliar las capacidades del servidor. Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. También 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.

Apache es una muestra, al igual que el sistema operativo Linux (un Unix desarrollado inicialmente para PC), de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar.

La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia permite hacer lo que desee con el código fuente (incluso bifurcaciones y productos propietarios) siempre que sea reconocido su trabajo. (21)

Gracias a las ventajas y características positivas que posee el componente de servidor web se decide trabajar con Apache versión 2.2.

1.10 Gestor de Base de Datos

1.10.1 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, de los más reconocidos a nivel mundial ya que su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. Es un software multiplataforma, de fuente abierta (fuente abierta significa que es posible para cualquier persona usarlo y modificarlo, cualquier persona puede bajar el código fuente de MySQL en internet y usarlo sin pagar). Sin embargo, está mantenido por una empresa, MySQL AB, con sede en Suecia. El código fuente de MySQL está sólo relativamente abierto y disponible para modificaciones, puesto que es la empresa MySQL AB la que contrata y coordina los trabajos de mantenimiento del producto.

La evolución de MySQL se produce con las sugerencias de los usuarios, canalizadas por la empresa MySQL AB, que contrata a programadores de todo el mundo para que, a través de Internet, vayan ampliando y mejorando el producto. Las versiones, como es costumbre en este tipo de software libre, se van colgando en Internet para que los usuarios puedan disponer de ellas.

Las principales características de este gestor de bases de datos son:

1. Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
4. Gran portabilidad entre sistemas.
5. Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

(22)

Por todas las características antes mencionadas y porque 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.

1.11 Controlador de Versiones

El control de cambios es necesario para el desarrollo de cualquier tipo de proyectos de software. En la gran mayoría de los proyectos tanto de código abierto como privados es necesario trabajar en equipos en donde son varios los desarrolladores que acceden al mismo código fuente. Coordinar las modificaciones concurrentes al código de una forma manual es una tarea titánica, sobre todo en proyectos muy complejos o con una gran cantidad de desarrolladores. De aquí la necesidad de utilizar en el proyecto una herramienta que permita mezclar distintos cambios al código para consolidar un código único final de una forma automatizada.

1.11.1 Subversion

Subversion, también conocido con las siglas SVN, es un sistema de control de versiones que permite administrar los archivos de un proyecto, promoviendo con esto la colaboración entre los miembros de un equipo de trabajo y, de una manera muy cómoda, coordinar las tareas entre ellos.

Un repositorio, depósito o archivo es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos. Pues bien Subversion es un repositorio

en forma de árbol con una jerarquía de directorios y archivos que recuerda cada cambio que se haya realizado en el repositorio, cuándo se realizaron, y quién los realizó. Provee la habilidad de leer estados anteriores del sistema de archivos. (23)

Esto explica su utilización en la solución propuesta, independientemente de permitir el acceso a los archivos del proyecto, observar el trabajo, realizar cambios y guardar estos cambios en el repositorio, donde se almacenan todos los archivos involucrados, y que cualquier persona es libre de descargar, modificar, y redistribuir Subversion como desee. Para esta solución se decide trabajar con la versión 1.6.

1.12 Entorno de Desarrollo Integrado

Un IDE (Development Environment Integrated, del español Entorno Integrado de Desarrollo) es un sistema que facilita el trabajo del desarrollador de software, integrando sólidamente la edición orientada al lenguaje, la compilación o interpretación, la depuración, las medidas de rendimiento, la incorporación de las fuentes a un sistema de control de fuentes, normalmente de forma modular. (24)

1.12.1 Eclipse

Eclipse es un entorno de desarrollo integrado, de código abierto y multiplataforma. Mayoritariamente se utiliza para desarrollar lo que se conoce como “Aplicaciones de Cliente Enriquecido”, opuesto a las aplicaciones “Cliente liviano” basadas en navegadores. Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web.

Actualmente es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La licencia de Eclipse es EPL (Eclipse Public License), que no reúne todos los requisitos para ser compatible con la licencia GNU, aunque sus desarrolladores tienen la intención de adaptarlo en el futuro. (25)

Eclipse 3.5.2 fue desarrollado inicialmente por IBM y liberado por Eclipse Foundation en 2006, que se caracteriza por la posibilidad de extender la funcionalidad del entorno mediante la adición de plug-ins y el empleo de la interfaz gráfica de usuario JFace, que simplifica la construcción de aplicaciones basadas en SWT.

Otras características de Eclipse 3.5.2 son:

- Editor de texto con resaltado de sintaxis.
- Compilación en tiempo real.
- Posibilidad de emplear asistentes para la creación de proyectos, clases.
- Control de versiones con CVS. (26)

1.12.2 NetBeans

NetBeans IDE es un entorno de desarrollo, 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. Existe además un número importante de módulos para extender el NetBeans IDE. Lo más importante es que NetBeans IDE es un producto libre y gratuito sin restricciones de uso. (27)

Aunque Symfony fue ideado para trabajar desde la línea de comandos, a partir de la versión 6.8 del NetBeans se incluye el soporte a Symfony a fin de ayudar al desarrollador en la elaboración de aplicaciones complejas.

De esta integración se pueden sacar muchos provechos debido a las características completadas para el soporte de este marco para NetBeans 6.8. Algunas de esas características vienen dadas por el completado de código que ofrece a las variables de Symfony adecuadas en los archivos de sus vistas, los atajos de teclado asignables para acciones específicas de Symfony, la facilidad de navegación entre vistas y acciones, de ida y vuelta, y la posibilidad de ejecutar comandos de Symfony. (28) Por estas razones NetBeans 6.8 es el IDE seleccionado para trabajar en la solución.

1.15 Conclusiones Parciales

En este capítulo se abordaron los temas referentes a las tendencias de los reportadores dinámicos en la industria del software. Se realizó el estudio de varios reportadores dinámicos quedando seleccionado el GDR 1.7 como herramienta para la generación de reportes. También se realizó la selección de las herramientas para el desarrollo: Visual Paradigm v8.0 para el modelado de los datos, PHP v5.1 como lenguaje de programación, Symfony v1.0 como marco de desarrollo, Apache v2.2 para el servidor de aplicaciones, para la gestión de la base de datos MySQL v5.1, para el control de las versiones realizadas en la aplicación Subversion v1.6, NetBeans v6.8 como entorno de desarrollo, y la metodología a utilizar es RUP.

Capítulo 2: Análisis y diseño de la solución

Introducción

Para un mejor entendimiento estructural y funcional de un producto de software, se hace necesario su diseño desde ambas aristas. Por eso en este capítulo se abordarán los temas referentes a las reglas del negocio, a través de un modelo de dominio con su descripción pertinente. Se reflejarán, a partir de las necesidades detectadas del cliente, los requerimientos funcionales, así como los requerimientos no funcionales que de alguna manera mantendrán al usuario identificado y satisfecho con la aplicación. También se expondrán los diagramas que intervienen en el diseño de la solución: diagrama de casos de uso del sistema, diagrama de clases del diseño, diagrama de secuencia y diagrama de despliegue.

2.1. Objeto de estudio

2.1.1. Objetivos estratégicos de la organización

La Red Nacional de Genética está integrada por varios centros ubicados en todos los municipios y provincias del país, coordinados por el Centro Nacional de Genética Médica. Las investigaciones las realizan los especialistas ubicados en el área de Atención Primaria en los Policlínicos y Centros Municipales de Genética, los especialistas del área de Atención Secundaria en los Centros Provinciales de Genética y los especialistas ubicados en el Centro Nacional de Genética Médica. El objetivo fundamental es desarrollar proyectos de investigación e innovación, evaluar e introducir nuevas tecnologías para el diagnóstico, tratamiento y asesoramiento en relación con las enfermedades genéticas, todo esto encaminado a mejorar la calidad de vida de la población. (29)

2.1.2. Flujo actual de los procesos

El genetista ingresa al sistema alasMEDIGEN donde selecciona, desde la interfaz principal, el reporte con el que desea trabajar. Cuando el sistema muestra la interfaz del reporte seleccionado, el genetista establece los criterios de selección por los cuales desea realizar la búsqueda. Luego de efectuada una consulta a la base de datos se exponen los resultados obtenidos, a partir de los cuales el genetista realiza los diagnósticos o estudios pertinentes.

Actualmente para poder obtener algún reporte que no se encuentra disponible en la aplicación, es necesario crear directamente desde el gestor de base de datos las consultas correspondientes y realizarlos desde allí.

2.1.3. Análisis crítico de la ejecución de los procesos

El estudio de las enfermedades genéticas y defectos congénitos es bastante complejo en la actualidad. Toda la información arrojada por estos estudios es almacenada y gestionada en el CNGM desde alasMEDIGEN. Este sistema desde su creación ha ido desarrollando una serie de reportes que se encuentran distribuidos por módulos. Reporte por Prevalencia al Nacimiento, Reporte por Diagnóstico, Reporte por Atención, Reporte por Visitas, Reporte por Sexo, Reporte por Tipo de discapacidad, Reporte por Amparo Filial, Reporte por Capacidad Laboral, Reporte por Ocupación, Reporte por Vínculo Laboral, Reporte Estadístico y Reporte por Enfermedades, son los reportes existentes en la aplicación. Los mismos representan un bajo porcentaje de la información que los especialistas necesitan para poder realizar un análisis más detallado de los datos recogidos. Se caracterizan por ser estáticos, por tener muchas líneas de código, predefinidos para devolver datos muy superficiales, donde el usuario apenas introduce criterios de búsqueda y en otros casos obtiene muy poca información.

Dada la ausencia de determinados reportes en el sistema, surgen disímiles problemas para los genetistas a la hora de realizar un estudio o diagnóstico determinado, ya que se debe contar en el centro con la presencia de un especialista capaz de realizar las consultas necesarias a la base de datos. El mismo debe tener un amplio dominio de la estructura del sistema. Esto trae como consecuencia que no se obtengan los resultados esperados, demoras en el tiempo de respuesta, y que la visualización de los resultados no sea del mejor agrado, dificultando una correcta toma de decisiones.

2.2. Objeto de automatización

El nuevo módulo Reportes Genéticos de la aplicación informática alasMEDIGEN brindará una mayor cantidad de reportes, elimina el engorroso proceso de tener que trabajar directamente con el gestor de base de datos para obtener un reporte que no se encuentra en la aplicación. Esto permitirá la realización de estudios de Genética Poblacional a través de análisis estadísticos generando reportes que llegan a facilitar la toma de decisiones a partir de los resultados obtenidos.

2.3. Modelo de Dominio

Un Modelo de Dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase inicial, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. Su objetivo es capturar lo necesario para comprender donde va a funcionar el sistema que se está diseñando.

El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Esto es así ya que cuando se realiza la programación orientada a objetos, se supone que el funcionamiento interno del software va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de dominio constituye una primera versión del sistema. (30)

2.3.1. Diagrama de clases modelo de dominio

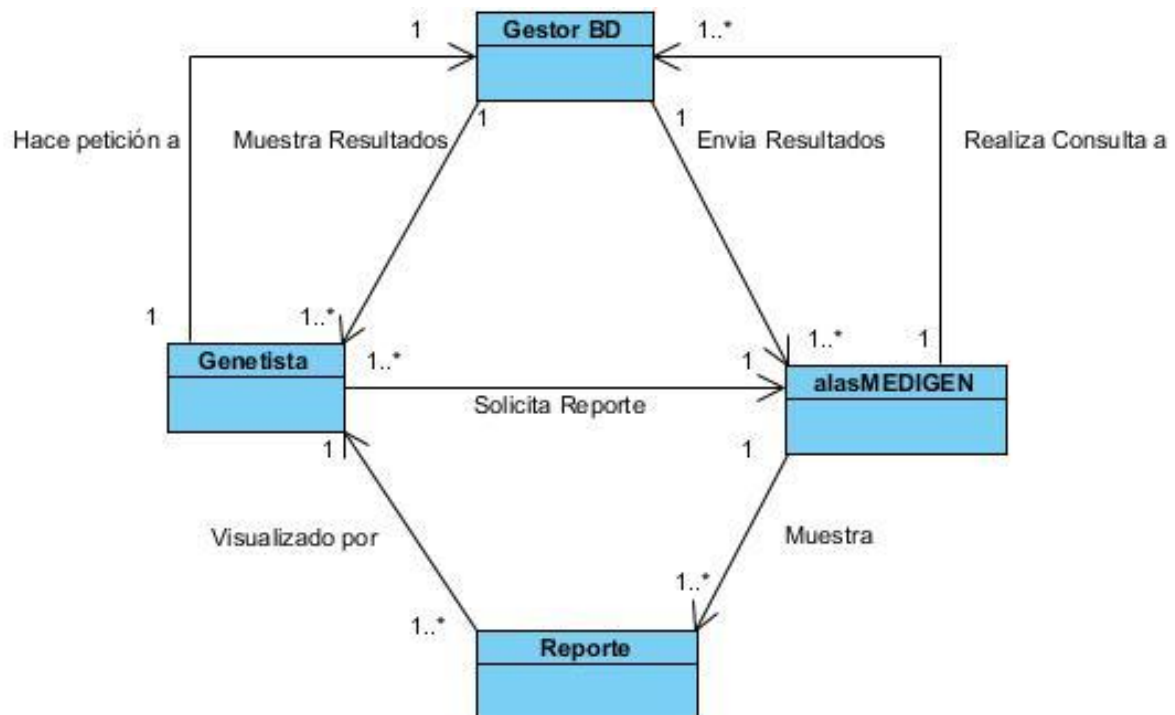


Figura 1. Diagrama de Clases del Modelo de Dominio

El genetista interactúa directamente con el sistema alasMEDIGEN seleccionando el reporte que desea analizar. Establece los criterios de búsqueda, se realiza una consulta a la base de datos y luego le son mostrados los resultados al genetista para que realice las acciones estimadas. En caso de que se desee trabajar con un reporte determinado y el mismo no se encuentre en el sistema, se debe recurrir a su creación directamente desde el gestor de base de datos.

2.3.2. Descripción de las clases del modelo de dominio

Clases	Descripción
Genetista	Especialista que interactúa con el sistema solicitando un reporte determinado.

alasMEDIGEN	Sistema Informático que utiliza el GDR para visualizar un reporte según el criterio de búsqueda establecido por el genetista.
Reporte	Documento que muestra los resultados según criterios de búsqueda.
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

2.3.3. Reglas del negocio

- Más de un paciente puede tener el mismo diagnóstico.
- Un paciente puede presentar más de una enfermedad.
- El Genetista Municipal solo tiene acceso a la información de los estudios realizados en su municipio.
- El Genetista Provincial tiene acceso a la información de los estudios realizados en su provincia y en su municipio.
- El Genetista Nacional tiene acceso a la información de los estudios realizados en todo el país.

2.4. Especificación de los requerimientos de la aplicación informática

2.4.1. Requerimientos funcionales

Los requerimientos funcionales son condiciones o capacidades que deben ser alcanzadas o poseídas por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Con ellos se pretende determinar de manera clara y concisa lo que debe hacer el sistema siguiendo un enfoque funcional. (31)

RF1 Generar Reporte de Discapacidad Intelectual por Diagnóstico por:

- Diagnóstico discapacidad intelectual
- Provincia
- Municipio

- Consejo popular

RF2 Generar Reporte de Discapacidad Intelectual por Atención por:

- Servicio de atención de salud
- Provincia
- Municipio
- Consejo popular

RF3 Generar Reporte de Discapacidad Intelectual por Visitas por:

- Servicio de atención de salud

RF4 Generar Reporte de Discapacidad Intelectual por Total de Discapacitados Intelectual

- Apto para trabajar
- Sexo
- Rango de Edades
- Dependencia
- Evaluación Funcional
- Nivel (provincial o nacional)

RF5 Generar Reporte de Discapacidad Intelectual por Total de Discapacitados que Necesitan Ayuda

- Sexo
- Rango de Edades
- Tipo de Ayuda
- Nivel (provincial o nacional)

RF6 Generar Reporte de Discapacidad Intelectual por Total de Huérfanos

- Sexo
- Provincia
- Municipio
- Rango de Edades

RF7 Generar Reporte de Discapacidad Física por Sexo por:

- Sexo
- Nivel (provincial o nacional)

RF8 Generar Reporte de Discapacidad Física por Tipo de Discapacidad por:

- Tipo de discapacidad
- Nivel (provincial o municipal)

RF9 Generar Reporte de Discapacidad Física por Amparo Filial por:

- Amparo filial
- Nivel (provincial o municipal)

RF10 Generar Reporte de Discapacidad Física por Capacidad Laboral por:

- Capacidad laboral
- Nivel (provincial o municipal)

RF11 Generar Reporte de Discapacidad Física por Ocupación por:

- Ocupación
- Nivel (provincial o municipal)

RF12 Generar Reporte de Discapacidad Física por Vínculo Laboral por:

- Vínculo laboral
- Nivel (provincial o municipal)

RF13 Generar Reporte de Discapacidad Física por Otras Discapacidades

- Apto para trabajar
- Sexo
- Tipo de Discapacidad
- Rango de Edades
- Dependencia
- Nivel (provincial o municipal)

RF14 Generar Reporte de Discapacidad Física por Necesidad de Ayuda de Otras Discapacidades

- Apto para trabajar
- Sexo
- Tipo de Discapacidad
- Rango de Edades
- Dependencia
- Nivel (provincial o municipal)

RF15 Generar Reporte de Discapacidad Física por Total de Discapacitados Asociados

- Consume Drogas
- Incorporado Asociación
- Desea Asociarse (en caso de que no esté incorporado en ninguna)
- Sexo
- Rango de Edades
- Nivel (provincial o municipal)

RF16 Generar Reporte de Discapacidad Física por Total de Ambas Discapacidades

- Encamado
- Sexo
- Rango de Edades
- Evaluación Funcional
- Nivel (provincial o municipal)

RF17 Generar Reporte de Discapacidad Física por Protección Familiar

- Protección Familiar
- Escolarizado
- Sexo
- Rango de Edades
- Nivel (provincial o municipal)

RF18 Generar Reporte de Discapacidad Física por Condiciones de Vida

- Consume Drogas
- Sexo
- Rango de Edades
- Necesita
- Ingresos
- Nivel

RF19 Generar Reporte de Discapacidad Física por Total de Necesitados de una Discapacidad

- Sexo
- Tipo de Discapacidad
- Rango de Edades
- Nivel (provincial o municipal)

RF20 Reporte de Discapacidad Física Total por Tipo de Discapacidad

- Sexo
- Tipo de Discapacidad
- Rango de Edades
- Nivel (provincial o municipal)

RF21 Generar Reporte de Enfermedades Genéticas por Enfermedades por:

- Provincial
- Municipio

- Clasificación de la enfermedad
- Enfermedad
- Sexo
- Rango de edades

RF22 Generar Reporte de Gemelos por Total de Gemelos por:

- Sexo
- Padecimiento
- Adoptado
- Complicaciones
- Nivel (provincial o municipal)

RF23 Generar Reporte de Gemelos por Total de Parejas de Gemelos por:

- Gemelos Diczigóticos
- Padecimiento
- Ambiente
- Nivel (provincial o municipal)

RF24 Generar Reporte de Gemelos por Factores de Riesgo por:

- Padecimientos
- Factores de Riesgo
- Provincia

RF25 Generar Reporte de Gemelos por la Funcionalidad de la Familia por:

- Relaciones en Casa
- Padecimientos
- Nivel (provincial o municipal)

RF26 Generar Reporte de Gemelos por Antecedentes Familiares y Enfermedades Mentales en Gemelos por:

- Familiar Hospitalizado Antes
- Trastornos
- Nivel (provincial o municipal)

RF27 Generar Reporte de Gemelos por Antecedentes Familiares y Enfermedades Mentales en Parejas de Gemelos por:

- Familiar Hospitalizado Antes
- Trastornos

- Padecimientos
- Nivel (provincial o municipal)

RF28 Generar Reporte de Anomalías Cromosómicas por Anomalías por:

- Rango de edades
- Sexo
- Tipo de muestra
- Motivo de Indicación
- Código de Resultado
- Es variante o Anomalía

2.4.2. Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, rápido o confiable. Normalmente están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. (31)

- Apariencia o interfaz externa:

Se deben utilizar imágenes y colores similares al sistema alasMEDIGEN v 1.0 (32)

- Usabilidad:

La aplicación informática debe garantizar un acceso fácil y rápido, contando con un menú que satisfaga las necesidades de los usuarios. (32)

- Rendimiento:

Los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información. (32)

- Soporte:

Se debe asegurar el soporte para los usuarios de manera que se puedan satisfacer sus necesidades a partir de mejoras, una vez puesta en marcha la aplicación. (32)

- Seguridad:

El sistema debe tener 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 municipalmente. Sus

siglas significan seguridad, autenticación, autorización y auditoría 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.

- Software:

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

- Hardware:

Para el desarrollo y ejecución de la aplicación se necesitará:

Para el servidor de aplicación:

- Microprocesador Pentium IV a 3 GHz o superior.
- 1Gb de RAM o superior.
- Mínimo 5 Gb 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 o superior.
- Conexión al servidor.

- Disponibilidad:

Se debe garantizar el funcionamiento de la aplicación durante las 24 horas del día y los siete días de la semana, con el menor tiempo posible de recuperación de fallos. (32)

- Requisitos Legales:

Las herramientas y las tecnologías en que estará basada la aplicación informática deberán cumplir con las licencias de software libre. (32)

- 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. (32)

2.5 Definición de los Casos de Uso del Sistema

Los diagramas de caso de uso son uno de los cinco tipos de diagramas en UML para modelar aspectos dinámicos de sistemas. Son importantes para visualizar, especificar, y documentar el comportamiento de un elemento.

Un diagrama de casos de uso del sistema permite modelar los requerimientos de un sistema, que no es más que especificar qué es lo que hará el sistema, independientemente de cómo el sistema lo hará.

(33)

2.5.1 Actores del Sistema

Actor	Descripción
Genetista	Especialista en Genética que interactúa con el sistema, es el encargado de realizar los estudios y cálculos estadísticos sobre epidemiología.

Tabla 2. Actores del Sistema

2.5.2 Diagrama de Casos de Uso del Sistema

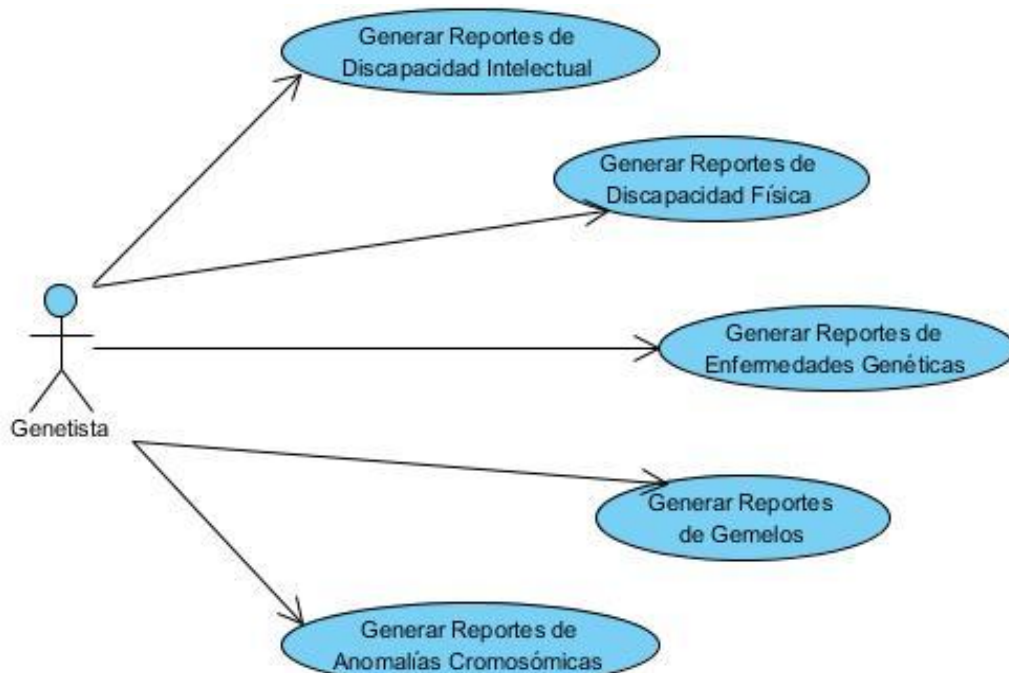


Figura 2. Diagrama de Caso de Usos del Sistema. Reportes Genéticos.

2.5.3 Descripción a Alto Nivel de los Casos de Uso del Sistema

Caso de Uso:	Generar Reporte de Discapacidad Intelectual
Actores:	Genetista (inicia)
Resumen:	El caso de uso se inicia cuando el genetista solicita algún reporte de discapacidad intelectual. El sistema le muestra la interfaz correspondiente para introducir un criterio de búsqueda. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
Precondiciones:	Estar autenticado
Referencias	RF1, RF2, RF3, RF4, RF5, RF6
Prioridad	

Tabla 3. Descripción del Caso de Uso del Sistema Generar Reporte de Discapacidad Intelectual

Caso de Uso:	Generar Reportes por Discapacidad Física
Actores:	Genetista (inicia)
Resumen:	El caso de uso se inicia cuando el genetista solicita algún reporte de discapacidad física. El sistema le muestra la interfaz correspondiente para introducir un criterio de búsqueda. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
Precondiciones:	Estar autenticado
Referencias:	RF7, RF8, RF9, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20
Prioridad:	

Tabla 4. Descripción del Caso de Uso del Sistema Generar Reporte de Discapacidad Física

Caso de Uso:	Generar Reportes por Enfermedades Genéticas
Actores:	Genetista (inicia)
Resumen:	El caso de uso se inicia cuando el genetista solicita algún reporte de enfermedades genéticas. El sistema le muestra la interfaz correspondiente para introducir un criterio de búsqueda. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
Precondiciones:	Estar autenticado

Referencia:	RF21
Prioridad:	

Tabla 5. Descripción del Caso de Uso del Sistema Generar Reporte de Enfermedades Genéticas

Caso de Uso:	Generar Reportes de Gemelos
Actores:	Genetista (inicia)
Resumen:	El caso de uso se inicia cuando el genetista solicita algún reporte de gemelos. El sistema le muestra la interfaz correspondiente para introducir un criterio de búsqueda. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
Precondiciones:	Estar autenticado
Referencia:	RF22, RF23, RF24, RF25, RF26, RF27
Prioridad:	

Tabla 6. Descripción del Caso de Uso del Sistema Generar Reporte de Gemelos

Caso de Uso:	Generar Reportes de Anomalías Cromosómicas
Actores:	Genetista (inicia)
Resumen:	El caso de uso se inicia cuando el genetista solicita algún reporte de anomalías cromosómicas. El sistema le muestra la interfaz correspondiente para introducir un criterio de búsqueda. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
Precondiciones:	Estar autenticado
Referencia:	RF28
Prioridad:	

Tabla 7. Descripción del Caso de Uso del Sistema Generar Reporte de Anomalías Cromosómicas

2.5.3 Descripción del Caso de Uso Extendido Generar Reportes de Discapacidad Intelectual

Caso de Uso:	Generar Reporte de Discapacidad Intelectual
Actores:	Genetista (inicia)
Resumen:	El caso de uso se inicia cuando el genetista solicita algún reporte de discapacidad intelectual. El sistema le muestra la interfaz correspondiente para introducir un criterio de búsqueda. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
Precondiciones:	Estar autenticado
Referencias	RF1, RF2, RF3, RF4, RF5, RF6
Prioridad	

Prototipo de Interfaz



Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El genetista escoge la opción Reporte de Discapacidad Intelectual y puede	2. El sistema muestra la interfaz correspondiente para seleccionar los criterios de búsqueda.

<p>visualizar:</p> <ul style="list-style-type: none"> a) Reporte por Diagnóstico (Ver sección Reporte por Diagnóstico) b) Reporte por Atención (Ver sección Reporte por Atención) c) Reporte por Visitas (Ver sección Reporte por Visitas) d) Reporte por Total de Discapacitados Intelectual (Ver sección Reporte por Total de Discapacitados Intelectual) e) Reporte por Total de Discapacitados que Necesitan Ayuda (Ver sección Reporte por Total de Discapacitados que Necesitan Ayuda) f) Reporte por Total de Huérfanos (Ver sección Reporte por Total de Huérfanos) 	
Sección “Reporte por Diagnóstico”	
Acción del Actor	Respuesta del Sistema
<p>1. El genetista escoge la opción: Reporte por Diagnóstico.</p>	<p>2. El sistema muestra una interfaz con los criterios de búsqueda para generar el reporte (retraso mental, provincia, municipio, consejo popular).</p>
<p>3. El genetista especifica los criterios por las cuales va a realizar el reporte.</p>	<p>4. El sistema muestra el reporte según los criterios entrados por el usuario.</p>
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>2.1 En caso de que no se especifique un campo obligatorio se muestra un mensaje indicando que el campo es de entrada obligatoria.</p>
Prototipo de Interfaz	

Sección “Reporte por Atención”	
Acción del Actor	Respuesta del Sistema
1. El genetista escoge la opción: Reporte por Atención.	2. El sistema muestra una interfaz con los criterios de búsqueda para generar el reporte (servicio, provincia, municipio, consejo popular).
3. El genetista especifica los criterios por las cuales va a realizar el reporte.	4. El sistema muestra el reporte según los criterios entrados por el usuario.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no se especifique un campo obligatorio se muestra un mensaje indicando que el campo es de entrada obligatoria.
Prototipo de Interfaz	
Sección “Reporte por Visitas”	
Acción del Actor	Respuesta del Sistema
1. El genetista escoge la opción: Reporte por Visitas.	2. El sistema muestra una interfaz con los criterios de búsqueda para generar el reporte (servicio).
3. El genetista especifica los criterios por las cuales va a realizar el reporte.	4. El sistema muestra el reporte según los criterios entrados por el usuario.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no se especifique un campo obligatorio se muestra un mensaje indicando que el campo es de entrada obligatoria.
Prototipo de Interfaz	
Sección “Reporte por Total de Discapitados Intelectual”	
Acción del Actor	Respuesta del Sistema
1. El genetista escoge la opción: Reporte por Total de Discapitados Intelectual.	2. El sistema muestra una interfaz con los criterios de búsqueda para generar el reporte (Apto para trabajar, sexo, provincia, municipio, rango de

	edades, dependencia).
3. El genetista especifica los criterios por las cuales va a realizar el reporte.	4. El sistema muestra el reporte según los criterios entrados por el usuario.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no se especifique un campo obligatorio se muestra un mensaje indicando que el campo es de entrada obligatoria.
Prototipo de Interfaz	
Sección “Reporte por Total de Discapacitados que Necesitan Ayuda”	
Acción del Actor	Respuesta del Sistema
1. El genetista escoge la opción: Reporte por Total de Discapacitados que Necesitan Ayuda.	2. El sistema muestra una interfaz con los criterios de búsqueda para generar el reporte (sexo, provincia, municipio, rango de edades, tipo de ayuda).
3. El genetista especifica los criterios por las cuales va a realizar el reporte.	4. El sistema muestra el reporte según los criterios entrados por el usuario.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no se especifique un campo obligatorio se muestra un mensaje indicando que el campo es de entrada obligatoria.
Prototipo de Interfaz	
Sección “Reporte por Total de Huérfanos”	
Acción del Actor	Respuesta del Sistema
1. El genetista escoge la opción: Reporte por Total de Huérfanos.	2. El sistema muestra una interfaz con los criterios de búsqueda para generar el reporte (sexo, provincia, municipio, rango de edades).
3. El genetista especifica los criterios por las cuales va a realizar el reporte.	4. El sistema muestra el reporte según los criterios entrados por el usuario.

Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no se especifique un campo obligatorio se muestra un mensaje indicando que el campo es de entrada obligatoria.
Prototipo de Interfaz	

Tabla 8. Descripción del Caso de Uso Extendido Generar Reporte de Discapacidad Intelectual

Para poder apreciar la descripción extendida del resto de los casos de uso del sistema, se recomienda la revisión del documento Modelo del Sistema que se encuentra en la carpeta de requisitos del Expediente de Proyecto.

1.13 Patrones de Diseño

Un patrón de diseño es: (34)

- una solución estándar para un problema común de programación.
- una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- un proyecto o estructura de implementación que logra una finalidad determinada.
- un lenguaje de programación de alto nivel.
- una manera más práctica de describir ciertos aspectos de la organización de un programa.
- conexiones entre componentes de programas.
- la forma de un diagrama de objeto o de un modelo de objeto.

Las principales características de un patrón de diseño:

- Son soluciones concretas (proponen soluciones a problemas concretos, no son teorías genéricas).
- Son soluciones técnicas (indican soluciones técnicas basadas en Programación Orientada a Objetos (POO), en ocasiones tienen más utilidad con algunos lenguajes de programación y en otras son aplicables a cualquier lenguaje).
- Se utilizan en situaciones frecuentes (ya que se basan en la experiencia acumulada de resolver problemas reiterativos).

- Favorecen la reutilización de código que ayudan a construir software basado en la reutilización (a construir clases reutilizables), los propios patrones se reutilizan cada vez que se vuelven a aplicar.
- Es difícil reutilizar la implementación de un patrón (al aplicar un patrón aparecen clases concretas que solucionan un problema concreto y que no será aplicable a otros problemas que requieran el mismo patrón). (34)

1.13.1 Patrones GRASP

Los patrones GRASP son parejas de problema-solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (34)

Patrón Experto: Este patrón se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. (34)

Patrón Alta Cohesión: Mantiene la complejidad dentro de límites manejables, es decir asigna una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Beneficios:

1. Mejoran la claridad y facilidad con que se entiende el diseño.
2. Se simplifica el mantenimiento y las mejoras de funcionalidad.
3. A menudo se genera un bajo acoplamiento.
4. Soporta mayor capacidad de reutilización. (34)

Patrón Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que permita conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. Lo que define este patrón es que una instancia de un objeto la tiene que crear el objeto que tiene la información para ello. ¿Qué significa esto?, pues que si un objeto A utiliza específicamente otro B, o si B forma parte de A, o si A almacena o contiene B, o si simplemente A tiene la información necesaria para crear B, entonces A es el perfecto creador de B. (34)

Patrón Controlador: Es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Asigna las responsabilidades de capturar los eventos del sistema a las clases.

De acuerdo con el patrón Controlador, se dispone de las siguientes opciones:

1. El “sistema” global (controlador de fachada).
2. La empresa u organización global (controlador de fachada).
3. Algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).
4. Un manejador artificial de todos los eventos del sistema de un caso de uso, generalmente denominados “Manejador<NombreCasodeUso>” (controlador de casos de uso).
5. En la decisión de cuál de las cuatro clases es el controlador más apropiado influyen también otros factores como la cohesión y el acoplamiento. (34)

Patrón Bajo Acoplamiento: Es la idea de tener las clases 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. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases.

Beneficios:

1. No se afectan por cambios de otros componentes.
2. Fáciles de entender por separado.
3. Fáciles de reutilizar. (34)

2.13.2 Patrones de diseño GOF

Los patrones GOF constituyen patrones de diseño surgidos en 1995 por Erich Gamma, Richard Helm, Ralph Jonson y John Vissidess, promueven una expansión de la programación orientada a objetos y se pueden clasificar según su propósito en patrones de creación (para creación de instancias), estructurales (relaciones entre clases, combinación y formación de estructuras mayores) y de comportamiento (interacción y cooperación entre clases). (35) Algunos de estos patrones son:

Patrones de creación: Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que

clases instanciar o sobre que objetos un objeto delegará responsabilidades. Uno de los tipos de patrones de creación que utiliza Symfony es el Singleton (Instancia única). (35)

Patrones estructurales: Describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. En la categoría de patrones estructurales, Symfony utiliza el patrón Decorator (Envoltorio). (35)

Patrones de comportamiento: Se utilizan para organizar, manejar y combinar comportamientos. En la categoría de patrones de comportamiento, Symfony utiliza el patrón Command (Acción). (35)

1.14 Patrones de Arquitectura

Los patrones arquitectónicos no son más que un patrón de alto nivel que fija la arquitectura global de una aplicación. Se definen como una descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. Los patrones arquitectónicos proveen un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen reglas y pautas para la organización de las relaciones entre ellos. (36) Dentro de los diversos patrones de arquitectura existentes, cabe destacar los más significativos: el patrón de arquitectura en capas y el patrón Modelo Vista Controlador (MVC).

1.14.2 Patrón Modelo Vista Controlador (MVC)

Para el desarrollo de la aplicación se hará uso del patrón Modelo-Vista-Controlador. Esta decisión viene dada no solo porque Symfony está basado en este clásico patrón de diseño web, sino también por el uso del GDR y alasMEDIGEN que implementan este patrón. El mismo está formado por tres niveles y la utiliza de la siguiente forma:

El modelo: Solo se encarga del acceso a los datos almacenados en el gestor de base de datos. Ha sido dividido en dos capas, la capa de acceso a los datos y en la capa de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no 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 bases de datos, solamente es necesario actualizar la capa de abstracción de la base de datos. (37)

La vista: Las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabeceras de la página, el layout (esquema) genérico, el pie de página y la

navegación global. En la mayor parte de las veces sólo cambia el interior de la página. Por este motivo, la vista se separa en un layout y en una plantilla. Normalmente, el layout es global en toda la aplicación o al menos en un grupo de páginas. La plantilla sólo se encarga de visualizar las variables definidas en el controlador. Para que estos componentes interactúen entre sí correctamente, es necesario añadir cierto código, código que será añadido a través de la lógica de la vista. (37)

El controlador: Una parte importante de su trabajo es común a todos los controladores de la aplicación. Entre las tareas comunes se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador se ha dividido en un controlador frontal, que se encarga de realizar las tareas comunes y las acciones, que incluyen el código específico del controlador de cada página. (37)

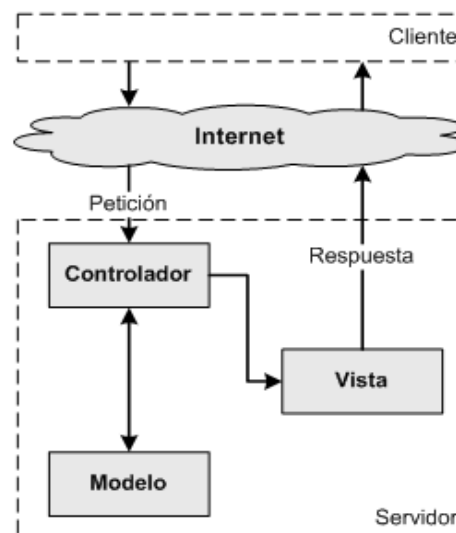


Figura 3. Patrón de Arquitectura de Symfony: MVC.

2.6 Diagramas de clases del diseño

Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación. Incluye información referente a:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Navegabilidad.
- Dependencias. (38)

2.6.1 Vista Lógica de la Arquitectura

A continuación se presenta la vista lógica de la arquitectura del sistema, en el cual se puede observar la aplicación del estilo arquitectónico MVC generado por Symfony. También se puede apreciar la relación existente entre los paquetes de diseño y cada uno de los subsistemas utilizados para un mejor desarrollo de la aplicación. Estos son el subsistema *Propel* que genera la capa de acceso a datos, el subsistema *Symfony* que contiene todos los componentes que éste brinda para su uso, el subsistema *ComponentesSISalud* que representa el servicio de salud con el cual interactúa la aplicación para establecer privilegios de acceso y el subsistema *GDR* que es el encargado de la generación y creación de los reportes que serán mostrados al usuario final.

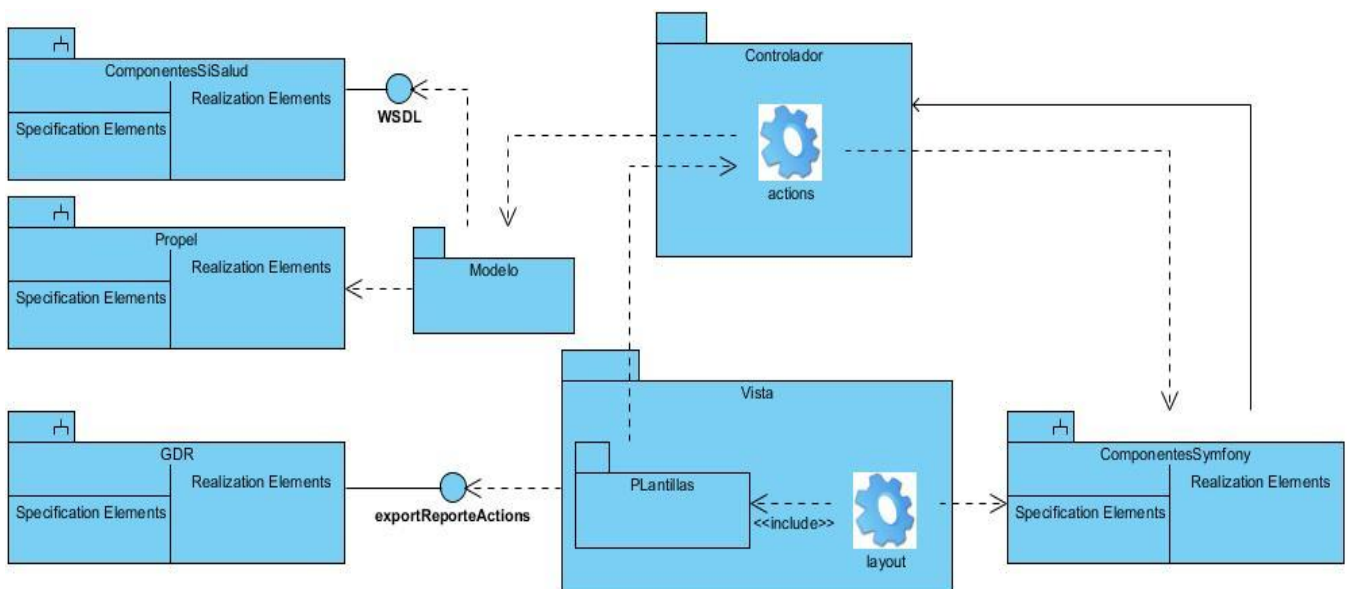


Figura 4. Vista Lógica. Módulo de Reportes Genéticos.

2.6.2 Diagrama de Clases de Diseño CU Generar Reportes de Discapacidad Intelectual

La siguiente figura muestra el diagrama de clases de diseño correspondiente al caso de uso Generar Reportes de Discapacidad Intelectual. Para el resto de los casos el diseño sigue la misma estructura.

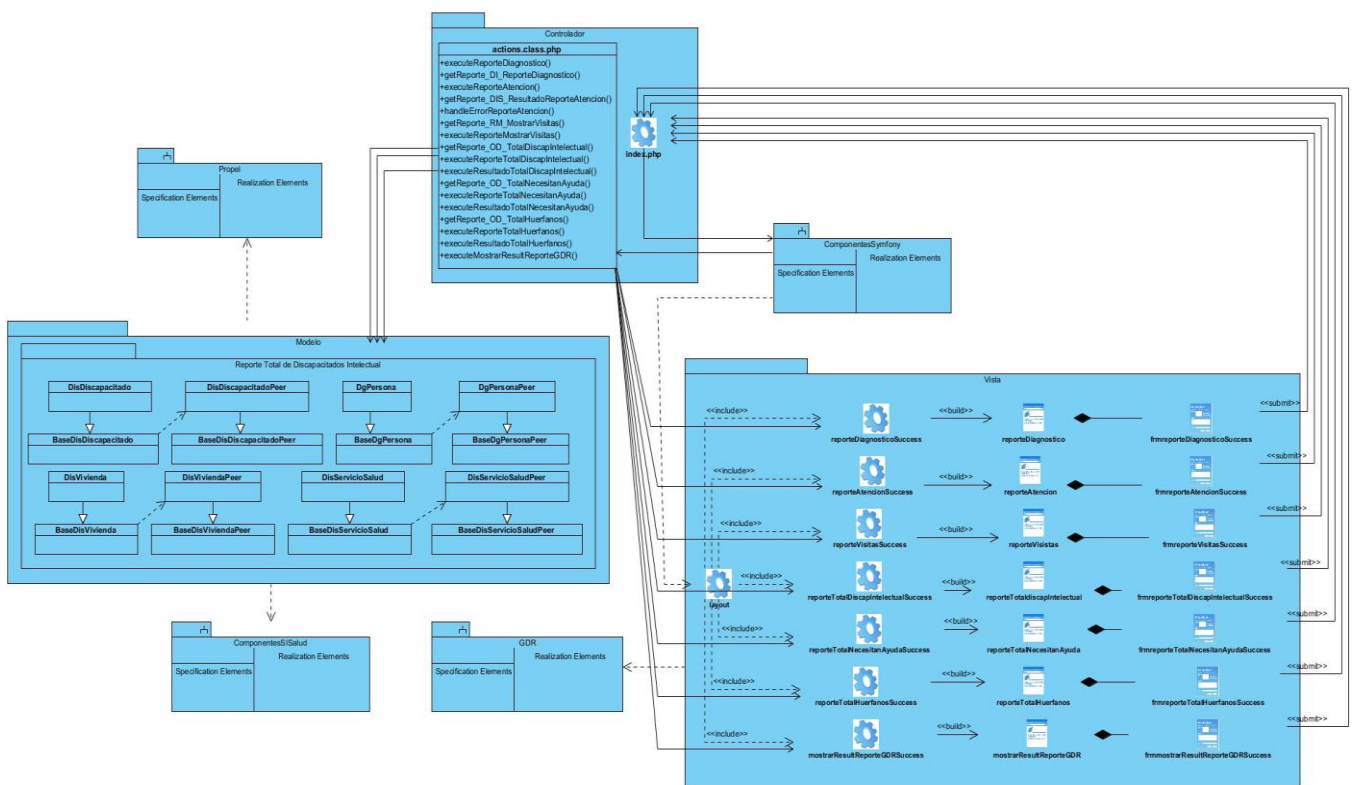


Figura 5. Diagrama de Clases del Diseño. Generar Reportes de Discapacidad Intelectual.

En el mismo se usan paquetes de diseño para distribuir y mostrar cuáles clases tienen cada responsabilidad del patrón Modelo-Vista-Controlador.

El paquete de diseño Vista se agrupan todas las clases de la presentación como las páginas servidoras, siendo estas las encargadas de construir las páginas clientes. Estas páginas a su vez están compuestas por un formulario. Este diagrama es representado a partir de estereotipos web.

En el paquete de diseño Modelo se encuentra la capa de acceso a datos. Este encapsula las cuatro clases fundamentales que genera Symfony por cada tabla de la base de datos y la relación que existe entre cada una de ellas.

En el paquete de diseño Controlador están el controlador frontal *index.php* que se encarga de redireccionar todas las peticiones que se hacen desde de la vista, y la clase *actions* que controla todo el proceso. Además, este controlador es el encargado de establecer la comunicación entre la vista y el modelo.

También se observan los cuatro subsistemas utilizados para el desarrollo del módulo y su relación con

cada uno de los paquetes de diseño.

2.6.3 Aplicación de Patrones de Diseño

GRASP (Experto): Este patrón se emplea al trabajar con el marco de trabajo Symfony y un ejemplo de ello es la inclusión de Propel para mapear la base de datos. Propel es el encargado de generar las clases para la gestión de las tablas en dicha base de datos con las responsabilidades debidamente asignadas. Cada clase cuenta con un grupo de funcionalidades que las convierte en experta de la información de la tabla a la que representa.

GRASP (Alta Cohesión): Symfony presenta entre sus principales características una organización bien definida en cuanto a estructura y responsabilidades. Esto permite que se trabaje con las clases con una alta cohesión. Un ejemplo de esto es el controlador, el cual delega en sus componentes funciones para el manejo de los eventos del sistema estrechamente relacionados entre sí y donde la clase *actions* solo se encarga de definir las acciones a realizar, así las tareas son realizadas con mayor eficiencia.

GRASP (Creador): La clase *actions* es la que contiene las acciones y es la encargada de ejecutarlas. Esta tiene todas las funciones definidas para el módulo de Reportes Genéticos. En dichas acciones se crean objetos de las clases que representan las entidades. Esto permite evidenciar que la clase *actions* es “creador” de estas entidades.

GRASP (Bajo Acoplamiento): Este patrón se evidencia en la aplicación ya que el sistema presenta poca dependencia entre las clases. Es en el modelo donde únicamente se encuentran algunas relaciones de asociación entre las clases, pero no representa una gran jerarquía.

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

2.6.4 Integración del GDR con alasMEDIGEN

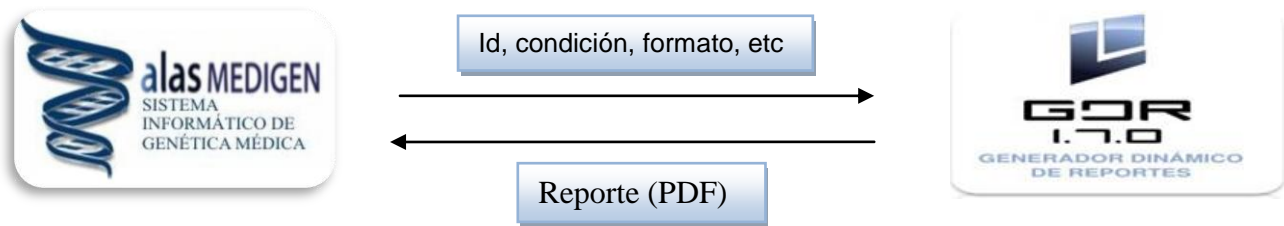


Figura 6. Integración de alasMEDIGEN con GDR

El sistema alasMEDIGEN solicita al GDR un reporte que está previamente diseñado en el mismo. El sistema junto con la petición envía criterios de búsqueda para filtrar los datos de dichos reportes, que lo harán variar en la cantidad de respuestas que devuelve al usuario. El sistema también envía al GDR un valor que es conocido como id o identificador, este valor es único en cada reporte, y su papel es distinguir en la base de datos del GDR el reporte que el sistema está solicitando. El GDR posee una API que vincula al sistema alasMEDIGEN con dicho reporteador, y gracias a esta API el sistema le envía al GDR las condiciones o criterios de búsquedas para filtrar los datos en el reporte con el id solicitado en la petición.

El GDR recibe lo enviado a través de la API y busca en su base de datos el reporte que tenga el id que viene en la petición del sistema alasMEDIGEN. Encontrado el reporte, se ejecuta la consulta a la base de datos por la que fue diseñado el mismo, añadiéndole a esta consulta los criterios de búsqueda que vienen en la petición hecha por el sistema mediante los filtros que posee cada reporte. Al concluir el proceso, los resultados se visualizan en el reporte de la manera que este tiene establecido para mostrar los datos y ese resultado final se le envía al sistema alasMEDIGEN para que muestre dicho reporte al usuario que lo está solicitando. Este reporte finalmente puede ser almacenado por el sistema en formato pdf.

2.7 Diagramas de Interacción

Los diagramas de interacción muestran la relación que se establece mediante el paso de mensajes entre los distintos objetos que participan en un escenario. Modelan también el comportamiento dinámico del sistema, es decir, el flujo de control en una operación. Describen secuencia de acciones o colaboración entre objetos y son representados por diagramas de secuencia o de colaboración. (39)

Diagramas de secuencia: Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar. (39)

2.8 Modelo de despliegue

El Diagrama de Despliegue representa la configuración de los nodos físicos sobre los cuales se despliega el módulo como parte de alasMEDIGEN. La forma en la que se distribuyen los servidores está en correspondencia con la definida para el sistema completo y se explica a continuación.

La siguiente figura muestra el diagrama de despliegue correspondiente al módulo Reportes Genéticos del sistema alasMEDIGEN.

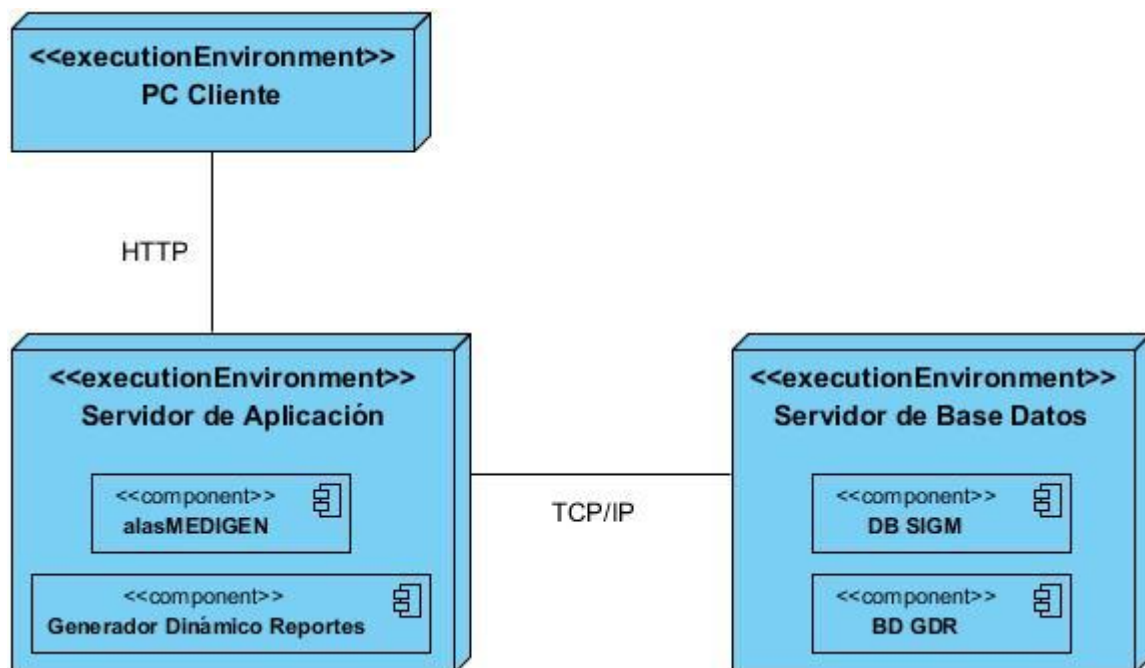


Figura 8. Diagrama de la Vista de Despliegue.

En este caso se necesita un servidor web como entorno de ejecución con Apache2, soporte para PHP 5 y sistema operativo basado en GNU/Linux en cualquiera de sus distribuciones. Se hace necesario contar también con un servidor de base de datos MySQL 5.0 o versiones superiores. Los usuarios del sistema deberán contar con un navegador Internet Explorer 7.0, Mozilla Firefox 2.0 o de las distribuciones de Linux, para poder acceder a las opciones que brinda el sistema.

Servidor Web: En este nodo se encuentra las dos aplicaciones que dan vida al módulo Reportes Genéticos: alasMEDIGEN y el Generador Dinámico de Reportes.

Servidor de Base de Datos: Es donde se encuentran almacenadas las bases de datos de alasMEDIGEN y el Sistema Generador Dinámico de Reportes.

PC Cliente: Desde este nodo se accede al sistema por medio de un navegador.

2.9 Conclusiones Parciales

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 en el mismo, dando la posibilidad de entender mejor la lógica de los procesos. Atendiendo a las necesidades del cliente, se definieron 28 requisitos funcionales que permitieron guiar el desarrollo de la solución con respecto a las funciones que debe cumplir el sistema. Se elaboró el diagrama de casos de uso del sistema con el fin de visualizar y especificar el comportamiento de cada elemento del sistema, y se realizó la descripción detallada de un caso de uso. Posteriormente se procedió a 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.

Capítulo 3: Implementación y pruebas de la solución.

Introducción

Antes de materializar las ideas que se tienen para el desarrollo de un producto de software, es precisa la comprensión de la interacción entre los componentes que van a intervenir en la solución final. Por tal motivo en este capítulo se analizará el modelo de implementación así como el diagrama de componentes. Además, se dará una explicación de las pruebas realizadas para poder validar el buen funcionamiento de la aplicación.

3.1 Implementación

En la implementación a partir de los resultados del análisis y diseño se implementa el sistema desde el punto de vista de componentes, es decir, ficheros de código fuente, archivos de procesamientos por lotes (scripts), ejecutables y similares. El objetivo fundamental es representar las partes físicas que componen el sistema, organizado de una manera que sea comprensible y manejable.

3.1.1 Diagrama de Componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes, etc.) y muestra las dependencias entre estos componentes. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. (40)

El siguiente diagrama de componentes corresponde al caso de uso Generar Reporte de Discapacidad Intelectual. En el mismo se muestran los componentes más significativos implementados y la relación existente entre ellos. La mayoría de estos componentes son archivos .php. 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.

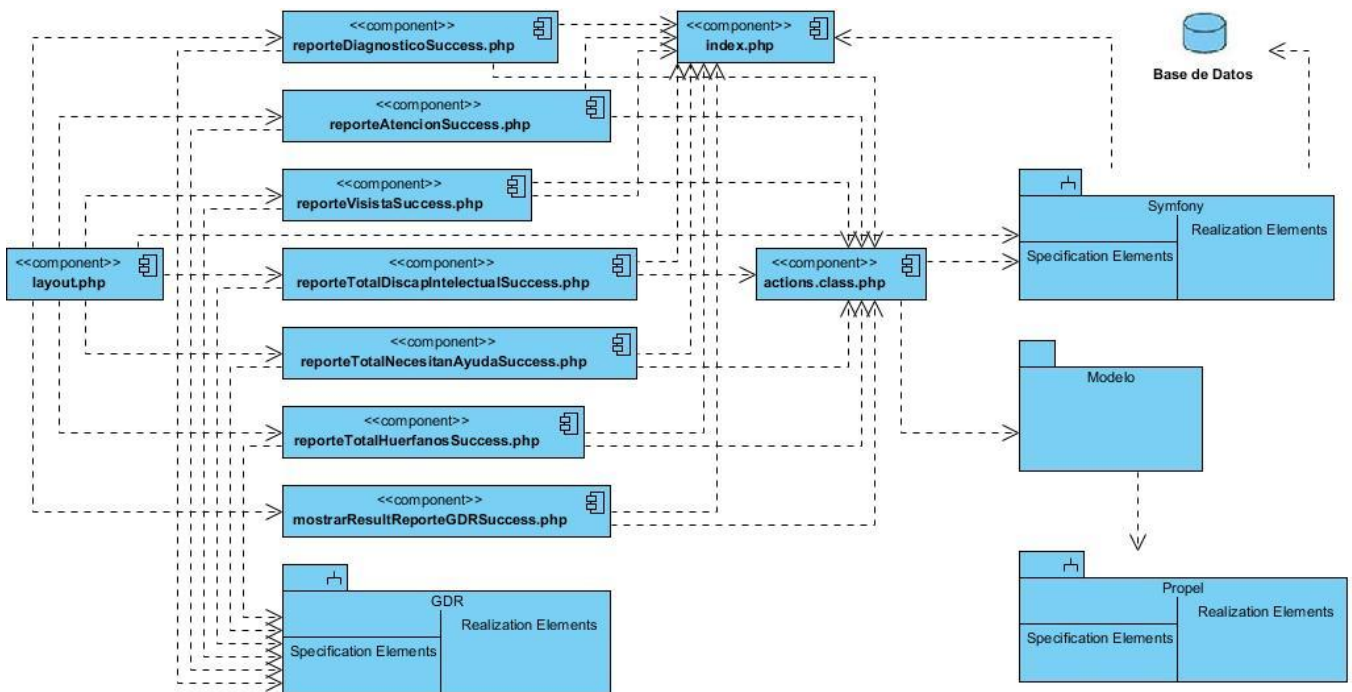


Figura 9. Diagrama de Componentes. Generar Reportes de Discapacidad Intelectual

3.2 Pruebas de Software

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

3.2.1 Métodos de pruebas de software

El uso de las técnicas de pruebas en el desarrollo de un producto de software ayudan a definir conjuntos de casos pruebas aplicando un cierto criterio. Cualquier producto de ingeniería se puede probar de dos formas:

Pruebas de Caja Negra (funcionales): Se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a sus especificaciones. Están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario (teclado, pantalla, ficheros, canales de comunicación). (41)

Pruebas de Caja Blanca (estructurales): Se realiza sobre funciones internas de un módulo. Las pruebas de caja blanca convencen de que un programa hace bien lo que hace; pero no de que haga lo

que realmente se necesita. (41)

3.2.2 Tipo de pruebas de software

Prueba unitaria: es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. (42)

Prueba funcional: es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. (42)

Pruebas de integración: son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. (42)

Pruebas de validación: son el proceso de revisión que el sistema de software producido cumple con las especificaciones y que cumple su cometido. (42)

Prueba del sistema: se encarga de asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. (42)

Pruebas de regresión: son una estrategia de prueba en la cual las pruebas que se han ejecutado anteriormente se vuelven a realizar en la nueva versión modificada, para asegurar la calidad después de añadir la nueva funcionalidad. (42)

3.2.3 Diseño de Caso de Pruebas y Registro de No Conformidades

Un caso de prueba se diseña según las funcionalidades descritas en los casos de usos. Este diseño se elabora previamente a realizar las pruebas funcionales a la aplicación. Se parte de la descripción de los casos de uso del sistema, como apoyo para las revisiones. Cada planilla de caso de prueba recoge la especificación de un caso de uso, dividido en secciones y escenarios, detallando las funcionalidades descritas en él y escribiendo cada variable que recoge el caso de uso. En el proceso de pruebas para el Módulo de Reportes se hace uso del método de prueba de software de Caja Negra por ser éste el método empleado en la liberación del producto software, aplicando el tipo de prueba del sistema.

El diseño de casos de prueba del sistema para el Módulo Reportes Genéticos, en específico para el caso de uso Generar Reportes por Discapacidad Intelectual, incluye los siguientes escenarios:

1. Generar Reporte de Discapacidad Intelectual por Diagnóstico
2. Generar Reporte de Discapacidad Intelectual por Atención

3. Generar Reporte de Discapacidad Intelectual por Visitas
4. Generar Reporte de Discapacidad Intelectual por Total de Discapacitados Intelectual
5. Generar Reporte de Discapacidad Intelectual por Total Necesitan Ayuda
6. Generar Reporte de Discapacidad Intelectual por Total de Huérfanos

Para la realización de las pruebas se seleccionó el escenario Generar Reporte de Discapacidad Intelectual por Diagnóstico. El caso de uso se encarga de mostrar un reporte con la información referente al diagnóstico de los pacientes con discapacidad intelectual. Para la ejecución de esta acción primeramente el usuario debe estar autenticado y luego seleccionar el vínculo Reporte por Diagnóstico.

A partir de esta descripción se detallan las variables que se encuentran en la interfaz asociada al caso de uso que se le diseñó el caso de prueba.

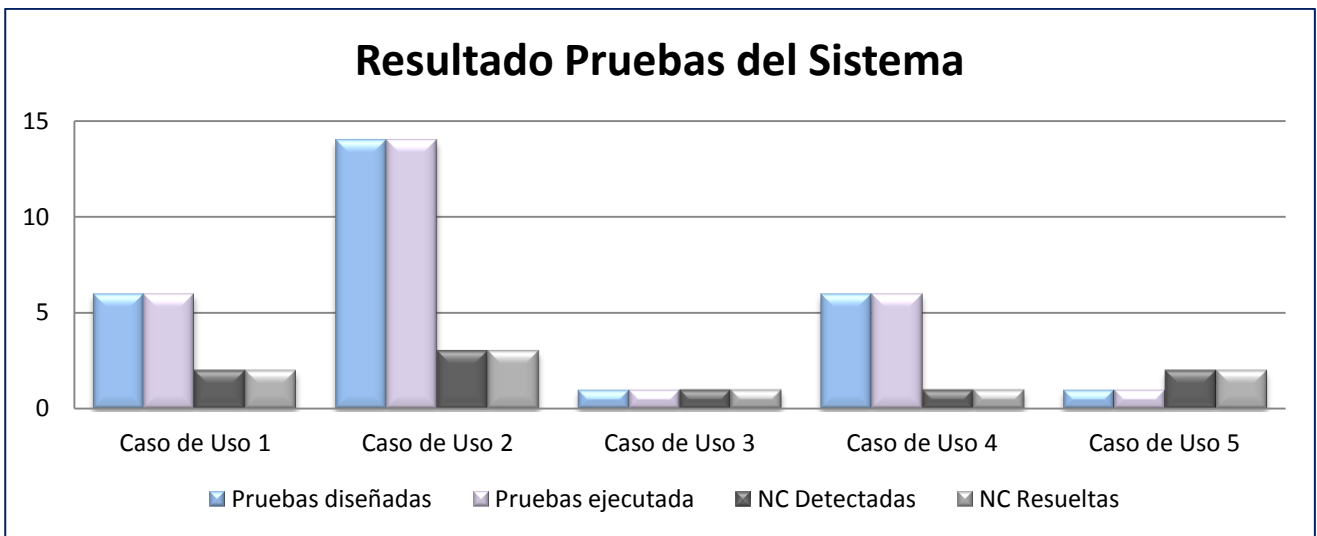
No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	<i>Diagnóstico Retraso Mental</i>	<i>campo de selección</i>	<i>No.</i>	<i>Se debe especificar si como resultado se obtendrán pacientes con diagnóstico de retraso mental o no.</i>
2	<i>Provincia</i>	<i>lista desplegable</i>	<i>Si.</i>	<i>Se selecciona la provincia por la que se desea filtrar.</i>
3	<i>Municipio</i>	<i>lista desplegable</i>	<i>Si.</i>	<i>Se selecciona el municipio por el que se desea filtrar.</i>
4	<i>Consejo Popular</i>	<i>lista desplegable</i>	<i>Si.</i>	<i>Se selecciona el consejo popular por el que se desea filtrar.</i>

Escenario	Descripción	Diagnóstico Retraso Mental	Provincia	Municipio	Consejo Popular	Respuesta del sistema	Flujo central
EC 1.1 Generar Reporte de Discapacidad Intelectual por Diagnóstico	Inserta los datos correctamente	V	V	V	V	Se muestra el reporte correspondiente	1. Se escoge la opción: Reporte por Diagnóstico. 2. Se muestra una interfaz con los criterios de búsqueda para realizar el reporte. 3. Se especifica los criterios a partir de los cuales se desea realizar el reporte. 4. Se muestra el reporte según los criterios entrados por el usuario.
		Si	La Habana	Habana del Este	Alamar Alturas		
EC 1.2 Generar Reporte de Discapacidad Intelectual por Diagnóstico con datos incompletos	Inserta los datos incorrectamente	l sin seleccionar	NA	NA	NA	Se muestra un mensaje definiendo el tipo de error	
EC 1.3 Cancelar la acción	Se cancela la inserción de datos	NA	NA	NA	NA	Retorna a la página de búsqueda para establecer un nuevo criterio de búsqueda	

Este proceso fue realizado para cada uno de los casos de usos del sistema. Los resultados de las pruebas que no fueron satisfactorios pasaron a ser no conformidades y se emitieron en el registro de defectos y dificultades detectados que se encuentra en la parte final de cada diseño de caso de prueba. En la siguiente tabla se muestran algunas de las no conformidades encontradas tras la revisión del software.

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Clasificación	Estado NC	Respuesta de Equipo Desarrollo
Aplicación	1	A la hora de generar el reporte por diagnóstico es obligatorio seleccionar si presenta retraso mental o no, y aunque no se seleccione ninguna de las dos opciones se muestra el reporte	Generar Reportes de Discapacidad Intelectual por Diagnóstico	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 18 de marzo de 2012 RA: 1 de abril de 2012	Se realizaron las validaciones pertinentes
Aplicación	2	En la inserción del código del resultado este campo acepta cualquier símbolo del teclado (*,p,/)	Generar Reportes de Anomalías Cromosómicas por Anomalías	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 18 de marzo de 2012 RA: 2 de abril de 2012	Se realizaron las validaciones pertinentes
Aplicación	3	Para generar un reporte por vínculo laboral a nivel provincial se debe seleccionar la provincia, y aunque no se seleccione se obtienen resultados	Generar Reportes de Discapacidad Física por Vínculo Laboral	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 18 de marzo de 2012 RA: 6 de abril de 2012	Se realizaron las validaciones pertinentes
Aplicación	4	Cuando se desea seleccionar la clasificación de la enfermedad por la que se desea filtrar el sistema no cargan dichas clasificaciones	Generar Reportes de Enfermedades Genéticas por Enfermedades	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 20 de marzo de 2012 RA: 6 de abril de 2012	Se realizaron las validaciones pertinentes

3.2.3 Resultado Obtenidos



Con el objetivo de garantizar el correcto funcionamiento de la aplicación se realizaron las pruebas de sistema. Para ello se diseñaron 28 pruebas, uno por cada requerimiento funcional, distribuidos en los 5 casos de uso, que representan los cinco módulos existentes en la aplicación. Se ejecutaron las 28 pruebas, de las cuales se detectaron 9 no conformidades que fueron tratadas y finalmente solucionadas con éxito.

Luego de analizar los datos arrojados por las pruebas se puede asumir que la aplicación cumple cabalmente con las expectativas del cliente.

3.3 Conclusiones Parciales

Con la realización de este capítulo se describió como fue implementada la aplicación en términos de componentes de cada uno de los casos de usos del sistema. Al mismo tiempo se brindó una breve descripción de las pruebas realizadas al sistema. En total se llevaron a cabo 28 pruebas de sistema, en las cuales se detectaron 9 no conformidades que fueron evaluadas y finalmente solucionadas con éxito. Por lo que se asume que la aplicación se encuentra en condiciones para ser utilizada.

CONCLUSIONES

- El Módulo de reportes del sistema alasMEDIGEN incrementó la cantidad de reportes disponibles para los genetistas, ofreciéndoles información más detallada para el estudio de las enfermedades genéticas.
- Las herramientas y metodología utilizadas en el desarrollo del Módulo de Reportes del sistema alasMEDIGEN están ajustadas a las especificaciones establecidas por el marco regulatorio del GIS y favorecieron la integración con el sistema GDR.
- El diseño guiado por patrones de diseño y el estilo arquitectónico MVC facilitó el mantenimiento del sistema propuesto al desacoplar los componentes y proporcionó una lógica organizacional que conserva la alta cohesión y el bajo acoplamiento entre los componentes del sistema.
- La implementación del Módulo de reportes del sistema alasMEDIGEN está basada en componentes comprensibles y manejables.
- El diseño y ejecución de pruebas del sistema permitieron comprobar el correcto funcionamiento del Módulo de Reportes para el sistema alasMEDIGEN, luego de quedar resueltas todas las no conformidades detectadas.

Recomendaciones

Teniendo en cuenta la importancia de los estudios de enfermedades genéticas, defectos congénitos y el bienestar de la población, se recomienda el despliegue del Módulo de Reportes genéticos en el CNGM, además de una futura integración con el módulo de Visualización Geográfica, el cual permite visualizar geográficamente los datos contenidos en un reporte determinado. Esta integración contribuirá a un mejor diagnóstico y prevención de estos problemas.

Referencias

1. **sld**. Centro Nacional de Información de Ciencias Médicas¹. [En línea] 24 de octubre de 1999-2011. http://www.sld.cu/sitios/genetica/verpost.php?pagina=1&blog=http://articulos.sld.cu/genetica&post_id=195&c=2987&tipo=2&idblog=141&p=1&n=de.
2. **Slideshare**. SlideShare Inc. Estadística en la Genética. [En línea] 2011. [Citado el: 26 de octubre de 2011.] <http://www.slideshare.net/primusformax/estadistica-en-la-gentica>.
3. **ciberhabitat**. Historia de la Informática. [En línea] [Citado el: 26 de octubre de 2011.] <http://www.ciberhabitat.gob.mx/museo/historia/>.
4. **sld**. Centro Nacional de Información de Ciencias Médicas. [En línea] 1999-2011. [Citado el: 24 de octubre de 2011.] http://www.sld.cu/sitios/genetica/verpost.php?blog=http://articulos.sld.cu/genetica&post_id=391&c=2987&tipo=2&idblog=141&p=1&n=de.
5. **sipec**. Reportes. [En línea] [Citado el: 2 de noviembre de 2011.] <http://sipec.sep.gob.mx/WebHelp/sipecwebhelp.htm#reportes/reporte.htm>.
6. **BuenasTareas**. BuenasTareas. *BuenasTareas*. [En línea] 2011. [Citado el: 20 de noviembre de 2011.] <http://www.buenastareas.com/ensayos/Sistemas-De-Informacion/1449538.html>.
7. **myCRMweb**. myCRMweb. *myCRMweb*. [En línea] 1995. [Citado el: 5 de octubre de 2011.] <http://www.mycrmweb.com/es/news-es/news-es/109-mycrmweb-crystal-reports-es>.
8. **Intelligence, Business**. BI - Business Intelligence. [En línea] 27 de mayo de 2009. [Citado el: 10 de octubre de 2011.] <http://bi-businessintelligence.blogspot.com/>.
9. **reportman**. Report Manager. *Report Manager*. [En línea] [Citado el: 10 de octubre de 2011.] <http://reportman.sourceforge.net/indexes.html>.
10. **instantatlas**. InstantAtlas. *InstantAtlas*. [En línea] 2009. [Citado el: 15 de noviembre de 2011.] http://www.instantatlas.com/es/ia_features.xhtml.
11. **Rodríguez Duran, Aurelio**. *Implementación del módulo Diseñador de Reportes para el Generador Dinámico de Reportes incluido en la Plataforma de Ayuda para la Toma de Decisiones y Soluciones Integrales*. Ciudad de La Habana : s.n., 2011.
12. **Softel**. *Marco Regulatorio GIS*.
13. **Padrón, Carlos, Hernández, Luis y Izquierdo, Liandy**. *Sistema Informático de Genética Médica: Diseño*. 2008.
14. **Write Export**. METODOLOGIA RUP Rational Unified Process (RUP). *METODOLOGIA RUP Rational Unified Process (RUP)*. [En línea] [Citado el: 1 de noviembre de 2011.] <https://export.writer.zoho.com/public/dianacovarrubias/metodologia-rup/fullpage>.
15. **Ramírez, Alejandro, Vanpeperstraete, Philippe y Rueckert, Andreas**. *ArgoUML: Manual de Usuario*. 2011.
16. **Criptom, Jimmy**. Intercambios Virtuales. *Intercambios Virtuales*. [En línea] <http://www.intercambiosvirtuales.org/software/rational-rose-enterprise-edition-suite-2003>.
17. **freedownloadmanager**. Sitio de Descarga de Software VP. *Sitio de Descarga de Software VP*. [En línea] [Citado el: 25 de noviembre de 2011.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
18. **Desarrollo Web**. Breve historia de PHP. [En línea] 7 de junio de 2001. [Citado el: 15 de noviembre de 2011.] <http://www.desarrolloweb.com/articulos/436.php>.
19. **synergix**. Visión de Synergix de los Sistemas de Información y la Ingeniería del Software. *Visión de Synergix de los Sistemas de Información y la Ingeniería del Software*. [En línea] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
20. **CAVSI**. CAVSI. CAVSI. [En línea] <http://www.cavsi.com/preguntasrespuestas/que-es-un-servidor-web/>.
21. **Ciberaula**. Una Introducción a Apache. *Una Introducción a Apache*. [En línea] 2010. [Citado el: 24

- de octubre de 2011.] http://linux.ciberaula.com/articulo/linux_apache_intro.
22. **Scribd**. Scribd. *Scribd*. [En línea] 2011. [Citado el: 20 de noviembre de 2011.] <http://es.scribd.com/doc/58274826/MySQL-Diapositivas>.
23. **Briano, Fernando**. *Control de versiones con Subversion*. 2008.
24. **Atenas**. Introducción al software libre. *Introducción al software libre*. [En línea] [Citado el: 2 de noviembre de 2011.] http://www.atenas.cult.cu/rl/informatica/manuales/sl/introduccion_al_SL/sec-ide.html.
25. **Guía Ubuntu**. Guía Ubuntu Eclipse. *Guía Ubuntu Eclipse*. [En línea] 20 de mayo de 2011. [Citado el: 12 de noviembre de 2011.] <http://www.guia-ubuntu.org/index.php?title=Eclipse>.
26. **Softdownload**. Eclipse 3.5.2. *Softdownload*. [En línea] 2006. <http://www.softdownload.org/programas/eclipse-3-5-2.asp>.
27. **NetBeans**. NetBeans. *NetBeans*. [En línea] 2011. [Citado el: 15 de noviembre de 2011.] http://netbeans.org/index_es.html.
28. **netbeans**. NetBeans(symfony). [En línea] 2011. [Citado el: 15 de noviembre de 2011.] http://wiki.netbeans.org/NB68symfony_es.
29. **sld**. CNGM Sitio web Genética Médica. *CNGM Sitio web Genética Médica*. [En línea] <http://www.sld.cu/sitios/genetica/>.
30. **synergix**. Visión de Synergix de los Sistemas de Información y la Ingeniería del Software1. *Visión de Synergix de los Sistemas de Información y la Ingeniería del Software1*. [En línea] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
31. **JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James**. *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
32. **Abad, Ingrid y Curiel, Jorge Daniar**. *alasMEDIGEN: Registro Cubano de Anomalías Cromosómicas*. 2009.
33. **MCC**. Diagramas de Casos de Uso del Sistema. *Diagramas de Casos de Uso del Sistema*. [En línea] <http://www.mcc.unam.mx/~cursos/Objetos/Cap17/cap17.html>.
34. **Dorado, Andrés**. Patrones de diseño. [En línea] <http://www.slideshare.net/2008PA2Info3/tema-de-revision-2003>.
35. **Soto, Juan**. Presentacion Patrones De Diseno GoF. [En línea] 8 de octubre de 2010. <http://www.slideshare.net/juansoto86/presentacion-patrones-de-diseno-gof>.
36. **Crespo García, Yosúan y Sterling Velásquez, Otto Leosdán**. *Sistema Informático para la Red Nacional de Genética Médica (SIGM): Registro Cubano de Retraso Mental versión 2.0*. Facultad 6. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.
37. **Potencier, F. y Zaninotto, F**. *Symfony la guía definitiva*. 2007.
38. **Ferré Grau, Xavier y Sánchez Segura, María Isabel**. Scribd,diagrama clase diseño. [En línea] <http://es.scribd.com/>.
39. **Pressman, R.S**. *Ingeniería de Software. Un enfoque práctico*. 2002.
40. **Jacobson, Booch y Rumbaugh, J**. *El Proceso Unificado de Desarrollo de Software*. s.l. : PEARSON EDUCACIÓN S.A, 2000.
41. **AVENDAÑO ROMERO, LEIDY JOHANNA**. *TECNICAS DE PRUEBA DE SOFTWARE*. Bogota : s.n., 2008.
42. **Méndez, Laura Cecilia**. *Tipos de pruebas*. 2012.

Bibliografía

1. sld. Centro Nacional de Información de Ciencias Médicas1. [En línea] 24 de octubre de 1999-2011. http://www.sld.cu/sitios/genetica/verpost.php?pagina=1&blog=http://articulos.sld.cu/genetica&post_id=195&c=2987&tipo=2&idblog=141&p=1&n=de.
2. Slideshare. SlideShare Inc. Estadística en la Genética. [En línea] 2011. [Citado el: 26 de octubre de 2011.] <http://www.slideshare.net/primusformax/estadstica-en-la-gentica>.
3. ciberhabitat. Historia de la Informática. [En línea] [Citado el: 26 de octubre de 2011.] <http://www.ciberhabitat.gob.mx/museo/historia/>.
4. sld. Centro Nacional de Información de Ciencias Médicas. [En línea] 1999-2011. [Citado el: 24 de octubre de 2011.] http://www.sld.cu/sitios/genetica/verpost.php?blog=http://articulos.sld.cu/genetica&post_id=391&c=2987&tipo=2&idblog=141&p=1&n=de.
5. sipec. Reportes. [En línea] [Citado el: 2 de noviembre de 2011.] <http://sipec.sep.gob.mx/WebHelp/sipecwebhelp.htm#reportes/reporte.htm>.
6. BuenasTareas. BuenasTareas. *BuenasTareas*. [En línea] 2011. [Citado el: 20 de noviembre de 2011.] <http://www.buenastareas.com/ensayos/Sistemas-De-Informacion/1449538.html>.
7. myCRMweb. myCRMweb. *myCRMweb*. [En línea] 1995. [Citado el: 5 de octubre de 2011.] <http://www.mycrmweb.com/es/news-es/news-es/109-mycrmweb-crystal-reports-es>.
8. Intelligence, Business. BI - Business Intelligence. [En línea] 27 de mayo de 2009. [Citado el: 10 de octubre de 2011.] <http://bi-businessintelligence.blogspot.com/>.
9. reportman. Report Manager. *Report Manager*. [En línea] [Citado el: 10 de octubre de 2011.] <http://reportman.sourceforge.net/indexes.html>.
10. instantatlas. InstantAtlas. *InstantAtlas*. [En línea] 2009. [Citado el: 15 de noviembre de 2011.] http://www.instantatlas.com/es/ia_features.xhtml.
11. Rodríguez Duran, Aurelio. *Implementación del módulo Diseñador de Reportes para el Generador Dinámico de Reportes incluido en la Plataforma de Ayuda para la Toma de Decisiones y Soluciones Integrales*. Ciudad de La Habana : s.n., 2011.
12. Softel. *Marco Regulatorio GIS*.
13. Padrón, Carlos, Hernández, Luis y Izquierdo, Liandy. *Sistema Informático de Genética Médica: Diseño*. 2008.
14. Write Export. METODOLOGIA RUP Rational Unified Process (RUP). *METODOLOGIA RUP Rational Unified Process (RUP)*. [En línea] [Citado el: 1 de noviembre de 2011.] <https://export.writer.zoho.com/public/dianacovarrubias/metodologia-rup/fullpage>.
15. Ramirez, Alejandro, Vanpeperstraete, Philippe y Rueckert, Andreas. *ArgoUML: Manual de Usuario*. 2011.
16. Cripto, Jimmy. Intercambios Virtuales. *Intercambios Virtuales*. [En línea] <http://www.intercambiosvirtuales.org/software/rational-rose-enterprise-edition-suite-2003>.
17. freedownloadmanager. Sitio de Descarga de Software VP. *Sitio de Descarga de Software VP*. [En línea] [Citado el: 25 de noviembre de 2011.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
18. Desarrollo Web. Breve historia de PHP. [En línea] 7 de junio de 2001. [Citado el: 15 de noviembre de 2011.] <http://www.desarrolloweb.com/articulos/436.php>.
19. synergix. Visión de Synergix de los Sistemas de Información y la Ingeniería del Software. *Visión de Synergix de los Sistemas de Información y la Ingeniería del Software*. [En línea] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
20. CAVSI. CAVSI. *CAVSI*. [En línea] <http://www.cavsi.com/preguntasrespuestas/que-es-un-servidor-web/>.
21. Ciberaula. Una Introducción a Apache. *Una Introducción a Apache*. [En línea] 2010. [Citado el: 24

- de octubre de 2011.] http://linux.ciberaula.com/articulo/linux_apache_intro.
22. Scribd. Scribd. *Scribd*. [En línea] 2011. [Citado el: 20 de noviembre de 2011.] <http://es.scribd.com/doc/58274826/MySQL-Diapositivas>.
23. Briano, Fernando. *Control de versiones con Subversion*. 2008.
24. Atenas. Introducción al software libre. *Introducción al software libre*. [En línea] [Citado el: 2 de noviembre de 2011.] http://www.atenas.cult.cu/rl/informatica/manuales/sl/introduccion_al_SL/secide.html.
25. Guia Ubuntu. Guia Ubuntu Eclipse. *Guia Ubuntu Eclipse*. [En línea] 20 de mayo de 2011. [Citado el: 12 de noviembre de 2011.] <http://www.guia-ubuntu.org/index.php?title=Eclipse>.
26. Softdownload. Eclipse 3.5.2. *Softdownload*. [En línea] 2006. <http://www.softdownload.org/programas/eclipse-3-5-2.asp>.
27. NetBeans. NetBeans. *NetBeans*. [En línea] 2011. [Citado el: 15 de noviembre de 2011.] http://netbeans.org/index_es.html.
28. netbeans. NetBeans(symfony). [En línea] 2011. [Citado el: 15 de noviembre de 2011.] http://wiki.netbeans.org/NB68symfony_es.
29. sld. CNGM Sitio web Genética Médica. *CNGM Sitio web Genética Médica*. [En línea] <http://www.sld.cu/sitios/genetica/>.
30. synergix. Visión de Synergix de los Sistemas de Información y la Ingeniería del Software1. *Visión de Synergix de los Sistemas de Información y la Ingeniería del Software1*. [En línea] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
31. JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
32. Abad, Ingrid y Curiel, Jorge Daniar. *alasMEDIGEN: Registro Cubano de Anomalías Cromosómicas*. 2009.
33. MCC. Diagramas de Casos de Uso del Sistema. *Diagramas de Casos de Uso del Sistema*. [En línea] <http://www.mcc.unam.mx/~cursos/Objetos/Cap17/cap17.html>.
34. Dorado, Andrés. Patrones de diseño. [En línea] <http://www.slideshare.net/2008PA2Info3/tema-de-revision-2003>.
35. Soto, Juan. Presentacion Patrones De Diseno GoF. [En línea] 8 de octubre de 2010. <http://www.slideshare.net/juansoto86/presentacion-patrones-de-diseno-gof>.
36. Crespo García, Yosván y Sterling Velásquez, Otto Leosdán. *Sistema Informático para la Red Nacional de Genética Médica (SIGM): Registro Cubano de Retraso Mental versión 2.0*. Facultad 6. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.
37. Potencier, F. y Zaninotto, F. *Symfony la guía definitiva*. 2007.
38. Ferré Grau, Xavier y Sánchez Segura, María Isabel. Scribd,diagrama clase diseño. [En línea] <http://es.scribd.com/>.
39. Pressman, R.S. *Ingeniería de Software. Un enfoque práctico*. 2002.
40. Jacobson, Booch y Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*. s.l. : PEARSON EDUCACIÓN S.A, 2000.
41. AVENDAÑO ROMERO, LEIDY JOHANNA. *TECNICAS DE PRUEBA DE SOFTWARE*. Bogota : s.n., 2008.
42. Méndez, Laura Cecilia. *Tipos de pruebas*. 2012.
43. Potencier, Fabien y Zaninotto, François. *Symfony la guía definitiva*. 2008.
44. Perodín, Y.S. *Estilo de código del Sistema de Información de Genética Médica*. Ciudad Habana. : Facultad de Bioinformática. Universidad de Ciencias Informáticas, 2008.
45. Nicolás, Eduardo y Inés, Virginia. *Business Intelligence: Negocios Inteligentes para Empresas Inteligentes*. Universidad Nacional de La Rioja. Argentina : s.n.
46. LARMAN, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : s.n., 1999.

47. Garcia Alpa, Viki Clara y Quintuña, Rodrigo. *Sistema Generador de Reportes Dinámicos para Web, configurable para las plataformas de Bases de datos conocidas*. Escuela Politécnica Nacional. Quito : s.n., 2010.
48. Franco, L y Catrin, M. *Uso practico de CVS para control de versiones*. 2003.
49. Armas, E.M. and E.O. Fonseca. *Sistema Informático para la Red Nacional de Genética Médica: Definición de la Arquitectura de Software*. Ciudad Habana : Facultad 6, 2008.
50. Arevalo, Maria Eugenia. Introducción al Patrón de Arquitectura por Capas . [En línea] [Citado el: 2 de diciembre de 2011.] <http://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas/>.
51. Los Portables. *Los Portables. PHP ReportMaker 3.0.2*. [En línea] <http://www.losportables.info/php-report-maker-3-0-2-portable.html>.
52. intergraphicdesigns. *IntergraphicDESIGNS*. [En línea] <http://www.intergraphicdesigns.com>.
53. *Definición de Servidor IIS*.
54. vivaphp. *Completado el soporte de Symfony para NetBeans 6.8*. [En línea] <http://www.vivaphp.com.ar/frameworks/symfony-para-netbeans>].
55. desarrolloweb. *Breve Historia de PHP*. [En línea] julio de 2001. <http://www.desarrolloweb.com/articulos/436.php>.
56. alasMEDIGEN. *alasMEDIGEN 1.2: SISTEMA INFORMÁTICO DE GENÉTICA MÉDICA*.

Glosario de Términos

API: Application Programming Interface, Interfaz de Programación de Aplicaciones.

BSD: Berkeley Software Distribution del español Distribución de Software de Berkeley.

CASE: Computer Aided Software Engineering del español Ingeniería de Software Asistida por Computadoras.

CNGM: Centro Nacional de Genética Médica.

CSV: Comma Separated Values (valores separados por coma), formato sencillo que almacena datos en tablas (números y texto) en formato de texto.

CVS: Concurrent Version System.

Data Managers: Datos de Administradores.

FTP: Protocolo de Transferencia de Archivos.

GDR: Generador Dinámico de Reportes.

GPL: General Public License.

HTML: HyperText Markup Language (lenguaje de marcado de hipertexto).

IBM: International Business Machines.

IDE: Integrated Development Environment del español Entorno Integrado de Desarrollo.

IBM: International Business Machines.

JDK: Java Development Kit.

J2RE: Run Time Environment.

MPL: Mozilla Public License.

NNTP: Protocolo de Transferencia de Noticias a través de la Red.

OEM: Original Equipment Manufacturer, del español fabricante de equipamiento original.

PDF: Formato de Documento Portátil.

Publisher: Editor.

RECUAC: Registro Cubano de Anomalías Cromosómicas.

RECUDI: Registro Cubano de Discapacidad Intelectual.

RECUDIS: Registro Cubano de Discapacitados.

RECUEC: Registro Cubano de Enfermedades Comunes.

RECUEGEN: Registro Cubano de Enfermedades Genéticas.

RECUGEM: Registro Cubano de Gemelos.

RECUHCL: Registro Cubano de Historias Clínicas Genéticas.

RECUMAC: Registro Cubano de Malformaciones Congénitas.

RTF: Rich Text Format (formato de texto enriquecido).

RUP: ProcesoUnificadoRacional (rational unified processor).

SAAA: Seguridad Autenticación Autorización Auditoria

SDK: Software Development Kit.

SMTP: Protocolo Simple de Transferencia de Correo.

Templates: Plantillas.

UML: Lenguaje Unificado de Modelado.

XLS: Formato de Archivo de Microsoft Excel.

XML: Extensible MarkupLanguage (lenguaje de marcas extensible).