

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
DIRECCIÓN DE FORMACIÓN POSTGRADUADA



MÉTODOS COMPUTACIONALES PARA LA REPRESENTACIÓN Y ANÁLISIS DE ÁRBOLES GENEALÓGICOS

TESIS PRESENTADA
EN OPCIÓN AL TÍTULO DE
MÁSTER EN INFORMÁTICA APLICADA

Autor: *Ing.* Reynaldo Alvarez Luna

Tutores: *M.Sc.* Longendri Aguilera Mendoza
Dra.Sc. Beatriz Marcheco Teruel

Ciudad de La Habana, Julio de 2011

DECLARACIÓN JURADA DE AUTORÍA

Yo, Reynaldo Alvarez Luna con carné de identidad 84082423909, declaro que soy el autor principal del resultado que expongo en la presente tesis titulada Métodos computacionales para la representación y análisis de árboles genealógicos, para optar por el título académico de Máster en Informática Aplicada. Este trabajo fue desarrollado durante el periodo comprendido entre 2009 y 2011, en colaboración con los colegas de equipo Ing. Yosvany Arrastia Machin, Ing. Rayko Emilio Torres Cruz, Ing. Reynaldo Rosado Roselló, Ing. Enrique Sánchez Fernández, Ing. Daniel Revilla, quienes reconocen a nombre de este autor la responsabilidad principal del resultado expuesto. A ellos, el reconocimiento por sus cualidades para el trabajo en equipo y el agradecimiento sincero. En especial, agradecer al MSc. Longendri Aguilera Mendoza y a la Dra.C Beatriz Marcheco Teruel, quienes fungieron como tutores y a los profesores MSc. Yeleny Zulueta Veliz, MSc. Héctor Raúl González Diez, MSc. David Silva que contribuyeron a través de sus consejos y guías oportunas durante el proceso de la investigación. A todos ellos, así como a otros colegas y amigos que no he mencionado por razones de espacio, les doy las más sinceras gracias. Finalmente, declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional. Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana, a los días del mes de julio del año 2011.

Ing. Reynaldo Alvarez Luna
Firma del autor

a mi especial familia

Resumen

El árbol genealógico es el recurso más utilizado por los especialistas en genética clínica para el estudio de las familias o individuos en el proceso de asesoramiento genético. Luego de una valoración de las herramientas existentes con este objetivo en el ámbito internacional, el Centro Nacional de Genética Médica impulsó el desarrollo de un sistema propio para la representación de árboles genealógicos denominado *ARBOGEN*. Sistema que al ser desplegado presentó una serie de limitaciones en la representación gráfica de los árboles genealógicos y su análisis. En consecuencia, al evaluar el sistema se identifica la necesidad de replantear la solución en cuanto a la arquitectura y la estructura de datos utilizadas para la modelación de los árboles genealógicos. Se impone el desarrollo de métodos computacionales y componentes de software basados en una estructura de datos eficiente para la representación de relaciones como los grafos, que faciliten el análisis de árboles genealógicos y su representación desde diferentes perspectivas. Los métodos y componentes implementados se integraron en una herramienta informática desarrollada en Java que cuenta además con una base de datos, un sistema basado en reglas para la investigación de enfermedades según la historia familiar, entre otras funcionalidades visuales relevantes. La integración de estos métodos computacionales y componentes en una nueva versión de la herramienta, hace de esta una solución funcional en total correspondencia con las tendencias actuales en el desarrollo de las TICs para esta área y las necesidades de los especialistas cubanos en el proceso de asesoramiento genético y en sus investigaciones.

Palabras claves: Algoritmos, Árbol Genealógico, Genética, Métodos computacionales.

Abstract

The genealogical tree is the most used resource by specialists in clinical genetics for the study of families or individuals in the process of genetic counseling. After an assessment of existing tools for this purpose in the international scope, the National Center for Medical Genetics promoted the development of an own system for the representation of genealogical tree called *alasARBOGEN*. System to be deployed shows a number of limitations in the graphical representation of genealogical tree and its analysis. Consequently, when evaluating the system identifies the need of rethink the solution in terms of architecture and data structure for genealogical tree modeling. It requires the development of computational methods and software components based on an efficient data structure for representation of relationships as graphs, to facilitate the analysis of genealogical tree and its representation from different perspectives. The methods and implemented components are integrated into an informatics tool developed in Java that also has a database, a rules-based system for the research of diseases according to family history, among other relevant visual features. The integration of these computational methods and components in a new version of the tool makes this a functional solution in all correspondence with the current trends in ITC development for this area and the needs of Cuban specialists in the genetic counseling and their researches.

Keywords: Algorithms, Computational Methods, Clinical Genetics, Genealogical tree.

Índice general

DECLARACIÓN JURADA DE AUTORÍA	i
Resumen	iii
Introducción	1
1 Revisión bibliográfica del proceso de representación y análisis de árboles genealógicos	8
1.1 Estandarización de la representación de árboles genealógicos	9
1.2 Herramientas informáticas para la representación y análisis de árboles genealógicos	9
1.2.1 Herramientas web para la representación de árboles genealógicos. . .	10
1.2.2 Herramientas de escritorio para la representación de árboles genealógicos	11
1.3 Representación de árboles genealógicos	14
1.4 Análisis de árboles genealógicos	17
2 Materiales y métodos para el desarrollo de los métodos computacionales	20
2.1 Estructura de dato grafo	20

2.2	Plataforma Java	21
2.2.1	Eventos visuales	22
2.2.2	Anotaciones	22
2.2.3	Reflexión	23
2.3	Entorno y herramientas de desarrollo	23
2.3.1	IDE Eclipse	23
2.3.2	Hibernate	24
2.3.3	Spring	24
2.3.4	Drools	24
2.4	Arquitectura y Tecnologías	25
2.4.1	Arquitectura cliente-servidor	25
2.4.2	Estilos arquitectónicos	26
2.5	Metodología para la representación de árboles genealógicos	27
2.5.1	Manejo de eventos visuales	27
2.5.2	Visualización y edición de los datos personales	28
2.6	Análisis sobre árboles genealógicos	29
2.6.1	Algoritmos sobre grafo	29
2.6.2	Sistemas expertos basados en reglas	30
2.7	Metodología de software	30
2.7.1	Requisitos funcionales y no funcionales	31
2.7.2	Componentes de software	32

2.8	Definición de los indicadores a evaluar	32
3	Resultados y discusión	35
3.1	Métodos computacionales para la representación de árboles genealógicos . .	35
3.1.1	Representación del árbol genealógico en el grafo diseñado	36
3.1.2	Implementación de los eventos visuales y componentes necesarios para la visualización computacional de los árboles genealógicos . .	36
3.2	Métodos computacionales para el análisis de árboles genealógicos	38
3.2.1	Algoritmo para la unión de árboles genealógicos	38
3.2.2	Algoritmo para la acotación de árboles genealógicos	39
3.2.3	Algoritmo para el ordenamiento de los árboles genealógicos	40
3.2.4	Sistema basado en reglas para el cálculo de riesgo genético	41
3.3	Sistema informático para la integración de los métodos computacionales . .	44
3.3.1	Componentes de la herramienta informática implementada	46
3.4	Validación de los resultados de la implementación	50
3.4.1	Análisis de la complejidad temporal de los métodos computacionales implementados.	50
3.4.2	Pruebas funcionales al sistema informático implementado.	50
3.4.3	Usabilidad del sistema informático.	51
3.5	Visión global del trabajo realizado	51
	Conclusiones	54
	Recomendaciones	55

Referencias Bibliográficas	56
Anexos	61

Introducción

La aplicación de la informática en el sector de la salud se ha manifestado a través de sistemas para el control hospitalario, sistemas expertos de diagnóstico y la prevención de enfermedades, entre otros más especializados en alguna rama de la medicina. Han sido desarrollados con el objetivo de mejorar la calidad de vida de la población y la eficiencia de los servicios médicos. Cuba, considerada una potencia médica y reconocida por el prestigio de sus galenos, ha creado la Red de Salud (Infomed)¹, para facilitar el intercambio de especialistas, crear espacios para la publicación, y desplegar diversas aplicaciones informáticas que se han desarrollado paulatinamente en el país dedicadas a esta rama.

Con el surgimiento del Centro Nacional de Genética Médica (CNGM) que tiene como misión fundamental la dirección del Programa Cubano de Diagnóstico Manejo y Prevención de Enfermedades Genéticas y Defectos Congénitos (Estudios de Discapacidad, Gemelos, Malformaciones congénitas, entre otros) y de la Red Nacional de Genética (184 centros en todo el país) Cuba eleva el nivel de la atención primaria en esta especialidad y desarrolla múltiples programas para la atención a la población. De inmediato como parte fundamental de la ejecución de su misión impulsa proyectos para la automatización de sus estudios y servicios.

Los árboles genealógicos representan la genealogía (definida como el “Estudio del desarrollo genético de individuos y razas; enumeración de ancestros y descendientes; filogenia” [1] de una persona en forma de árbol, constituyen la base de todo estudio genético acerca de la historia familiar de los pacientes. Por lo que “El recurso más utilizado por los profesionales de la genética médica desde hace un siglo, es el árbol genealógico familiar que realizamos a un paciente referido a nuestra consulta ante un posible diagnóstico genético, para evaluar la transmisión del rasgo o enfermedad en la familia y asesorar sobre el riesgo a los miembros que lo soliciten.” [2]

¹Disponible en: <http://www.sld.cu/>

Los avances científicos ocurridos en los últimos años ponen a disposición de la medicina adelantos tecnológicos que han hecho que algunos planteen que el método clínico está en peligro [3]. A través de estudios de laboratorio es posible determinar las bases moleculares de las enfermedades, los fármacos ideales según las características del genoma y personalizar la atención médica. Pero estos estudios no están al alcance de todos, los precios de los reactivos, la tecnología y los materiales son elevados, nuestro país tiene un acceso limitado a estas técnicas avanzadas. Debido a esto los especialistas cubanos, utilizan como herramienta primaria para la atención de los pacientes, su historia familiar y esencialmente, el árbol genealógico.

Antecedentes

Los árboles genealógicos en la prevención de enfermedades de origen genético

El árbol genealógico es la herramienta de diagnóstico más tradicional en genética clínica. Según los familiares afectados, o no, se identifican los patrones de herencia, se calculan los riesgos y se distinguen los factores genéticos. El árbol sirve además para tomar decisiones en el manejo médico, tales como la estrategia en los estudios que se van a indicar y las opciones reproductivas que se pueden brindar; pero además el proceso de confección del árbol genealógico es una excelente oportunidad para establecer rapport con el paciente y luego, de trazado, sirve para educarlo.[3]

A nivel mundial las organizaciones de salud consultoras genéticas coinciden en la utilización de los árboles genealógicos como primera herramienta en el diagnóstico y en la predicción de riesgos, tal es el caso de la International Cancer Consultants que define como el primer paso de su consulta de consejo genético para el cáncer hereditario, la construcción del árbol genealógico.[4]

La genética comunitaria en Cuba

En nuestro país son tratadas numerosas enfermedades de origen genético, y se pueden prevenir a través del estudio de los árboles genealógicos, según el tipo de herencia y el grado de parentesco que tengan los individuos. La experiencia ha sido positiva en la prevención de cáncer hereditario, la retinosis pigmentaria, entre otras enfermedades.

Cuba dispone hoy de 1 asesor genético por cada 15 695 habitantes, ubicado en la

atención primaria. Se trata de un profesional entrenado para trabajar en la medicina comunitaria con un enfoque multidisciplinario e integrador, que permite colocar a la genética médica y sus avances tecnológicos en el primer nivel de contacto entre los individuos, la familia y la comunidad con el sistema nacional de salud y garantiza de manera permanente la atención sanitaria en este campo.[5]

Este carácter de la genética en Cuba, permite desarrollar estudios sin precedentes en otros países, manejar la información genética desde la comunidad con la ética y la profesionalidad que caracteriza a los especialistas cubanos. Unido a sistemas que permitan una buena gestión de la información y faciliten el desarrollo de investigaciones, es una garantía para la prevención y detección temprana de enfermedades.

Sistema para la representación de árboles genealógicos: alasARBOGEN 1.0

En el año 2004, se priorizó la implementación de un sistema informático para la representación de árboles genealógicos, que permitiera realizar el árbol genealógico a un paciente referido a la consulta. Este sistema fue desarrollado por la Universidad de las Ciencias Informáticas (UCI) y desplegado en su primera versión en el año 2009 (ver Anexo 1.7) bajo el nombre de alasARBOGEN 1.0. Este sistema significó un paso de avance en el desarrollo del programa de informatización del CNGM en el país e implicó el despliegue en la Red Nacional de Genética de un sistema propio basado en los requerimientos de los especialistas cubanos, permitiendo mejoras en el trabajo de los especialistas y el intercambio de criterios entre estos.

El uso generalizado de alasARBOGEN 1.0 y el crecimiento de las demandas de los estudios genéticos y psicosociales desarrollados en Cuba para mejorar la capacidad de diagnóstico y prevención de enfermedades, han expuesto limitaciones del sistema tales como: problemas en la gestión gráfica, manejo de ficheros de texto o binarios únicamente, formato de los árboles genealógicos, seguridad de los árboles, entre otros.

Al realizar un análisis profundo del sistema informático, se observa una ineficiente definición de las estructuras de datos seleccionadas para la modelación de las clases y entidades del negocio. Las relaciones entre las personas son modeladas a través de listas para cada tipo de relación, lo que trae como consecuencia la implementación de métodos computacionales complejos espacial y temporalmente, y limitaciones en la implementación de nuevas funcionalidades que sean complejas sobre esa versión del sistema informático.

Este estudio además implicó una actualización de los requerimientos del sistema por parte de los clientes y una profundización en la mejora del proceso de asesoramiento

genético según las tendencias actuales de las herramientas existentes, para definir las áreas fundamentales a automatizar. “El asesoramiento genético resulta un valioso instrumento de educación a pacientes y sus familiares en riesgo para cualquier enfermedad parcial o totalmente determinada por alteraciones genéticas” [2]. Este proceso de asesoramiento puede ser descrito a través una serie de actividades de las que se mencionarán las posibles a automatizar [6]:

1. Construcción de la historia familiar y el árbol genealógico mostrando todos los problemas médicos.
2. Analizar la historia familiar para determinar la medida en que puede tener una enfermedad hereditaria o congénita.
3. Verificación de la historia familiar.
4. Cálculo e interpretación del riesgo para ocurrencias o recurrencias de condiciones genéticas en la familia.
5. Discusión de la naturaleza de las condiciones, incluyendo la contribución de la herencia.
6. Discusión de las opciones disponibles para la reducción de riesgos, realización de exámenes, en ocasiones incluye planes de exámenes en el que otros miembros de la familia deben ser los primeros de ser posible, para tener más información.

Al analizar los pasos expuestos para el proceso de asesoramiento genético es posible identificar que la primera versión del sistema desplegada en la Red de genética estaba orientada únicamente a la construcción de la historia familiar por lo que el resto de los pasos posibles a automatizar en el proceso no han sido cubiertos y esto limita la gestión de los especialistas y el desarrollo de investigaciones en un mundo cada día más competitivo y enfocado al uso de tecnologías en todas las ramas de la ciencia.

Dada la importancia del estudio de la historia familiar y en particular de la construcción de los árboles genealógicos durante del proceso de asesoramiento genético, las limitaciones descritas en la automatización de dicho proceso así como la necesidad de garantizar un mayor aseguramiento en la gestión de los especialistas, se plantea como ***problema científico de la investigación***: ¿Cómo superar las deficiencias en la automatización del estudio de la historia familiar durante el proceso de asesoramiento genético en la genética comunitaria cubana?

El ***objeto de estudio*** ha sido identificado como: los sistemas informáticos para el análisis de la historia familiar en el proceso de asesoramiento genético y el ***campo de***

acción se ha determinado como: los métodos computacionales para la representación y análisis de árboles genealógicos.

Producto del estudio previo realizado en la etapa inicial de la investigación y en correspondencia con el problema identificado se ha definido como **objetivo general** de la presente investigación: Desarrollar métodos computacionales para la representación y análisis de árboles genealógicos que contribuyan a la automatización del proceso de asesoramiento genético en Cuba. Este objetivo general ha sido desglosado en los siguientes **objetivos específicos**:

1. Diseñar los métodos computacionales para la representación y análisis de los árboles genealógicos.
2. Implementar los métodos computacionales identificados y diseñados.
3. Integrar los métodos computacionales implementados en un sistema informático.
4. Validar el sistema informático implementado de acuerdo a los indicadores establecidos en la investigación.

Se ha identificado como **hipótesis** para esta investigación: El desarrollo de métodos computacionales eficientes para la representación y análisis de árboles genealógicos integrados en un sistema informático funcional permitirá elevar el nivel en la automatización del proceso de asesoramiento genético en Cuba.

Las **tareas de la investigación** que a continuación se relacionan, se han determinado para dar cumplimiento a los objetivos planteados:

1. Análisis de los datos genealógicos presentes en la historia familiar para el diseño de la estructura de datos.
2. Identificación de los criterios estéticos para la representación y análisis de árboles genealógicos.
3. Análisis de los sistemas informáticos existentes para la representación y análisis de árboles genealógicos.
4. Identificación de los métodos computacionales existentes para la representación y análisis de árboles genealógicos.
5. Implementación de la estructura de datos seleccionada.

6. Implementación de los algoritmos identificados.
7. Identificación de los requerimientos funcionales y no funcionales que debe tener el sistema informático para la representación y análisis de árboles genealógicos.
8. Definición de las herramientas y tecnologías para el desarrollo del sistema informático.
9. Definición de la arquitectura del sistema informático.
10. Implementación de los componentes identificados como parte de la arquitectura definida anteriormente.
11. Diseño de las clases e interfaces necesarias para la integración de los métodos computacionales al sistema informático.
12. Implementación de las clases e interfaces definidas para la integración.
13. Validación de los algoritmos implementados.
14. Realización de pruebas funcionales al sistema informático implementado.

El éxito de toda investigación científica está en la solución del problema científico, en alcanzar los objetivos y en la comprobación de la hipótesis y esto depende del acierto que se tenga en la selección del método, los procedimientos y técnicas de investigación [7]. Para el desarrollo de esta investigación particular se utilizará un enfoque sistémico y se utilizarán como *métodos teóricos*:

El análisis y la síntesis: permiten realizar el análisis del proceso de asesoramiento genético y su automatización a partir de la relación que existe entre los elementos que lo conforman; y a su vez, la síntesis se produce sobre la base de los resultados previos del análisis de las diferentes tendencias en la representación y análisis de árboles genealógicos a nivel global.

La modelación es justamente el método mediante el cual se crean abstracciones tales como los modelos de arquitectura, diagramas de clase entre otros artefactos que se generarán con vistas a explicar la solución.

Para la *recogida de la información* acerca del objeto de la investigación se ha definido la utilización de la entrevista y los cuestionarios como técnicas fundamentales para la recogida de información durante el proceso de identificación de los requisitos y la validación respectivamente.

Resultados esperados

- Visualización computacional de árboles genealógicos y su representación en una estructura de datos eficiente.
- Diseño e implementación de algoritmos para el análisis de árboles genealógicos más eficientes computacionalmente.
- Integración de los métodos computacionales desarrollados en un sistema informático de mayor calidad para los usuarios finales.

Novedad y aporte práctico

- Visualización de árboles genealógicos desde diferentes perspectivas, no abordadas en la mayoría de los sistemas de su tipo.
- Métodos y componentes reutilizables para sistemas desarrollados en Java.
- La arquitectura del sistema permitirá centralizar la información y arribar a la toma de decisiones diagnósticas basadas en mayores muestras de población.

El contenido del documento está dividido en tres capítulos. El capítulo 1 titulado “*Revisión bibliográfica del proceso de representación y análisis de árboles genealógicos*”, expone los fundamentos teóricos de la investigación, relacionados fundamentalmente con el proceso de representación de árboles genealógicos y su importancia para los estudios genéticos. Se analizan además las aplicaciones informáticas existentes con este objetivo, identificando sus potencialidades y deficiencias. Se exponen además las principales tendencias en cuanto a los procedimientos que se usan para la representación y análisis de árboles genealógicos. Por su parte el capítulo 2 denominado “*Materiales y métodos para el desarrollo de los métodos computacionales*”, describe las tecnologías, herramientas y aspectos teóricos relevantes para el desarrollo de la investigación, agrupados bajo el título de materiales. En una segunda parte describe cómo se utilizarán los materiales descritos para dar forma a la solución final y que metodología se guiará el desarrollo de la investigación.

El capítulo 3 “*Resultados y discusión*” describe los principales algoritmos resultados de la investigación para la representación y visualización de los árboles genealógicos, según la estructura de datos seleccionada. Así como los métodos u algoritmos que permiten el análisis de árboles genealógicos y la realización de algunas operaciones entre árboles genealógicos. Por otra parte se describe el proceso de validación de los algoritmos descritos y pruebas funcionales al sistema para la evaluación del cumplimiento de los requisitos funcionales y no funcionales del sistema informático implementado como resultado final de la investigación.

1

Revisión bibliográfica del proceso de representación y análisis de árboles genealógicos

En el presente capítulo, se exponen los resultados del análisis del estado del conocimiento según el objeto de estudio determinado en la investigación, relacionados fundamentalmente con el proceso de representación de árboles genealógicos y su importancia para los estudios genéticos. Se analizan además las aplicaciones informáticas existentes con este objetivo, identificando sus potencialidades y deficiencias. Así como la identificación de las principales tendencias para la representación y análisis de árboles genealógicos a través de sistemas informáticos y los métodos y algoritmos computacionales fundamentales que han sido implementados con este objetivo.

1.1 Estandarización de la representación de árboles genealógicos

A principios de la década de los 90 con el auge de la genética como disciplina emergente de la medicina y su madurez como ciencia se reconocen los árboles genealógicos como un marco de trabajo esencial para el estudio de las enfermedades genéticas, su tratamiento, diagnóstico y prevención; se utiliza además para el desarrollo de investigaciones y publicaciones como una herramienta importante para la visualización de los problemas genéticos.

Los especialistas de la rama comenzaron la definición de estándares para la representación gráfica de árboles genealógicos, siendo la más aceptada la “Pedigree Standardization Task Force” (PSTF) de un grupo de investigadores de la Sociedad Nacional de Asesores Genéticos de los Estados Unidos, planteada en 1995, este grupo en la actualidad ha adoptado el nombre de Pedigree Standardization Work Group (PSWG). El estándar planteado reduce las posibilidades de realizar interpretaciones incorrectas de la información médica y genética de los pacientes y sus familiares, además de impulsar la calidad de la atención a los pacientes por profesionales de la genética y facilita la comunicación entre investigadores involucrados en el estudio de la familia.[8]

Con el desarrollo de la tecnología y los sistemas de cómputo en el campo de la medicina, internacionalmente se han desarrollado diversos sistemas para la representación de árboles genealógicos, con el objetivo fundamental de agilizar los procesos de representación gráfica de árboles genealógicos, en su mayoría han tenido en cuenta al menos parcialmente lo establecido por Bennet, et al (1995) para la estandarización de los árboles genealógicos. En los próximos epígrafes se presenta el estudio realizado respecto a los sistemas existentes con este objetivo para evaluar sus funcionalidades y la pertinencia de su aplicación en los estudios genéticos cubanos.

1.2 Herramientas informáticas para la representación y análisis de árboles genealógicos

Numerosas herramientas han sido implementadas en diversos lenguajes y plataformas con el objetivo incrementar la eficiencia en el proceso de diagnóstico de enfermedades genéticas y el estudio de la historia familiar, el propósito de este epígrafe es realizar una valoración de las principales herramientas que existen, siendo divididas en dos grupos: herramientas

web y de escritorio.

1.2.1 Herramientas web para la representación de árboles genealógicos.

El desarrollo alcanzado por Internet ha facilitado la creación de árboles genealógicos en la web, en la actualidad existen varias aplicaciones web que brindan al usuario la posibilidad de realizar esta acción, permitiendo de esta forma crear una gran comunidad en la que pueden invitar a amigos y familiares, para que hagan sus árboles y más tarde unirlos para ver sus coincidencias. Este no es el propósito de los estudios genéticos incluso es preciso señalar “que la información que se obtiene a través del árbol genealógico tiene consecuencias personales, familiares y sociales, es por lo tanto confidencial.” [3]

En otro sentido la notación utilizada por estas aplicaciones web no sigue los estándares para la representación de árboles genealógicos establecidos internacionalmente, estas se dedican fundamentalmente al trabajo con multimedia (fotos, videos, y otros). Algunos de los estudiados con estas características fueron: MiParentela¹ , Myheritage² , Geni³ , Ancestry.com⁴ , Genom⁵ y PlusGenealogía⁶ . En la figura 1.1 se muestra un ejemplo de árbol genealógico construido con Genom. My Family Health Portrait⁷ es de acceso libre y gratuito, usa simbología médica estándar, tiene en cuenta enfermedades, y algunos datos relevantes como origen étnico, peso, talla. Se incluyen datos tabulares y genera el dibujo del árbol. Es una iniciativa del gobierno de los Estados Unidos para el estudio de árboles genealógicos. No incluye símbolos adicionales a los individuos, no tiene análisis de enfermedades. Realiza un reporte básico de los datos genealógicos en forma tabular.

Progeny Free online pedigree app⁸ es la más importante de las estudiadas por su flexibilidad al dibujar el árbol, permite a través de una leyenda personalizar los símbolos de los diferentes individuos, tiene como desventaja que sólo permite exportar la imagen del árbol y que no se puede representar todos los elementos relevantes en un árbol y la información que gestiona de los individuos es muy limitada. Es válido señalar que es gratuito el acceso y la representación de árboles genealógicos. La figura 1.2 muestra el

¹Disponible en <http://www.miparentela.com>

²Disponible en <http://www.myheritage.es>

³Disponible en <http://www.geni.com>

⁴Disponible en <http://www.ancestry.com>

⁵Disponible en <http://www.genoom.com/es>

⁶Disponible en <http://www.plusgenealogia.com/>

⁷Disponible en <https://familyhistory.hhs.gov>

⁸Disponible en <http://www.progenygenetics.com/online-pedigree/>



FIGURA 1.1: Árbol genealógico construido con Genom

área de la aplicación para el dibujo.

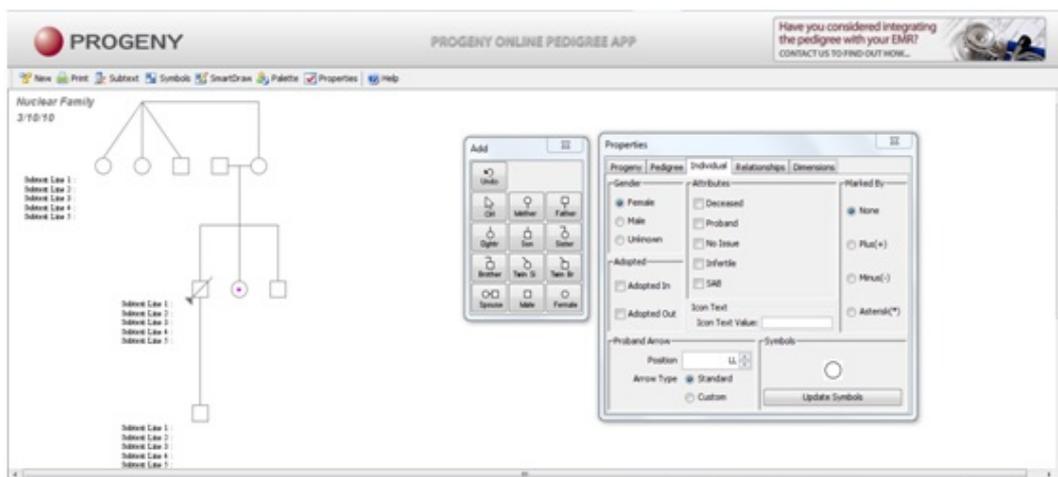


FIGURA 1.2: Pantalla principal de Progeny Free online pedigree app

1.2.2 Herramientas de escritorio para la representación de árboles genealógicos

Las herramientas de escritorio son las más comunes con este objetivo debido a la facilidad de manejo que ofrecen al usuario, su portabilidad y el propio desarrollo tecnológico, que

ha favorecido a este tipo de herramientas en cuanto a sus facilidades visuales a la hora de implementar componentes gráficos personalizados, que permitan “arrastrar y dibujar” componentes visuales en tiempo real.

Algunas de las herramientas estudiadas se exponen en lo adelante, se mencionarán las propietarias más importantes que existen, las cuales tienen costos por copia inaccesibles para nuestro país tales como Progeny Clinical y el Cyrillic, que son lo más reconocidos internacionalmente y tienen un costo de 900 y 770 USD respectivamente por copia.

Cyrillic posee todas las características necesarias para dibujar líneas genéticas. Es el software diseñado para el manejo de datos genéticos más completo, funcional y fácil de usar. Todas las herramientas están dispuestas de forma que usted puede, rápida y fácilmente, manejar los datos de los que disponga. Puede almacenar un elevado rango de datos para cada individuo, incluyendo información personal, clínica y genética. Como características importantes presenta:

- Actualización automática de los datos mientras se está dibujando.
- Perfeccionamiento del manejo para casos de gemelos y embarazos múltiples.
- Cálculo de riesgos para enfermedades familiares, permitiendo hacer el cálculo de riesgo de enfermedad para un individuo dentro de una familia donde exista una determinada enfermedad, basado en modelos propietarios que no son accesibles.
- Utilización de bases de datos como Access o Corel Paradox.

Este sistema en su versión 3.0 [9] incluye como funcionalidad importante el análisis de riesgo de padecimiento de cáncer de origen genético, tales como el cáncer de mama y próstata. Tiene una excelente organización del espacio de trabajo y de los individuos según su generación, su mayor desventaja unido a los altos precios de la licencia es su diseño para Windows únicamente.

Progeny Clinical es utilizado desde 1996 en instituciones de investigación o de servicios de genética clínica en el mundo, para representar todo tipo de árboles genealógicos de humanos o animales. Todas las opciones de dibujo están implementadas conforme a lo planteado por PSTF siempre que es posible. Tiene un elevado nivel en el manejo de datos genealógicos pues combina el poder de una herramienta para la representación de árboles genealógicos y una base de datos. Entre sus principales características se pueden destacar[10]:

- Gestión y representación de complejos árboles genealógicos.

REVISIÓN BIBLIOGRÁFICA

- Flexibilidad con el uso de iconos y subtextos.
- Edición de relaciones y adición de múltiples esposos(as) y gemelos
- Salva el árbol genealógico como jpg, tiff, bmp, png, o gif
- Permite crear sub-árboles para el análisis de individuos en particular
- Representación de haplotipos
- Crea reportes personalizados sobre el árbol.

La figura 1.3 muestra un árbol genealógico construido con Progeny, en el que se puede observar un adecuado uso de la simbología, el uso de leyenda para aclarar el propósito de la simbología seleccionada y como se planteaba anteriormente el uso de las recomendaciones de PSTF en el esquema.

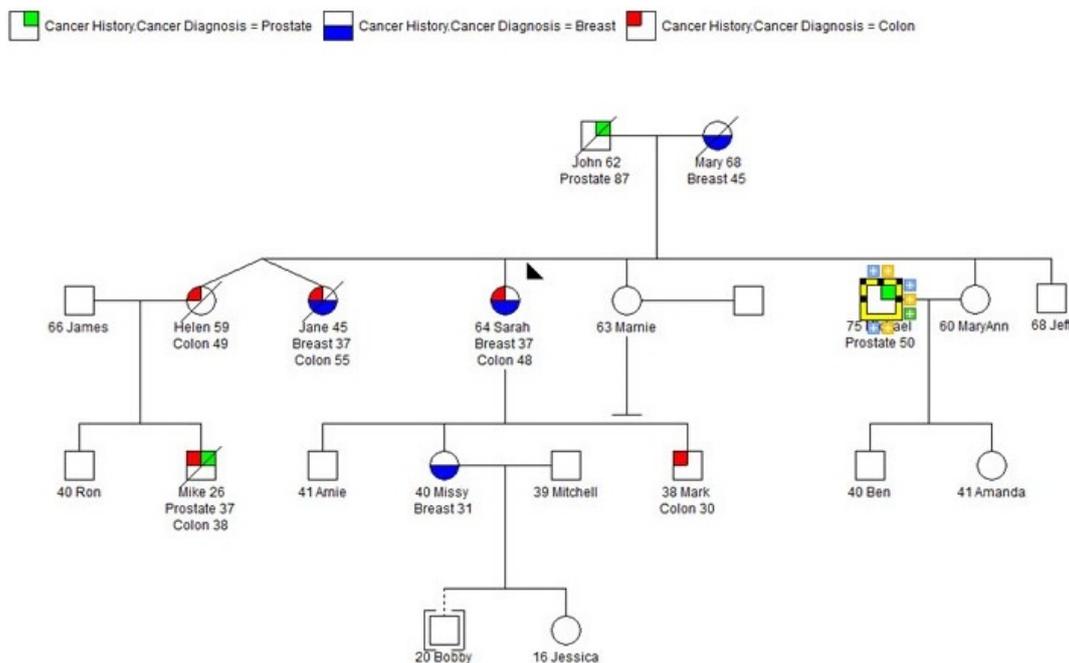


FIGURA 1.3: Árbol genealógico dibujado con Progeny Clinical

Como parte del estudio realizado se identificaron cuatro herramientas libres para la representación de árboles genealógicos que implementan algunos de los elementos de PSTF,

las mismas son: Cranefoot[11], Haplopainter [12], Kinship[13] and PedigreeQuery[14]. Todas presentan limitaciones al dibujar árboles genealógicos complejos. [15]

Madeline 2.0 PDE es un sistema de Licencia GPL de código abierto, desarrollado en C++ que a través de la línea de comandos genera el árbol genealógico basado en las relaciones entre los individuos, permite la representación de consanguinidad en las relaciones y esto hace que a veces se representen ciclos en el árbol genealógico. Actualmente está en desarrollo un servicio web para el acceso desde otros sistemas que permitirá la visualización en el navegador de un árbol genealógico. Usa una simbología estándar, implementando la mayoría de los estándares establecidos por PSTF. Maneja su información a través de formatos como XHTML, OASIS, Office Open XML, Madeline XML y, opcionalmente, archivos en formato MySQL desde la computadora local o de una ubicación externa usando el protocolo de red HTTP/S.[15]

Haplopainter es un sistema libre para Windows que permite un fuerte trabajo con los haplotipos, permitiendo el análisis de múltiples marcadores en el árbol genealógico, pero en otro sentido el sistema es muy ligero, y carece de facilidades para el trabajo gráfico, no presenta un manejo de símbolos u otros elementos estándares. Se centra fundamentalmente en cargar árboles desde diversos formatos tales como CSV y Linkage, o desde sistemas de bases de datos en PostgreSQL, MySQL u Oracle. Importa además los haplotipos desde base de datos reconocidas tales como Simwalk, GeneHunter, Merlin y Allegro. Exporta los árboles en varios formatos, entre ellos CSV y PNG.[12]

Vale destacar además que **Cranefoot** y **Kinship package in R** tienen menos funcionalidades, pero están en desarrollo, todos son para uso de médicos e investigadores y usan parte del estándar PSTF, otra de sus desventajas es que permiten la construcción del árbol solo a partir de la línea de comandos como se muestra en la figura 1.4.

1.3 Representación de árboles genealógicos

Los programadores reconocen a los árboles como una estructura de datos extremadamente común. Los árboles modelan muchos problemas reales y se caracterizan por su eficiencia y utilidad. Igualmente importante, una buena visualización de un árbol es una intuitiva y poderosa herramienta en la modelación de problemas. Por ello Wetherell y Shanon en 1979 publican la implementación de algoritmos para visualizar árboles ordenados. Definiendo un árbol como: “Un árbol es formalmente un grafo finito, dirigido, conexo y aciclico, en el que cada nodo tiene a lo sumo un predecesor denominado raíz”. [16] Como elemento positivo estos autores identificaron tres aspectos estéticos fundamentales sobre los cuales

```

# File instructions:
PedigreeFile      (string)
PedigreeName      (string)

# Structural instructions:
SubgraphVariable  (string)
NameVariable      (string)
FatherVariable    (string)
MotherVariable    (string)
GenderVariable    (string)
AgeVariable       (string)

# Visualization instructions:
ArrowVariable     (string)
ColorVariable     (string)
PatternVariable   (string)
ShapeVariable     (string)
SlashVariable     (string)
TextVariable      (string)
TracerVariable    (string)
# label          code
ColorInfo         (string) (integer) # multiple
PatternInfo       (string) (integer) # multiple
ShapeInfo         (string) (integer) # multiple

```

FIGURA 1.4: Vista de Cranefoot en consola

implementaron sus algoritmos para visualizar un árbol ordenado, los mismos son [16]:

1. Los nodos del árbol de igual peso deben ser dibujados a lo largo de una línea recta, y las líneas rectas que definen los niveles deben ser paralelas.
2. En un árbol binario, cada hijo izquierdo debe ser ubicado a la izquierda del padre y cada hijo derecho a la derecha del padre.
3. El ancestro debe estar centrado sobre los hijos.

Walker (1990) plantea que el dibujo de un árbol consta de dos etapas: la determinación de la posición de cada nodo(y su representación) y la interconexión de los individuos [17]. Plantea además que los algoritmos planteados por Wetherell y Shannon [16] y Tilford [18] fallan al ubicar los nodos interiores del árbol correctamente. El algoritmo de Walker organiza correctamente las posiciones de los nodos en dos iteraciones, teniendo en cuenta dos consideraciones prácticas: la orientación alternativa de los nodos del árbol, tamaños variables de los nodos y las condiciones de fuera de rango. Radack también construyó a partir del trabajo de Tilford una solución con un algoritmo diferente en cuatro iteraciones [19].

Kennedy (1996) plantea otra solución en Pearl para la organización de árboles genealógicos y define el problema como: “dado un árbol etiquetado, asignar a cada nodo de una posición en la página para dar una representación estética del árbol. Suponemos que

los nodos de la misma profundidad se colocan en la misma línea horizontal en la página, por lo que el problema se reduce a la búsqueda de una posición horizontal para cada nodo”[20]. Este autor reconoce los mismos criterios estéticos planteados anteriormente por Walker y Radack.

En el año 2005 Mäkinen y otros indentifican una serie de reglas básicas para una visualización óptima de árboles ordenados [11] que agregan a las identificadas por Wetherell y Shanon las siguientes:

1. Los nodos no deben superponerse.
2. Las líneas rectas de los hijos a sus respectivos padres no se deben cruzar.
3. (A) Un subárbol debe ser elaborado de la misma manera, independientemente de su posición. (B) Después de invertir el orden de un nodo, el nuevo dibujo debe ser un reflejo del original.

Pero en 2007 Stein y Benteler [21]realizan un estudio profundo del problema de la representación de árboles y especifican algunos aspectos estéticos adicionales a los anteriormente planteados, ajustando el problema a nodos de tamaño uniformes y en árboles n-arios. Realizando un profundo análisis inicial acerca de los criterios precedentes a su investigación y agregan a los criterios anteriormente expuestos para árboles con nodos de tamaño uniforme:

- Los nodos padres estarán centrados respecto a su hijo más derecho y más izquierdo.
- La distancia horizontal entre los nodos del mismo nivel tiene una variación mínima mientras cumpla con la condición anterior y con la propiedad de reflexión que debe tener el árbol.

Luego del análisis completo de los algoritmos precedentes, la propuesta de Stein y Benteler consiste en un meta algoritmo que plantea como sus pasos fundamentales [21]:

1. Definición del tamaño del área de dibujo.
2. Reformulación de la topología.
3. Estructuración del esquema.
4. Mapeo reverso de la topología.

5. Dibujar el esquema definido.

Dado el análisis de los principales algoritmos y la definición de los aspectos estéticos fundamentales para la representación gráfica ordenada de árboles genealógicos se definirán algoritmos que teniendo en cuenta estos criterios permitan una adecuada representación de los árboles en la herramienta a implementar adicionando algunas restricciones y adecuaciones, debido a que los árboles genealógicos son más ricos en relaciones que los árboles n-arios, la aparición de ciclos debido a las relaciones introducen otras restricciones relevantes.

1.4 Análisis de árboles genealógicos

Desde los inicios de la genética y el estudio de los pacientes a través de su historia familiar como elemento esencial y primario en el proceso de asesoramiento genético, los especialistas dibujan el árbol genealógico y luego analizan según los esquemas de herencia y las relaciones entre los individuos las posibles consecuencias médicas que de derivan de este análisis. En una primera aproximación desde la informática aplicada se ha logrado la implementación de sistemas que garanticen la representación gráfica de los árboles, en particular alasARBOGEN en su primera versión, a pesar de las limitaciones descritas garantizaba un sistema propio para el desarrollo de investigaciones y el compartimiento de casos en la Red Nacional de Genética, pero no incluye ningún algoritmo que permitiera el análisis de los árboles.

Planteados anteriormente los elementos esenciales a tener en cuenta para la representación de los árboles como parte fundamental para su posterior análisis, se precisa entonces el establecer los principales criterios para la predicción de riesgos, la búsqueda de patrones, el comportamiento de enfermedades según su expresión entre los familiares de un árbol genealógico entre otros.

Los principales análisis que se pueden realizar de la representación de árboles genealógicos, según la revisión de la literatura especializada en la rama son: comportamiento de enfermedades genéticas, tipos de herencia, patrones de herencia, análisis de fenotipos, análisis de alelos y análisis de haplotipos.

En este sentido varios programas han incluido algoritmos para el análisis de árboles genealógicos, los primeros basados fundamentalmente en probabilidades, en 1979 Elston y Sobel publican un algoritmo para la obtención de una expresión general para la hallar la probabilidad de un conjunto de observaciones fenotípicas en una muestra aleatoria

de un árbol genealógico [22]. Otros como sistemas como GENEHUNTER combinan el análisis de ligamiento de los caracteres cuantitativos y métodos de varianza que permiten la consideración simultánea de todos los miembros del árbol genealógico [23].

El Cyrillic en su última versión incluye un completo análisis de riesgos, basado en modelos para la identificación de familiares con riesgo de padecer enfermedades según su genealogía e incluye el tratamiento de marcadores genéticos [9].

La automatización de este proceso de análisis de los árboles genealógicos tiene particular importancia en la actualidad, debido al surgimiento de enfermedades raras y la aparición de múltiples factores que influyen en la expresión de los genes, debido a factores medioambientales u otros. Para lograr un mayor desarrollo de las investigaciones en la rama en Cuba, es preciso incorporar a los sistemas algoritmos para el análisis de las relaciones en los árboles que faciliten el proceso y unifiquen criterios para el diagnóstico y prevención de este tipo de enfermedades u otras más conocidas.

En la actualidad las principales tendencias en la informática para este tipo de análisis se basan en la implementación de sistemas basados en casos o en reglas que contribuyen, a partir de bases de conocimiento y utilizando técnicas de inteligencia artificial arribar a conclusiones en menor tiempo, además de permitir la formalización del conocimiento de los especialistas. Por otra parte la expresión de los genes y el comportamiento de las enfermedades tienen una alta dependencia del entorno en que se desarrollan los individuos. Las reglas o casos de externos no pueden ser aplicados en todos los entornos, es válido entonces la implementación de sistemas propios, que valoren los factores particulares de Cuba.

Conclusiones parciales

Al concluir este capítulo se han identificado los elementos teóricos relacionados con el objeto y campo de acción de la investigación. El estudio detallado de los sistemas mencionados constituye una base esencial para el objetivo de la investigación, las tendencias para este tipo de sistemas y el estudio del estándar definido por PSTF permiten la determinación correcta de las funcionalidades para una nueva versión del sistema para la representación de árboles genealógicos alasARBOGEN, con el objetivo de incrementar su papel en el proceso de asesoramiento genético y en los estudios que se llevan a cabo por parte del Centro Nacional de Genética Médica.

Ninguno de los sistemas identificados en el estudio se pueden adaptar o implantar en

REVISIÓN BIBLIOGRÁFICA

la Red de Genética actual sin limitaciones, las principales desventajas de manera general lo constituyen:

- Madeline 2.0 PDE a pesar de ser de código abierto está diseñado para ser usado a través de la línea de comandos lo que es inconveniente en cuanto al nivel de usabilidad que impondría.
- Los más desarrollados únicamente pueden ser utilizados en Windows, aspecto que va en contra de las políticas de migración planteadas por el Ministerio de la Informática y las Comunicaciones (MIC) en Cuba.
- Aunque cumplen en alguna medida con el estándar PSTF aún requieren agregar otros elementos del estándar.
- No es posible integrarlos a Infomed.

Como elementos positivos se han determinado nuevas funcionalidades ajenas al negocio propiamente y que son parte de la mayoría de los sistemas, tales como:

- Manejo de datos genealógicos siguiendo estándares tales como GEDCOM, CSV, entre otros.
- Soporte de base de datos.
- Manejo de haplotipos y genotipos.
- Inclusión de elementos de PSTF tales como consanguinidad, y la construcción de árboles cíclicos.
- Seguridad en el manejo de los datos genealógicos.
- Integración a otros sistemas a través de servicios web.

Es fundamental además la definición de los aspectos estéticos a seguir para la representación de árboles genealógicos establecidos por una serie de investigadores que desde 1979 trabajan desde la computación este tema como se expone en la sección 1.3, así como el estudio de los principales algoritmos para el dibujo de árboles ordenados y los principales pasos para la solución de este problema.

2

Materiales y métodos para el desarrollo de los métodos computacionales

El presente capítulo describe las tecnologías, herramientas y posiciones teóricas que asume el autor relevantes para el desarrollo de la investigación, agrupados bajo el título de materiales. En una segunda parte se describe cómo se utilizarán los materiales descritos para dar forma a la solución final. Se establecen los métodos a utilizar para guiar el desarrollo de la investigación, especificando la metodología de desarrollo y los procedimientos de pruebas para la validación.

Materiales

2.1 Estructura de dato grafo

Tores y Barillot (2001) definen el concepto de árbol genealógico perfectamente dibujable (PDP por sus siglas en inglés) y demuestran la existencia de una solución perfecta que puede ser probada y dibujada en tiempo lineal si el árbol es PDP. Cuando el árbol no

es PDP, la solución no es trivial [24]. El enfoque usado por Mäkinen et al. (2005) define la transformación del árbol genealógico en un grafo acíclico.[11]

Los árboles genealógicos pueden ser divididos en tres categorías: acíclico rooted graphs, cíclico pero PDP, y grafos cíclicos no dibujables. La visualización de un árbol genealógico depende de qué tan bien un determinado conjunto de objetivos estéticos pueden ser atendidas por una presentación de dos dimensiones.

Un grafo matemáticamente es representado como $G(V;E)$ donde V es un conjunto finito y no vacío de vértices. E es un conjunto de pares de vértices, dichos pares se denominan aristas [25]. Los grafos permiten la representación de relaciones múltiples entre los diferentes vértices que lo componen y por este motivo han sido ampliamente aplicados en diferentes ramas de la ciencia que requieren la modelación de redes y la representación de relaciones. Tal es el caso de la genética que requiere un fuerte análisis de las relaciones entre las personas en el estudio de las relaciones familiares.

Los grafos como Tipo de Dato Abstracto tienen varias operaciones básicas para la gestión de los datos que maneja, entre los que se pueden mencionar: insertar/eliminar vértice, insertar/eliminar arco, vértices adyacentes, entre otros más complejos que son ampliamente difundidos y por lo tanto reutilizables, lo que favorece el proceso de implementación y diseño de los algoritmos.

2.2 Plataforma Java

Java es una de las plataformas más completas para la construcción de aplicaciones en la actualidad, según su sitio oficial “Hasta la fecha, la plataforma Java ha atraído a más de 6,5 millones de desarrolladores de software. Se utiliza en los principales sectores de la industria de todo el mundo y está presente en un gran número de dispositivos, equipos y redes.” [26] La versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para su aplicación a redes.

La ventaja fundamental de utilizar un lenguaje como Java que ha sido probado, mejorado y ampliado continuamente es el hecho de desarrollar software en una plataforma y ejecutarlo en prácticamente cualquier otra plataforma. Además de integrar muchos frameworks y tecnologías que permiten la construcción de aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados con múltiples propósitos.

En lo adelante se describirán algunas de las funcionalidades, paquetes o tecnologías que asociadas a Java como plataforma y lenguaje de programación se han definido para la implementación de los resultados de la investigación.

2.2.1 Eventos visuales

Un evento es la forma en que una clase notifica a los usuarios de un objeto cuando alguna acción es realizada sobre ese objeto. Esta notificación es llamada lanzar un evento. El objeto que lanza el evento se conoce como el publicador del evento.[27] Los eventos visuales en java son ejecutados a través del uso del paquete `java.awt.event` que brinda las interfaces y clases para el manejo de diferentes tipos de eventos capturados por componentes del Kit de Herramientas de Ventana Abstracta (del inglés Abstract Window Toolkit, AWT).[28]

Los eventos visuales en particular son vitales para la interacción con el usuario a través de componentes visuales, su captura y manejo garantizan en un primer nivel la representación gráfica de los árboles genealógicos en un ambiente visual y personalizable para los usuarios del sistema.

2.2.2 Anotaciones

Una anotación es un tipo especial de modificador, y puede ser utilizado en cualquier lugar que otros modificadores (como `public`, `static` o `final`) se puede utilizar. Por convención, las anotaciones preceden a otros modificadores. Consisten en un signo de arroba (@) seguido de un tipo de anotación y una lista entre paréntesis de pares de valores del elemento. Los valores deben ser constantes en tiempo de compilación.[29] La definición de anotaciones propias para el sistema contribuirá al desacoplamiento del sistema y permitirán el acceso a informaciones de los objetos en tiempo de ejecución, a continuación se expone un ejemplo usado en la clase `persona` para la muestra de sus propiedades en tiempo de ejecución, incluyendo incluso otras cuestiones como expresiones regulares para la validación de los campos.

Las anotaciones serán usadas además para la persistencia a través del uso de Java Persistence API, para evitar la definición de ficheros `hbm` (ficheros usados por Hibernate para la persistencia) para cada una de las tablas de la base de datos.

2.2.3 Reflexión

El API de java denominado `java.lang.reflect` garantiza las clases y funcionalidades para el uso de esta avanzada técnica de la programación en Java. Reflexión permite el trabajo con los elementos internos de las clases aunque no hayan sido construidas por el propio desarrollador. El código siguiente, tomado como ejemplo muestra como a través de la funcionalidad “`getDeclaredFields()`” de la clase `java.lang.Class` se obtienen todos los atributos de la clase `Persona`, de esta forma se puede acceder a otros elementos de la clase, tales como las anotaciones, si es pública o privada e incluso cuál es su superclase.

```
1 atributos = Persona.getDeclaredFields();
```

El uso de esta técnica permite un amplio desacoplamiento de las algunas funcionalidades y la implementación de componentes que puedan ser reutilizados en otros entornos. No obstante, el uso de esta técnica requiere un uso cauteloso y no indiscriminado, debido a que este trabajo del JDK en tiempo de ejecución eleva el consumo de memoria del sistema.[30]

2.3 Entorno y herramientas de desarrollo

2.3.1 IDE Eclipse

Eclipse es un entorno de desarrollo integrado (IDE) de código abierto y multiplataforma. Su fortaleza radica en su grado de extensibilidad. Eclipse es una superestructura formada por un núcleo y diversos plugins que van conformando la funcionalidad final y soporta todo el ciclo de desarrollo. [31]

Algunos de los plugins más importantes que se han identificado para el entorno de desarrollo de la investigación son: Visual Editor, Subclipse para el control de versiones, Hibernate Tools y JPA Tools para la persistencia de los datos, SDE for Eclipse para la generación de código y actualización del diseño, Spring para la inversión del control a través de inyección de dependencia entre otros que se requieren para el desarrollo en un entorno Java Enterprise Edition. Presenta un potente editor de código que permite realizar completamiento del mismo, incluso del código contenido en las librerías externas agregadas.”

2.3.2 Hibernate

Hibernate es un entorno de trabajo que tiene como objetivo facilitar la persistencia de los datos a través de base de datos relacionales y a la vez la consulta de bases de datos para obtener objetos. Es una herramienta de asociación entre modelos jerárquicos de objetos y esquemas relacionales de bases de datos, lo que se denomina comúnmente como una herramienta de Mapeo Relacional de Objetos (ORM).[32]

2.3.3 Spring

Spring es un framework de aplicación de desarrollado por la compañía Interface 21, para aplicaciones escritas en el lenguaje de programación Java. Es considerado un framework liviano y ligero, porque que no requiere de muchos recursos para su ejecución.

Un aspecto a destacar en la arquitectura del framework Spring es el sub-framework Contenedor de Inversión de Control (IoC), conocido también como Inyección de Dependencias. En lugar de que el código de la aplicación llame a una clase de una librería, un framework que utiliza IoC llama al código. Es por esto que se le llama “Inversión”, ya que invierte la acción de llamada a alguna librería externa. Este proceso posibilita que las dependencias entre objetos interrelacionados sean mínimas, lo que permite que Spring pueda ser removido sin invocar grandes cambios en el código de la aplicación.

En el núcleo contenedor de Spring se encuentran las funcionalidades básicas de este framework. Cuenta con una factoría Beans que constituye el centro de toda aplicación basada en Spring, la cual es implementada bajo el patrón Factory y que aplica técnicas de IoC para separar la configuración de la aplicación y las dependencias del código de la aplicación.

2.3.4 Drools

Drools es un sistema de gestión de reglas de negocio en java, que implementa el API del motor de reglas de java y utiliza una implementación mejorada de los algoritmos para la ejecución de las reglas. Es lo suficientemente flexible como para que coincida con la semántica del dominio de diversos problemas. El mismo está compuesto por “Drools Expert” como motor de reglas empresariales, que permite desacoplar reglas de negocio de nuestras aplicaciones y utilizar su motor de inferencia, que utiliza para la deducción

encadenamiento hacia adelante (forward chaining) y una extensión del Algoritmo RETE (llamada RETEEO). Siendo este su componente más relevante para esta investigación. Además se pueden mencionar “Drools RuleFlow” para la definición de procesos de negocio, “Drools Fusion” para el procesamiento de eventos complejos y permite que las reglas de negocio puedan realizar deducciones basadas en datos temporales y “Drools Guvnor” que es un sistema de administración de reglas que permite centralizar todos los recursos de reglas.

2.4 Arquitectura y Tecnologías

La arquitectura es la disciplina que cuenta con las herramientas adecuadas para definir un sistema desde un nivel abstracto, identificando los aspectos estructurales y funcionales más importantes para su desarrollo, así como los aspectos relevantes de la tecnología a utilizar y el ambiente de desarrollo.

Formalmente el Institute of Electrical and Electronics Engineers (IEEE), define la arquitectura de software como: la organización de los sistemas y sus componentes, las relaciones entre ellos y el medio, y los principios que guían su diseño y evolución.[33]

Esta práctica, recomendada por la IEEE identifica a través de un modelo conceptual los factores que inciden en la descripción de la arquitectura para un sistema, entre los que es preciso destacar: el modelo para la descripción con sus respectivas vistas, la misión del sistema según el entorno y los stakeholders [33]. Estos factores serán abordados con el objetivo de lograr una descripción de la arquitectura basada en buenas prácticas que tiene como centro de desarrollo la misión del sistema y su pertinencia de acuerdo a las preocupaciones de los beneficiados con el mismo.

2.4.1 Arquitectura cliente-servidor

La arquitectura cliente-servidor como su nombre lo indica tiene un cliente que realiza peticiones a otro programa (el servidor). En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. [34] La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no

se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico. La implementación de sistemas destinados a Infomed deben basarse en esta arquitectura, debido a que el Registro Informatizado de Salud definen su topología como: “Centralizada, en el almacenamiento de la informaciónz “distribuida, en la manipulación de la información”.[35]

2.4.2 Estilos arquitectónicos

Mary Shaw y Paul Clements conceptualizan los estilos arquitectónicos como un “conjunto de reglas de diseño que identifican clases de componentes y conectores que se pueden manejar para componer el sistema, junto con las restricciones locales o globales que determinan como se lleva a cabo la composición”.[36] Cada estilo describe una categoría del sistema que contiene:

- Un conjunto de componentes que realizan una función requerida por el sistema.
- Un conjunto de conectores que posibilitan la comunicación, la coordinación y la cooperación entre los componentes.
- Restricciones que definen cómo se pueden integrar los componentes que forman el sistema.
- Modelos semánticos que permiten al diseñador entender las propiedades globales de un sistema para analizar las propiedades conocidas de sus partes constituyentes.

En la presente investigación se adopta la clasificación propuesta por “Shaw y Garlan” quienes definen una clasificación de estilos, de los que se mencionan a continuación los más relevantes para esta investigación [37]: Estilo de llamada y retorno:

- Modelo-Vista-Controlador.
- Arquitectura basada en Componentes.

Estilo Peer - To - Peer:

- Arquitecturas basadas en Eventos.

- Arquitecturas Orientadas a Servicios (SOA).

Para la definición de la arquitectura del sistema en particular se utilizarán enfoques de los estilos mencionados prevaleciendo el Modelo-Vista-Controlador y la Arquitectura basada en Componentes, con el propósito de lograr un sistema que cumpla con los requisitos de calidad que se identifiquen como críticos para la arquitectura, teniendo en cuenta las restricciones del negocio y el entorno físico para su funcionamiento.

Métodos

2.5 Metodología para la representación de árboles genealógicos

La representación de los árboles genealógicos se realiza según la forma establecida por el PSWG, que establece el dibujo del árbol organizado jerárquicamente según las relaciones que se establecen, plantea la simbología adecuada para la representación de aspectos significativos relacionados con su información genética o enfermedades, así como los diferentes tipos de relaciones que deben ser representadas [38].

La figura 2.1 muestra un ejemplo de árbol construido según lo establecido para la estandarización de los árboles por el PSWG. La representación se realizará por parte de los especialistas en el sistema informático una vez implementado, para garantizar esto en lo adelante se describen como serán manejados los eventos visuales que se provocan al interactuar con el sistema, así como las características que tendrán los componentes visuales para este propósito.

2.5.1 Manejo de eventos visuales

Los eventos visuales capturados por la capa de presentación serán fundamentalmente eventos generados por el mouse, teclas o combinaciones de teclas, para elevar el nivel de accesibilidad a las funciones del sistema. Los eventos generados en la capa de presentación invocarán procedimientos de la capa de negocio o visuales y se mostrarán los resultados a los usuarios. Los eventos garantizan la interacción de los usuarios con el sistema como muestra la figura 2.2.

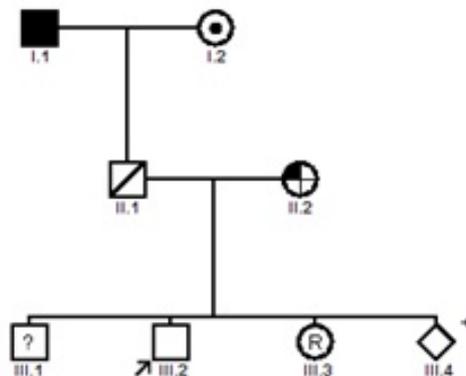


FIGURA 2.1: Árbol genealógico construido según lo planteado por PSWG

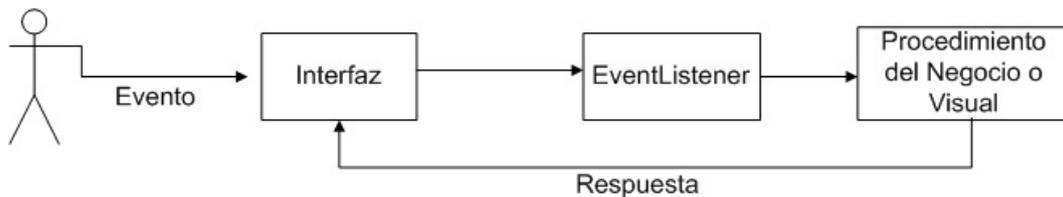


FIGURA 2.2: Esquema para el manejo de los eventos visuales

2.5.2 Visualización y edición de los datos personales

La visualización del árbol genealógico se realizará a través de paneles visuales (de la clase `javax.swing.JPanel`) según las formas definidas para cada individuo y se dibujarán líneas entre los mismos según las relaciones que se crean.

La visualización de los datos de las diferentes entidades se logra haciendo una combinación de dos potentes API de Java, como es el caso de Reflexión y Anotaciones. Tal combinación permite la carga y transformación de los datos que tienen las entidades. En las entidades persistentes del sistema, la información se encuentra anotada, con la anotación `@PropertyField`. Esta anotación denota que estas propiedades van a ser cargadas o transformadas por un componente con interfaces bien definidas, orientado a la carga de datos genéricos.

La edición de los datos personales se realizará a través de un visor de propiedades de los objetos, según el que esté seleccionado en el área del árbol, permitiendo un acceso

rápido a los datos de los individuos, se implementará además una vista tabular para la visualización simultánea de los datos de varios individuos del árbol.

2.6 Análisis sobre árboles genealógicos

En la sección 1.4 se definen las principales tendencias para el análisis de árboles genealógicos, se requiere entonces definir los mecanismos que se utilizarán en la investigación con este objetivo. Los que se materializan a través de la definición de algoritmos sobre grafos y la utilización de sistemas basados en reglas como se explica a continuación.

2.6.1 Algoritmos sobre grafo

Al definir la estructura de datos para modelar el árbol como un grafo, se precisa la definición de algoritmos y la utilización de algunos ya existentes para el análisis de árboles genealógicos. Como el objetivo del grafo es la representación de numerosas relaciones, estamos en presencia de un grafo conceptualmente diferente porque pueden existir varias relaciones o aristas entre un mismo par de vértices. Por lo que en una primera adaptación de la estructura de datos es necesario redefinir algunos de los procedimientos que tiene como Tipo de Dato Abstracto (TDA) y adaptar la representación de Lista de Vértices - Lista de Adyacencia a las condiciones del problema.

Teniendo en cuenta que el sistema tiene el objetivo de centralizar la información de los árboles genealógicos es relevante entonces implementar algoritmos que permitan unir o separar árboles genealógicos según la enfermedad que se estudie o la cantidad de niveles relevantes en un estudio determinado. Algoritmos que además permitan ordenar el árbol correctamente para favorecer el análisis, teniendo en cuenta la amplia gama de relaciones que se representan.

Para el reconocimiento de patrones de herencia o del comportamiento de una enfermedad específica también es preciso plantear procedimientos que identifiquen estos patrones, pudiendo ser combinados en una segunda aproximación al resultado de la inferencia de una serie de reglas, que pueden definirse para el sistema como se describe en el próximo subepígrafe.

2.6.2 Sistemas expertos basados en reglas

Los Sistemas Expertos (SE) son una rama de la Inteligencia Artificial que hace un amplio uso del conocimiento especializado para resolver problemas como un especialista humano. Es decir el especialista tiene conocimientos o habilidades especiales que la mayoría no conoce o de las que no dispone; puede resolver problemas que la mayoría no podría resolver, o los resuelve con mayor eficiencia. Un SE es un Sistemas Basados en Conocimiento (SBC) al que se le incorpora conocimiento proveniente de expertos en dicho dominio. La finalidad de un SE es la resolución de problemas del dominio para el que ha sido creado, aplicando técnicas de razonamiento sobre el conocimiento que alberga su base de conocimiento.

La estimación del riesgo o cálculo de riesgo de padecimiento de enfermedades de origen genético es competencia del médico especialista y se basa en la evaluación de la historia personal y familiar de cada paciente, donde se establecen estrategias de seguimiento y se implementan medidas de prevención, disminuyendo así la mortalidad por causa de la herencia de dichas enfermedades. Entonces lo que se propone es incorporar paulatinamente ese conocimiento a un sistema automatizado que contribuya a la toma de decisiones diagnósticas y con ello a la prevención de enfermedades.

En Cuba se realizan numerosos estudios de enfermedades en algunas sectores poblaciones de alta incidencia de las mismas, como inicialmente no existen fuentes de datos formales de las que se puedan inferir conclusiones acerca del comportamiento de las enfermedades. Se implementará un editor de reglas basado en la definición de variables que se definen de riesgo para la presencia de una enfermedad determinada. La inferencia del resultado del análisis de las reglas definidas permitirán a los especialistas avanzar en sus estudios y compartirlos de una manera más formal, contribuyendo al desarrollo de la rama en Cuba. Las reglas como forma de representación del conocimiento fomentarán el establecimiento de condiciones y de factores que pueden ser ajustados en otros entornos y tienen un valor significativo.

En futuros desarrollos es recomendable la implementación de un sistema basado en casos, cuando se cuente con una base de datos considerable que permita arribar a conclusiones más certeras.

2.7 Metodología de software

Debido a las características del desarrollo de la investigación que consisten fundamentalmente en la presencia de los interesados como parte del equipo y pocos miembros para

el desarrollo del proceso, se selecciona una metodología ágil como OpenUP que dirige la organización del trabajo en los niveles personal, de equipo y de interesados e identifica las actividades básicas para el desarrollo de un proyecto. OpenUP es un proceso mínimo que aplica los enfoques iterativos e incremental dentro de un ciclo de vida estructurado. OpenUP abarca una pragmática y ágil filosofía que está enfocada en la naturaleza colaborativa del desarrollo de software [39]. El OpenUP tiene elementos que ayudan a los equipos de trabajo a enfocar los esfuerzos a través del ciclo de vida de cada iteración de tal forma que se puedan distribuir funcionalidades incrementales de una manera predecible, una versión totalmente probada y funcional al final de cada iteración.

El OpenUP estructura el ciclo de vida de un proyecto en cuatro fases: concepción, elaboración, construcción y transición. El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y dirección para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos. Por lo tanto no provee lineamientos para todos los elementos que se manejan en un proyecto pero tiene los componentes básicos que pueden servir de base a procesos específicos. En función de esto se realizarán como tareas fundamentales para el proceso de desarrollo, según las definidas por OpenUP, basadas en el enfoque iterativo e incremental las siguientes:

- Identificación y refinamiento de los requisitos.
- Definición de la arquitectura.
- Elaboración de los artefactos del diseño.
- Implementación.
- Integración de los componentes.
- Pruebas.

En lo adelante se describen detalladamente algunos de los aspectos más relevantes en los que se basarán las iteraciones a desarrollar durante el desarrollo de la investigación.

2.7.1 Requisitos funcionales y no funcionales

La identificación de los requisitos funcionales y no funcionales constituyen el paso más importante en el proceso de desarrollo de software. De su definición correcta se determina la arquitectura, el plan de desarrollo y la complejidad en general del proyecto. La definición

de los requisitos funcionales del sistema a implementar son determinados a través de entrevistas a los especialistas del CNGM y el estudio de los sistemas que existen con este objetivo a nivel internacional como se manifiesta en la sección 1.2, para lograr un sistema que además de resolver los problemas de Cuba, se convierta en una solución competitiva en su ámbito.

Por su parte los requisitos no funcionales se determinarán además teniendo en cuenta las características de la Red Nacional de Genética, la seguridad y confidencialidad que requiere este tipo de sistema por la sensibilidad de la información que manejan, así como la necesidad de interoperabilidad con otros sistemas externos. La correcta determinación de estos son claves para la especificación de la arquitectura.

2.7.2 Componentes de software

El desarrollo del sistema se realiza basado en componentes de software que permiten su reutilización en otros entornos o aplicaciones, el objetivo fundamental es satisfacer las demandas de mantenimiento y soporte del sistema una vez implantado a través de la actualización, mejora o incluso nuevo acoplamiento de componentes en un tiempo aceptable.

Un componente de software es una unidad de composición con interfaces contractualmente especificadas y explícitas sólo con dependencias dentro de un contexto. Un componente de software puede ser desplegado independientemente y es sujeto a la composición de terceros.[40] Un componente debe estar caracterizado por ser: identificable, auto contenido, reemplazable, con acceso solamente a través de su interfaz, bien documentado, genérico, reutilizable dinámicamente e independiente de la plataforma.

2.8 Definición de los indicadores a evaluar

Para la definición de los indicadores a evaluar es preciso identificar las variables presentes en la hipótesis de la investigación y sus indicadores fundamentales, como se detalla a continuación:

- **Eficiencia de los métodos computacionales:** tiempo de ejecución, complejidad espacial, eficacia.
- **Funcionalidad del sistema informático:** cumplimiento de los requerimientos.

- **Nivel de automatización del proceso de asesoramiento genético:** verificación de la historia familiar, análisis desde diferentes perspectivas de representación, usabilidad del sistema.

Dado estos indicadores y priorizando algunos de ellos en función de su impacto en el resultado de la investigación, la validación de la propuesta se realizará a través de las siguientes actividades:

1. Análisis de la complejidad temporal según la cota máxima de los métodos computacionales implementados.
2. Pruebas funcionales al sistema informático implementado para determinar el cumplimiento de los requerimientos y la eficacia de los métodos computacionales integrados al sistema.
3. Usabilidad del sistema informático.

Se desarrollará además la implementación de otras perspectivas de representación de árboles genealógicos como una de las necesidades determinadas a partir de los indicadores identificados: las vistas concéntrica cuadrada y horizontal, que permiten la visualización y análisis de las relaciones próximas a un individuo seleccionado.

Para realizar pruebas funcionales, la especificación se analizará para derivar los casos de prueba. Se consideran condiciones inválidas e inesperadas de la entrada y tener en cuenta que la definición del resultado esperado es una parte vital de un caso de la prueba. Específicamente se utilizarán pruebas de caja negra, que se centran en los requisitos funcionales del software, es decir, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra intenta encontrar errores en las siguientes categorías [41]:

- Funciones incorrectas o ausentes
- Errores de interfaz
- Errores en estructuras de datos
- Errores de rendimiento
- Errores de inicialización y de terminación

Conclusiones parciales

La definición de las herramientas y tecnologías fundamentales para el desarrollo de la investigación basados en el estudio de las principales plataformas y lenguajes para el desarrollo de sistemas informáticos basado en tecnologías libres, constituye la base para el planteamiento de la arquitectura del sistema y las características de los métodos computacionales a desarrollar. En particular al seleccionar la plataforma Java para el desarrollo de la investigación existen múltiples tecnologías asociadas que permiten una solución heterogénea en cuanto al uso de librerías y componentes externos ya desarrollados y a la disposición de la comunidad.

La selección de los métodos y metodología a seguir durante el desarrollo de la investigación guiarán el proceso y establecerán las formas de trabajo tanto para los flujos de la ingeniería de software como para la definición de los métodos computacionales específicos para la representación y análisis de los árboles genealógicos.

3

Resultados y discusión

En el presente capítulo se describen los métodos computacionales para la representación y análisis de árboles genealógicos. Así como los principales resultados del proceso de desarrollo del sistema informático implementado. En una segunda parte se analizarán los resultados obtenidos a través de pruebas funcionales y unitarias. Se muestran algunas pantallas del sistema implementado.

Resultados

3.1 Métodos computacionales para la representación de árboles genealógicos

En la presente sección se describen los principales algoritmos y componentes visuales u otros, definidos para la representación de los árboles, los mismos han sido implementados sobre el grafo definido para la modelación de los datos genealógicos.

3.1.1 Representación del árbol genealógico en el grafo diseñado

La figura 3.1 muestra un mapeo de un árbol genealógico simple según el grafo definido para su modelación en el que, según la definición de grafo, los vértices contienen la información de las diferentes personas y su lista de adyacencia está formada por listas de arcos según las relaciones que se establecen entre las personas. Esta representación es la más eficiente

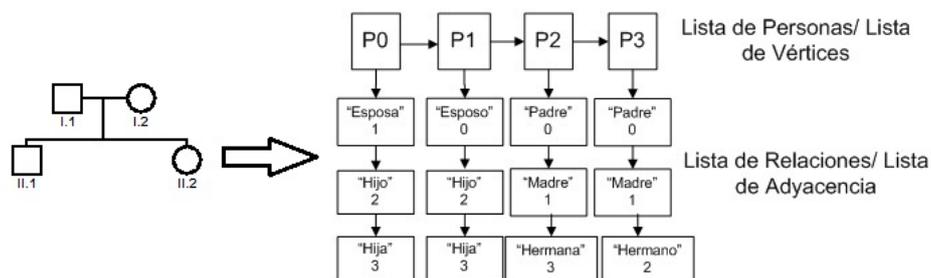


FIGURA 3.1: Representación del grafo

de grafos en cuanto a complejidad temporal y espacial de las operaciones debido a que la lista de adyacencia es una estructura dentada que sólo contiene los espacios en memoria reservados para las relaciones que se van creando y en cada arco solo se tiene la información de la relación y el índice del otro vértice vinculado en la relación. Las funcionalidades en el grafo en su mayoría son de orden n^2 según la cota máxima, pero con esta representación la cota exacta para las funciones es menor que n^2 a no ser que existiesen relaciones entre todos los pares de vértices, cosa que en un árbol genealógico no debe ocurrir.

3.1.2 Implementación de los eventos visuales y componentes necesarios para la visualización computacional de los árboles genealógicos

Los componentes definidos para la visualización del árbol consisten fundamentalmente en dos: "FamiliaVisual" y "PersonaVisual". Como su nombre lo indica la visualización se realiza en dos niveles fundamentales, la visualización de las personas con sus datos y simbología particular y la visualización del árbol con las personas y sus relaciones. El diagrama de clases representado en la figura 3.2 muestra la relación entre estos componentes para lograr la visualización de los árboles genealógicos.

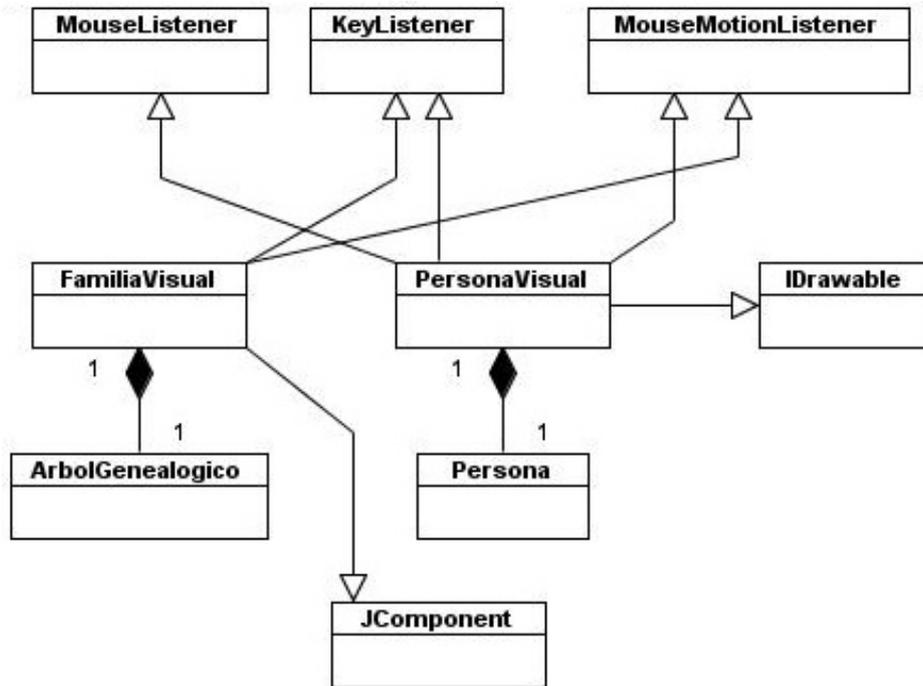


FIGURA 3.2: Clases fundamentales y sus relaciones para la visualización de los árboles genealógicos

Al interactuar con el sistema el área para el dibujo del árbol es el componente FamiliaVisual que muestra la figura, hereda de la clase JComponent para obtener el comportamiento de esta clase que contiene la lógica necesaria general para el comportamiento de un componente visual en Java. Por su parte el componente PersonaVisual se ha implementado extendiendo de la clase abstracta IDrawable que se utilizará para definir la forma en que se dibujará el componente a través de la implementación de su abstracción (método draw()). Ambas clases implementan además las interfaces MouseMotionListener, MouseListener y KeyListener para el manejo de los eventos que se producirán sobre los componentes visuales. Tienen además como parte de sus datos internos el objeto correspondiente a la entidad que representan, las clases Persona y ArbolGenealogico.

La forma de implementación de el componente FamiliaVisual en particular permite la visualización y representación de árboles complejos en cuanto a la cantidad de miembros, facilita a los usuarios el manejo de la topología del árbol y su organización de una manera flexible y personalizada, aspecto positivo debido a que otros sistemas implementan sobre el área de dibujo una matriz que ubica a los individuos de una forma rígida en el espacio que le corresponde.

3.2 Métodos computacionales para el análisis de árboles genealógicos

Los métodos implementados para el análisis de los árboles se han desarrollado teniendo en cuenta no sólo la organización correcta de los árboles para facilitar su análisis y el trabajo de unir o desacoplar árboles creados con anterioridad. La búsqueda de patrones de herencia y la inferencia a partir de reglas contribuirán en esta primera aproximación al análisis automático de los riesgos de padecimiento de enfermedades de origen genético.

3.2.1 Algoritmo para la unión de árboles genealógicos

Con la centralización de los datos genealógicos es relevante la implementación de algoritmos que permitan unir árboles genealógicos para un individuo dado y con ello obtener mayor información de la historia familiar, el conocimiento de otros elementos que pueden estar presentes en otros árboles almacenados y la verificación de la historia familiar. Para la implementación del algoritmo para la unión de árboles genealógicos se han definido como principios y pasos lógicos los siguientes:

1. Debe estar identificado el individuo propósito del árbol. En la figura 2.1 el individuo señalado con una flecha en la parte inferior izquierda es identificado como el propósito del árbol.
2. Buscar el individuo propósito seleccionado en los árboles genealógicos almacenados.
3. Identificar el árbol o los árboles a unir si la búsqueda produce más de un resultado.
4. Complementar las estructuras de datos de los grafos seleccionados en la original o en nuevo grafo.
5. Ordenar el grafo según las generaciones de acuerdo al nivel del propósito seleccionado.
6. Dibujar el árbol resultado de la unión.

El algoritmo formalmente descrito se detalla en 1

Algorithm 1 mergeGenealogicTreesOfAPerson(Person Ind)

Require: Individuo propósito seleccionado.

Ensure: Grafo completo G_{Res} con la unión de individuos y relaciones de todos los grafos a los que pertenece Ind .

- 1: Create $G = mainTree$ {grafo resultado de la unión}
- 2: Crear la Lista de grafos $Lgraphs$
- 3: busca a Ind en todos los árboles y devuelve todos los árboles en que aparece Ind en la lista $Lgraphs$
- 4: Crear una lista de relaciones $Lrelations$
- 5: **while** $G_i.isnotEmpty()$ **do**
- 6: **if** $node_i$ of $G_i \subset G$ **then**
- 7: $G_{Res}.insert(node_i)$
- 8: $lrelations.Concatena(relaciones(node_i))$
- 9: **end if**
- 10: **end while**
- 11: $G_{Res}.updateRelations(lrelations)$
- 12: $G_{Res}.Ordenar()$
- 13: repaint G_{Res}
- 14: **return** $subG$.

3.2.2 Algoritmo para la acotación de árboles genealógicos

En algunos casos de estudio es relevante un número de relaciones mínimo determinado y se requiere el análisis de una sección del árbol. El propósito de este algoritmo es reducir las ramas por grados de relación o siguiendo los estándares genealógicos siguientes:

- Costados: Muestra las relaciones directas (no muestra a tíos, tíos abuelos, etc.). Se expande de forma exponencial en cada generación, mostrando padre y madre, cuatro abuelos/as, ocho bisabuelos/as.
- Agnaticia: línea masculina de sangre o linaje, esquema que muestra una línea directa solo entre varones (normalmente la que sigue al primer apellido).
- Cognaticia: línea femenina o de ombligo, esquema que muestra una línea directa solo entre hembras.

Esto formalmente constituye la determinación de un subgrafo G_i del grafo original que contiene a todos los miembros del árbol. El algoritmo planteado en 2 describe el procedimiento.

Algorithm 2 getSubTree

Require: Individuo seleccionado $Ind \in G$, Tipo de subárbol $Mode$.

Ensure: Subgrafo según el tipo de representación seleccionado $subG$.

```

1: Create  $subG = subgrafo$  {a subgraph Result}
2: if( $Mode ==$  "Costados").
3: if( $Ind.haveMother()$ ).
4:  $subG.Merge(getSubTree(Ind.getMother(), "Costados"))$ .
5: end if.
6: if( $Ind.haveFather()$ ).
7:  $subG.Merge(getSubTree(Ind.getFather(), "Costados"))$ .
8: end if.
9: end if.
10: else.
11: if( $Mode ==$  "Agnaticia" &  $Ind.getSex() ==$  F).
12: if( $Ind.haveMother()$ ).
13:  $subG.Merge(getSubTree(Ind.getMother(), "Agnaticia"))$ .
14: end if.
15: end if.
16: else.
17: if( $Mode ==$  "Cognaticia" &  $Ind.getSex() ==$  M).
18: if( $Ind.haveFather()$ ).
19:  $subG.Merge(getSubTree(Ind.getFather(), "Cognaticia"))$ .
20: end if.
21: end if.
22: Create a new GenealogicTree for  $subG$ .
23:  $subG.Ordenar()$ .
24:  $subG.repaint()$ .
25: return  $subG$ .

```

3.2.3 Algoritmo para el ordenamiento de los árboles genealógicos

El algoritmo para el ordenamiento se realizará basado en el meta algoritmo de Stein y Benteler [21], se define a continuación la ejecución de los pasos que describen los autores para la solución específica en el sistema a implementar. A continuación se detallan los pasos, el procedimiento formal está descrito en 3:

1. Tamaño del área de dibujo: el componente FamiliaVisual permite la representación

de árboles de tamaños dinámicos a través del uso de “Scroll” horizontal y vertical. El tamaño del área de dibujo estará determinada por la función Tamaño(entero subárbol) que sumará recursivamente el tamaño de cada subárbol. El tamaño será calculado por el eje de las ‘x’ según la función descrita y en el eje ‘y’ según la cantidad de niveles. Para el árbol de la figura 2.1 el tamaño es de $x=4$ y $y=3$.

2. Topología: se ordenarán los individuos según su generación y bajo los criterios estéticos definidos en la sección 1.3.
3. Estructuración del esquema: se asignarán coordenadas a los diferentes subárboles de acuerdo al tamaño establecido en el paso 1 y la topología determinada en el paso 2. Para esto se utilizará lo planteado por el algoritmo de Walker en su variante más eficiente planteada por Buchheim [42].
4. Mapeo reverso de la topología: se asignan valores de coordenadas a los objetos de tipo PersonaVisual pertenecientes a cada uno de los subárboles según el tamaño definido.
5. Se dibuja el árbol ordenado en el componente FamiliaVisual a través del uso del API de java para dibujo.

Algorithm 3 orderTree

Require: Grafo visual *GrafoNoPonderadoLA* $< IDrawable >$.

Ensure: Grafo visual G' ordenado.

- 1: PersonaVisual persona = BuscarIndividuo(G); {Buscar el primer individuo del árbol}
 - 2: ubicarPorNiveles(persona,G, ancho / 2 - 25, 50).
 - 3: eliminarSolapamientos(G);
 - 4: centrarPadres(G);
 - 5: aumentarPanel(G);
 - 6: Repaint().
 - 7: **return** G' .
-

3.2.4 Sistema basado en reglas para el cálculo de riesgo genético

El sistema basado en reglas implementado consta de dos componentes: uno para la edición de reglas basadas en la información que se maneja en el árbol y los factores de riesgos definidos para la enfermedad a estudiar y otro para la visualización de los resultados de la inferencia denominado “cálculo de riesgos”. Como se describe en la sección 2.6.2 en un primer acercamiento a la predicción automática de riesgos el objetivo del sistema se

enfoca al planteamiento de reglas por parte de los especialistas que permitan la madurez de modelos propios basados en sus investigaciones.

El anexo 3.5 muestra el diagrama de clases definidas para el sistema basado en reglas, es importante señalar que el diseño define una interfaz “Variables comunes.^a” a través de la cual se define el comportamiento de las clases a definir para las variables de las diferentes enfermedades, en particular se definieron las variables para el cáncer de próstata, mama y ovarios, por ser identificados como unas de las enfermedades de mayor incidencia en la mortalidad de las de su tipo en Cuba y por la cantidad de estudios que se han realizado en diferentes poblaciones por especialistas de la genética y oncología.

Edición de las reglas

El componente para la edición de reglas permite a los especialistas crear archivos para las reglas y su edición. Esto genera un archivo en formato DRL¹) con las reglas generadas por el editor.

Una vez ejecutada la aplicación alasARBOGEN 2.0, a partir de la cual el genetista accede al Editor, se podrá comenzar a desarrollar el proceso de elaboración de ficheros de reglas. Para esto lo primero que se debe hacer es crear un nuevo archivo de reglas. Luego se debe definir las variables relevantes según el estudio a desarrollar. La edición de las reglas se realizará entonces de acuerdo a las variables definidas y operadores lógicos y aritméticos. El usuario puede introducir tantas comparaciones como crea necesarias para formar las reglas.

Se define al finalizar el resultado esperado según las reglas definidas para el estudio, este resultado es recomendable que se calcule a través de procedimientos estadísticos y probabilísticos en alguna herramienta externa para este propósito. La figura 3.3 muestra la interfaz del componente para la edición de reglas.

Cálculo de riesgo

El cálculo de riesgo de padecimiento de una enfermedad de origen genético se realiza a partir de la selección en un árbol genealógico del paciente a estudiar, para obtener los datos que en él se almacenan y definir algunos adicionales relacionados con los factores

¹Extensión de archivos para Drools.

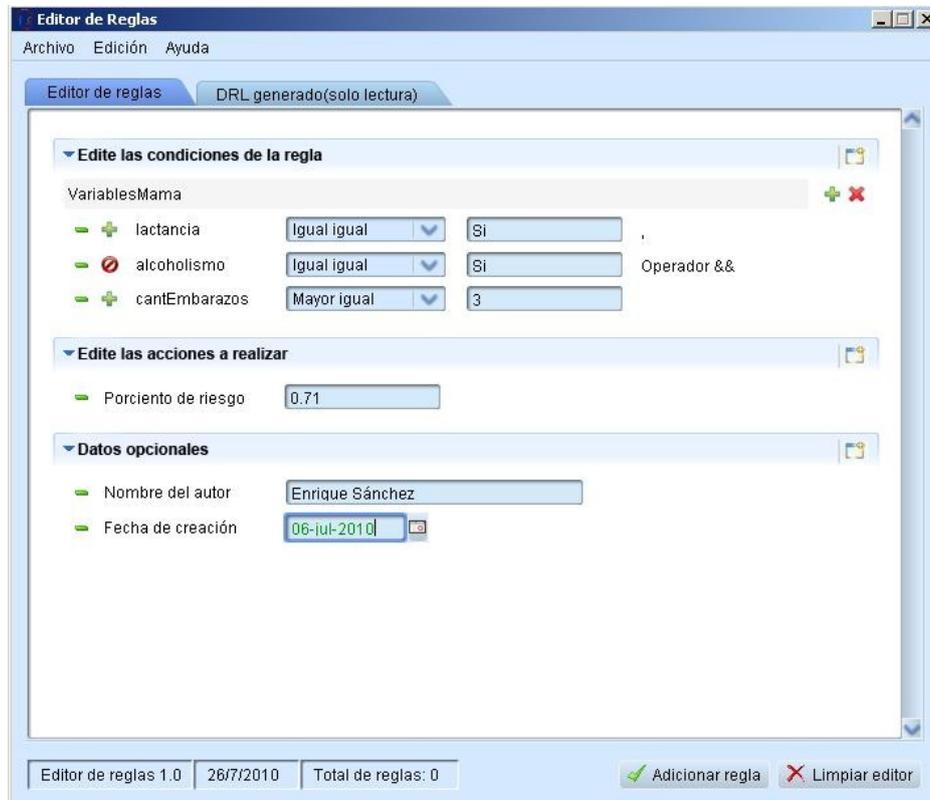


FIGURA 3.3: Interfaz del editor de reglas

identificados en la influencia de alguna enfermedad (denominados en el componente como variables).

La vista de este componente muestra las enfermedades para las que se han definido variables para su estudio, al seleccionar la enfermedad, se ahí procede a introducir los datos extras requeridos para el estudio de la enfermedad seleccionada. Se debe definir igualmente el archivo de reglas que ha sido descrito para la enfermedad en particular. Una vez introducidos los datos correctamente procede a realizar el cálculo que muestra una gráfica con la predicción del riesgo en intervalos de cinco años, añadiendo la opción de impresión del resultado obtenido. La figura 3.4 muestra un ejemplo de lo descrito.

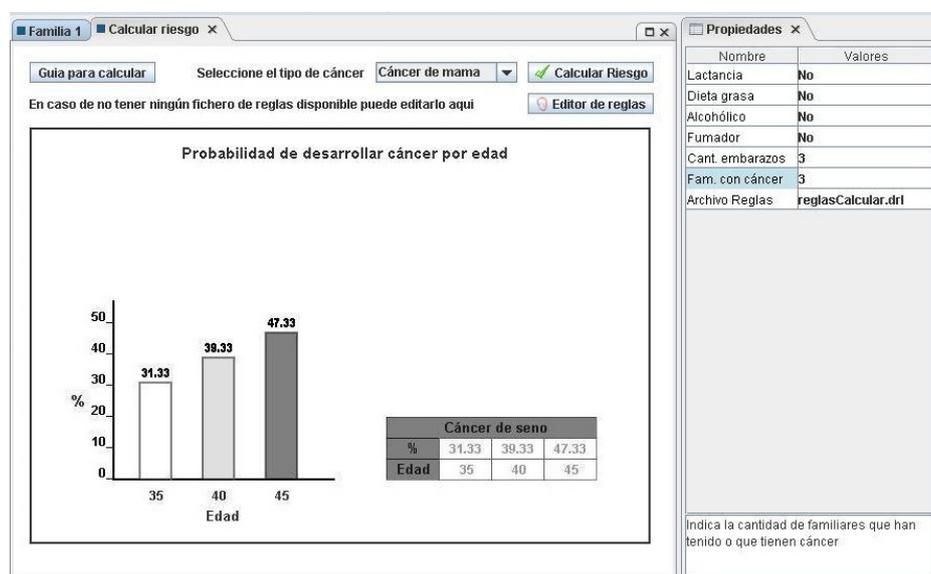


FIGURA 3.4: Área para el cálculo de riesgos

3.3 Sistema informático para la integración de los métodos computacionales

Los métodos computacionales antes descritos tendrán su funcionalidad integrados en la segunda versión del sistema alasARBOGEN, con este objetivo se realiza una actualización de los requisitos de la primera versión teniendo en cuenta el resultado del análisis del estado del conocimiento expuesto en la sección ?? y se detallan los nuevos requerimientos funcionales del sistema como se muestra en el anexo 3.5.

Es preciso definir los principios que se establecieron para la definición de la arquitectura del sistema, luego de analizado el entorno de despliegue, y pudiera definirse como base para el desarrollo de sistemas para la Red Nacional de Genética Médica:

- El sistema debe permitir el consumo de servicios web y la implementación de servicios propios.
- Será desarrollado utilizando tecnologías libres.
- El acceso a los sistemas debe ser a través del SAAA.
- Todos los centros de genética tienen acceso a la red, con limitaciones en la velocidad, debido a que fundamentalmente se conectan a través de módems.

RESULTADOS Y DISCUSIÓN

- Los árboles genealógicos contienen información confidencial de las personas y no pueden ser divulgados, ni publicados sin autorización.
- Los usuarios del sistema pueden tener sus propios árboles genealógicos almacenados localmente en sus computadoras personales, además de la base centralizada para el sistema.
- Los árboles genealógicos contruidos deben tener la capacidad de ser importados por sistemas similares en el mundo basados en formatos estandares para el almacenamiento de la información.

La figura 3.5 muestra un esquema general de la arquitectura del sistema definida según los criterios antes expuestos. Basado en los requisitos funcionales y no funcionales se determinaron como componentes de la arquitectura para el sistema:

- Una aplicación de escritorio para la representación y análisis de los árboles genealógicos: encargada de la representación y análisis de árboles genealógicos, permite almacenar la información de los datos genealógicos en ficheros, imágenes y hacia la base de datos.
- Aplicación Web: permitirá cargar ficheros de árboles construidos a partir de la aplicación de escritorio y almacenarlos a la base de datos. Visualizar y compartir árboles genealógicos entre los especialistas. Brindará soporte al despliegue de la aplicación de escritorio permitiendo la descarga y actualización de la misma a través de la red utilizando la tecnología Java Web Start.
- Base de datos: es común para ambas aplicaciones, maneja toda la información de las personas y sus relaciones, así como todos los nomencladores y la información para la administración del sistema relacionada con los usuarios y roles.
- Reporteador dinámico: aplicación para el diseño de reportes dinámicos que permitirá el estudio de los datos almacenados con flexibilidad, debido a los múltiples factores que intervienen en la prevención de las enfermedades y el estudio de las mismas.

A continuación se describen algunos elementos internos de la aplicación de escritorio, en la que se integran los métodos computacionales descritos.

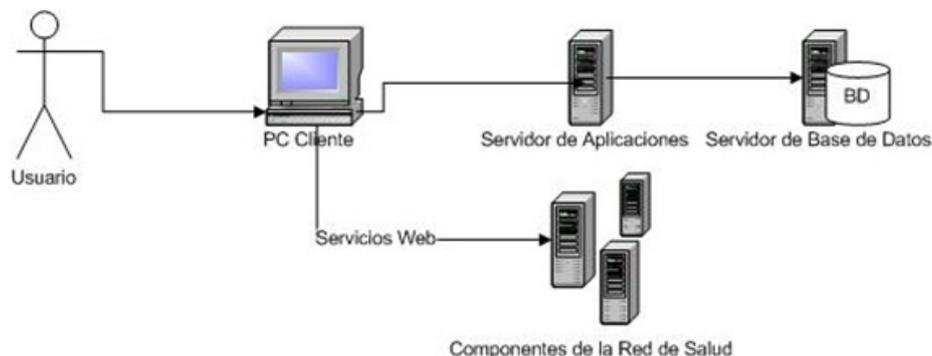


FIGURA 3.5: Esquema general de la arquitectura del sistema

3.3.1 Componentes de la herramienta informática implementada

Los componentes fundamentales del sistema son: la base de datos, el visor de propiedades de los datos, el componente para el dibujo del árbol y su ventana principal contenedora de los componentes visuales. Han sido descritos anteriormente por su relevancia el componente de dibujo, compuesto por los clases `FamiliaVisual` y `PersonaVisual`, en este epígrafe se describirá el resto.

Ventana principal

La ventana principal es una clase del tipo `JFrame` que contiene todos los componentes visuales del sistema, como aspecto relevante es necesario destacar, que el entorno visual de la aplicación se configuró haciendo uso de la biblioteca “`idw-gpl`”², lo que permite:

- Acoplamiento de nuevas interfaces visuales y áreas de trabajo de forma sencilla.
- Un acceso rápido a todas las áreas de trabajo de la aplicación.
- Cambiar el diseño de la aplicación de forma dinámica y sencilla.

Esta configuración, en otros términos permite el dibujo de varios árboles genealógicos simultáneamente, así como el acoplamiento de componentes en el tiempo que adquieren

²Disponible en: <http://dock.javaforge.com/>

automáticamente el diseño del sistema, lo que es relevante para el mantenimiento y extensibilidad del sistema en futuras versiones. Propicia además la adición del resultado de otros grupos de trabajo y otras áreas de la genética como la epidemiología, y podría sentar las bases para la integración de los sistemas de escritorio que se implementan para el CNGM y su red. Una vista de esta área se puede observar en el epígrafe 3.4.3.

Visor de propiedades

El visor de propiedades como su nombre lo indica se utiliza para la visualización de los datos de las diferentes entidades a través de anotaciones, a continuación se describen fragmentos de código que se encuentran en la clase “Persona”, que posibilitan gestionar los datos de la persona a través del visor de propiedades, utilizando los componentes visuales adecuados para visualizar cada atributo anotado. Para utilizar el visor de propiedades, es necesario hacer el uso de la anotación “@PropertyField” que es la que indica que el atributo que se encuentra anotado va a constituir una propiedad de esa entidad. Para conocimiento de los eventos que ocurren en el visor de propiedades se tiene que adicionar un “Listener”. Al visor de propiedades, creando una nueva instancia de la interface “IPropertyListener”, haciendo uso del método “registrarListener(IPropertyListener object)”. El visor de propiedades reconoce el tipo de editor visual que se va a utilizar acorde a la palabra clave que se le indique haciendo uso de la propiedad “propertyType()”. Las palabras que representan los datos para la muestra de los diferentes componentes en función de la anotación, son las siguientes:

- **Comments:** Indica que ese atributo al que se está haciendo referencia va a contener un comentario.
- **Boolean:** Indica que dicho atributo, contiene valores booleanos solamente.
- **Date:** Indica que el atributo lo que contiene son valores de fecha.
- **Color:** Indica que el atributo lo que contiene son valores de tipo color.
- **Font:** Indica que en ese atributo lo que se guarda son valores de fuente.
- **Text:** Indica que el atributo representa un texto normal.
- **List:** Indica que el atributo va a representar una lista de valores, posible a desplegarse en un JComboBox.
- **Tree:** Indica que en ese campo se va representar una composición y que se va a crear una sub tabla dentro de la misma tabla, para representar los valores de la composición. Se logra el funcionamiento del JTree.

- **Add:** Representa en algún campo que requiera de la entrada de valores.
- **AddMultiple:** Se tiene en cuenta cuando se va a llenar un atributo de tipo lista, desde una lista de datos origen.

Otro aspecto importante que se puede evidenciar en el visor de propiedades, es que se le puede indicar cuando los valores que va a tener son estáticos y cuando se van a cargar los valores de forma dinámica. Para darle este funcionamiento se utiliza la anotación “datosExternos()” que recibe un valor booleano y se le tiene que indicar el nombre del método haciendo uso de la anotación “nombreMetodo()”. Esta anotación por defecto está en blanco, por si no se desean cargar los valores de un método externo, en caso contrario, el nombre del método que se va a especificar en esa anotación, debe pertenecer a la clase que está anotada para cargar los valores externos. A continuación se muestran fragmentos de códigos de la clase “Persona”, donde queda reflejado con claridad como se utilizan las anotaciones para el visor de propiedades.

```
1 @PropertyField(propertyName="Nombre", propertyCharType = "L", Expresion = "  
    [[a-zA-Z][á-ú][Á-Ú]+\s]*", propertyType="Text", propertyToolTip="Nombre  
    del individuo")  
2 protected String nombre;  
3  
4 @PropertyField(propertyName = "CI", propertyType="Text", propertyCharType =  
    "N", Expresion = "[0-9]*", propertyToolTip="Carne de identidad del  
    individuo")  
5 protected String ci;  
6  
7 @PropertyField(propertyName = "Sexo", propertyType="List", noEditable={"  
    Masculino", "Femenino"}, propertyToolTip="Sexo del individuo",  
    datosExternos = true, nombreMetodo = "getListaSexo")  
8 protected Sexo sex;  
9  
10 @PropertyField(propertyName = "Fecha Muerte", propertyType="Date",  
    propertyToolTip="Fecha en que muere el individuo")  
11 protected Date fechaMuerte;  
12  
13 @PropertyField(propertyName = "Dirección", propertyType="Tree",  
    propertyToolTip="Dirección del individuo")  
14 protected Direccion direccion;  
15  
16 @PropertyField(propertyName = "Propósito", isEditable= true, propertyType="  
    Boolean", propertyToolTip="Indica si el individuo es el propósito del  
    estudio")  
17 protected boolean proposito;  
18  
19 @PropertyField(propertyName = "Enfermedades", propertyType="AddMultiple",  
    propertyToolTip="Adicionar enfermedades al individuo", datosExternos =  
    true, nombreMetodo="getListaEnfermedades")
```

```
20 private List<Enfermedad> enfermedades ;
21
22
23 @PropertyField(propertyName = "Símbolo Adicional", propertyType="List",
    propertyToolTip="Símbolo adicional del individuo en la familia",
    datosExternos = true, nombreMetodo="getSimbolosAdicionales")
24 protected SimboloAdicional simboloAdicional;
25
26 @PropertyField(propertyName = "Comentario", propertyType="Comments",
    propertyToolTip="Comentarios sobre las enfermedades del individuo")
27 protected String comentarios;
```

Esta implementación del componente visor de propiedades lo hace genérico y por tanto reutilizable en otros sistemas o herramientas desarrolladas en Java.

Base de datos

Luego de la definición de las entidades en el modelo de diseño y sus correspondientes clases persistentes. Se implementa la base de datos para el sistema que consta de 25 tablas, para el almacenamiento de las personas, sus relaciones y los nomencladores para la gestión de árboles genealógicos. Para el acceso a los datos se implementa una capa de acceso utilizando Hibernate y JPA, que permite el acceso a la base de datos con independencia del gestor de base de datos a utilizar, así como la generación de la base de datos a partir de las entidades y sus anotaciones, lo que incrementa la facilidad de actualización del modelo de datos.

La base de datos se encuentra en tercera forma normal (3FN) por lo que no existen atributos multievaluados, ni dependencias transitivas entre las relaciones, como tampoco redundancia de la información. La 3FN es la más usada en la totalidad de los productos que utilizan bases de datos porque garantiza una redundancia casi nula de información. Se utilizaron métodos teóricos para la evaluación de la integridad de la información que evaluaron la integridad referencial, de dominio y de la entidad, arrojando resultados positivos y permitiendo la corrección de los problemas detectados. Se eliminaron las inconsistencias detectadas a través del uso de nomencladores para todos aquellos datos comunes dentro de la base de datos. Entre los nomencladores definidos en la BD del sistema se pueden mencionar símbolo, tipo de embarazos, color de la piel, color de los ojos, enfermedades, municipios, provincias, país entre otros.

Discusión

3.4 Validación de los resultados de la implementación

La validación de los resultados de la implementación se expone en los subepígrafos siguientes, según las actividades identificadas para la validación en ??.

3.4.1 Análisis de la complejidad temporal de los métodos computacionales implementados.

La implementación de los algoritmos basados en la estructura de datos de tipo grafo descrita permite minizar el tiempo de ejecución para la creación de relaciones y la reutilización de métodos propios de grafo que han sido implementados con anterioridad. Como base se define la complejidad temporal de operaciones según O para la estructura de datos utilizada en la tabla 3.1. Para los métodos de abordados en la sección de resultados el

Operación	Complejidad
Insertar Vértice	1
Eliminar Vértice	n^2
Insertar Arco	n
Eliminar Arco	n
Buscar Vértice	n

TABLA 3.1: Tiempos de ejecución según la cota máxima de las operaciones básicas de grafos

análisis de los tiempos de ejecución fue calculado teniendo como base estas operaciones, los detalles del cálculo de la complejidad en cada caso se exponen en la tabla 3.2. Los tiempos de ejecución obtenidos son de orden polinomial en el peor de los casos, por lo que los tiempos de ejecución en función de la entrada (n que consiste en la cantidad de nodos del grafo) está dado por una función de crecimiento poco trascendente (a lo sumo de grado 2) y más eficientes que las estructuras de listas anidadas que contenía la versión anterior del sistema.

3.4.2 Pruebas funcionales al sistema informático implementado.

Las pruebas de caja negra realizadas al sistema garantizan el 100 por ciento del cumplimiento de los requisitos funcionales, basado en los 34 requisitos funcionales descritos en el anexo 3.5, se efectuaron un total de 105 casos de prueba con entradas válidas e inválidas,

Algoritmo	Complejidad
mergeGenealogicTreesOfAPerson	n^2
getSubTree	$n \log n$
orderTree	n
drawTree	n

TABLA 3.2: Tiempos de ejecución según la cota máxima de los algoritmos implementados sobre el grafo

arrojando resultados satisfactorios. Los problemas detectados fueron fundamentalmente relacionados con la validación de los campos y con la visualización de algunas relaciones, siendo corregidos.

3.4.3 Usabilidad del sistema informático.

Para la evaluación de la usabilidad del sistema implementado se utilizó la herramienta DataLogger 5.1.1 desarrollada en Visual Basic sobre Microsoft Excel [43] por un grupo de expertos en pruebas de usabilidad, dicha herramienta brinda la posibilidad a través de un test a los usuarios de obtener su percepción sobre el sistema de acuerdo a un grupo determinado de **tareas en diferentes escenarios** que para el **análisis de los datos reales** obtenidos en la evaluación del sistema según lo establecido por [44]. Se identificaron específicamente 8 tareas de diferente nivel de complejidad especificadas en el anexo 3.5. Para su realización se seleccionaron 5 usuarios finales del sistema pertenecientes al CNGM a los que se les realizó el test de "Utilidad y Facilidad de uso" definido por [43] (expuesto en el Anexo 3.5). Los resultados arrojaron un saldo positivo en cuanto a Facilidad de uso con un índice de 6.5 de Utilidad y un 6.8 de Facilidad de Uso como muestra el anexo 3.5.

La figura 3.6 muestra la pantalla principal del sistema, donde se pueden apreciar las diferentes áreas de trabajo y visualización de los datos personales. Se puede apreciar en la misma un árbol correctamente ordenado y que tiene los símbolos y relaciones según lo establecido por PSWG.

3.5 Visión global del trabajo realizado

El trabajo realizado constituye un aporte práctico al proceso de informatización de la salud cubana y en particular a la Red Nacional de Genética Médica. Como se ha expuesto

RESULTADOS Y DISCUSIÓN

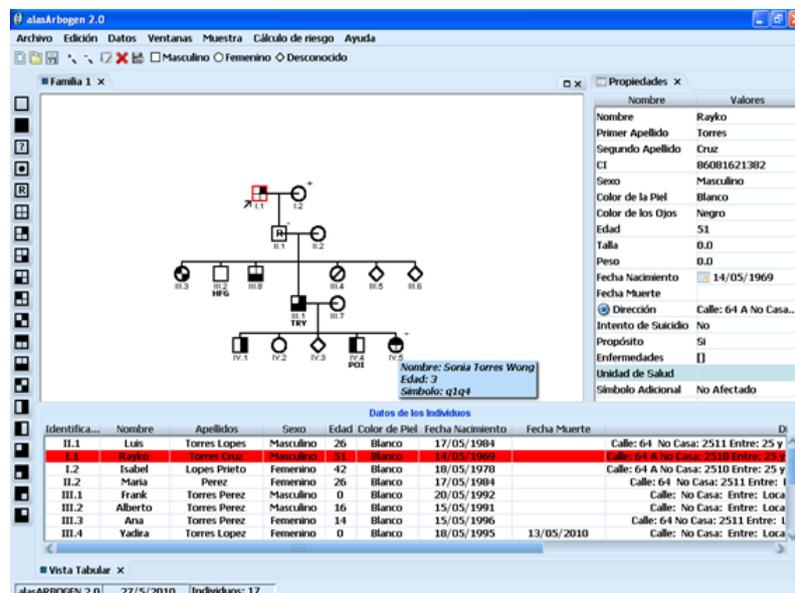


FIGURA 3.6: Pantalla principal de alasARBOGEN 2.0

en los epígrafes precedentes, la implementación de métodos computacionales sobre la estructura de datos adecuada, propicia una elevación en la eficiencia del sistema informático implementado. La validación de los resultados de la implementación según los indicadores definidos en la hipótesis permiten afirmar su cumplimiento. La solución contiene además otros elementos arquitectónicamente relevantes que hace de esta versión del sistema un producto que tiene una serie de beneficios tangibles e intangibles para el proceso de asesoramiento genético y para el CNGM en particular:

- Intangibles
 - Incremento de la cultura informática de la institución en el uso de la tecnología como instrumento de trabajo diario.
 - Aumento del alcance.
 - Mayor satisfacción de los usuarios del sistema.
 - Estandarización de los procesos.
 - Aumento de la efectividad organizacional.
 - Información más relevante para el análisis y toma de decisiones.
- Tangibles
 - Base de datos centralizada para la gestión los árboles genealógicos.

- Libertad tecnológica.
- Escalabilidad y soporte.
- Soporte de múltiples plataformas o sistemas operativos.
- Soporte a la definición de reglas de producción sobre la información genealógica.
- Interoperabilidad con otros sistemas similares basado en ficheros GEDCOM.
- Los árboles genealógicos son representados siguiendo los elementos estéticos definidos como estándares para la comunidad científica de esta rama.

Conclusiones parciales

Se presentaron los principales resultados de la investigación descritos como los algoritmos para la representación y análisis de árboles genealógicos, según la estructura de datos basada en grafos diseñada y teniendo en cuenta investigaciones precedentes en este campo. Se describen los componentes fundamentales del sistema informático implementado, destacando los logros fundamentales de la implementación en función de lograr un sistema acorde al desarrollo actual de la Red Nacional de Genética. Se realiza la validación del sistema de acuerdo a los indicadores establecidos tales como la complejidad temporal de los algoritmos implementados, la funcionalidad del sistema y la usabilidad del sistema.

Conclusiones

- El análisis del estado del conocimiento acerca de la representación y análisis de árboles genealógicos permitió la identificación de algoritmos y procedimientos para ese propósito, así como la definición de la estructura de datos adecuada para la modelación de los datos genealógicos.
- Se diseñaron e implementaron cuatro algoritmos fundamentales para la representación y análisis de árboles genealógicos y un sistema basado en reglas con editor propio para la inferencia de riesgos y el desarrollo de investigaciones clínico-genéticas.
- La actualización de los requerimientos funcionales y no funcionales del sistema informático a implementar permiten una definición estructural del sistema acorde al entorno de despliegue y permite en particular la definición de una aplicación informática capaz de garantizar la representación correcta de los árboles genealógicos y un adecuado manejo de los datos genealógicos.
- Se implementaron los componentes de la aplicación de escritorio orientados a la extensibilidad y mantenimiento, utilizando tecnologías multiplataforma y libres que permiten su reutilización en aplicaciones similares.
- Se validó la aplicación de escritorio implementada y los algoritmos una vez integrados utilizando pruebas de caja blanca y caja negra, permitiendo el refinamiento de 5 funcionalidades y obteniendo finalmente una aplicación completamente funcional según los requisitos identificados y los indicadores definidos para su evaluación.

Recomendaciones

- Desarrollar niveles de prueba más profundas para garantizar la idoneidad del sistema en su entorno, pudieran ser aplicadas pruebas de carga y estrés.
- Desarrollar servicios web para favorecer la integración del sistema a Infomed.
- Desarrollar un sistema basado en casos para potenciar el diagnóstico automático de las enfermedades más conocidas e integrarlo al sistema.

Referencias Bibliográficas

- [1] in *Diccionario médico-biológico, histórico y etimológico*, 2010.
- [2] B. Marcheco Teruel, “Genética Médica y Enfermedades Crónicas: el camino de la Prevención,” *Revista Cubana de Genética Comunitaria*, vol. 2, 2008.
- [3] E. M. Peralta, “La anamnesis y el examen físico en el estudio genético clínico de las sorderas hereditarias,” *Revista Cubana de Pediatría*, vol. 76, 2004.
- [4] “¿qué es el consejo genético en el cáncer hereditario?” International Cancer Consultants, S.L., 2010.
- [5] B. Marcheco Teruel, “Genética médica y enfermedades crónicas: el camino de la prevención,” *Revista Cubana de Genética Comunitaria*, vol. 2, 2009.
- [6] W. A. S. Martin, Jeffrey R., “Genetic counseling in primary care: longitudinal, psychosocial issues in genetic diagnosis and counseling,” *Prim Care*, vol. 31, no. 3, pp. 509–24, viii–ix, 2004.
- [7] C. A. de Zayas, *Metodología de la investigación científica*. 1995.
- [8] R. L. Bennett, K. A. Steinhaus, S. B. Uhrich, C. K. O’Sullivan, R. G. Resta, D. Lochner-Doyle, D. S. Markel, V. Vincent, and J. Hamanishi, “Recommendations for standardized human pedigree nomenclature,” *Journal of Genetic Counseling*, vol. 4, pp. 267–279, 1995. 10.1007/BF01408073.
- [9] “Cyrillic 3 standard version,” CyrillicSoftware, 2000.
- [10] “Pedigree drawing and data management,” Progeny Software, LLC, 2010.
- [11] V.-P. Mäkinen, M. Parkkonen, M. Wessman, P.-H. Groop, T. Kanninen, and K. Kasiki, “High-throughput pedigree drawing,” *Eur J Hum Genet*, vol. 13, no. 8, pp. 987–9, 2005.

- [12] H. Thiele and P. Nürnberg, “Haplopainter: a tool for drawing pedigrees with complex haplotypes.,” *Bioinformatics*, vol. 21, no. 8, pp. 1730–2, 2005.
- [13] J. Zhao, “Pedigree-drawing with r and graphviz.,” *Bioinformatics*, 2006.
- [14] A. V. Kirichenko, “An algorithm of step-by-step pedigree drawing,” *Russian Journal of Genetics*, vol. 40, pp. 1176–1178, 2004. 10.1023/B:RUGE.0000044764.33578.69.
- [15] E. H. Trager, R. Khanna, A. Marrs, L. Siden, K. E. H. Branham, A. Swaroop, and J. E. Richards, “Madeline 2.0 pde: a new program for local and web-based pedigree drawing.,” *Bioinformatics*, vol. 23, no. 14, pp. 1854–6, 2007.
- [16] C. Wetherell and A. Shannon, “Tidy drawings of trees,” *IEEE Trans. Softw. Eng.*, vol. 5, pp. 514–520, September 1979.
- [17] J. Q. Walker, II, “A node-positioning algorithm for general trees,” *Softw. Pract. Exper.*, vol. 20, pp. 685–705, July 1990.
- [18] E. M. Reingold and J. S. Tilford, “Tidier drawings of trees,” *IEEE Trans. Softw. Eng.*, vol. 7, pp. 223–228, March 1981.
- [19] G. M. Radack, “Tidy drawing of m-ary trees,” Tech. Rep. CES-88-24, Department of Computer Engineering and Science, Case Western Reserve University, Cleveland, Ohio, Noviembre 1988.
- [20] A. Kennedy, “Drawing trees,” *J. Funct. Program.*, vol. 6, no. 3, pp. 527–534, 1996.
- [21] B. Stein and F. Benteler, “On the generalized box-drawing of trees: Survey and new technology,” in *Proceedings of the 7th International Conference on Knowledge Management (I-KNOW 07)*, Graz (K. Tochtermann and H. Maurer, eds.), Journal of Universal Computer Science, pp. 408–415, Springer, September 2007.
- [22] R. C. Elston and E. Sobel, “Sampling considerations in the gathering and analysis of pedigree data.,” *Am J Hum Genet*, vol. 31, no. 1, pp. 62–9, 1979.
- [23] D. M. J. K. L. Pratt, S. C., “Exact multipoint quantitative-trait linkage analysis in pedigrees by variance components.,” *Am J Hum Genet*, vol. 66, no. 3, pp. 1153–7, 2000.
- [24] F. Tores and E. Barillot, “The art of pedigree drawing: algorithmic aspects.,” *Bioinformatics*, vol. 17, no. 2, pp. 174–9, 2001.
- [25] E. Horowitz, D. Mehta, and S. Sahni, *Fundamentals of Data Structures in C++*. New York, NY, USA: W. H. Freeman & Co., 1995.
- [26] “Java,” Oracle, 2010.

REFERENCIAS BIBLIOGRÁFICAS

- [27] “Events (csharp vs. java),” Microsoft Corporation, 2008.
- [28] “Package java.awt.event,” Sun Microsystems.
- [29] “Annotations,” Oracle, 2010.
- [30] “The reflection api,” Oracle, 2010.
- [31] T. E. Foundation, “What is eclipse?,” 2010.
- [32] “About hibernate,” Jboss Community, 2010.
- [33] L. ren Jen and Y. jye Lee, “Working group. ieee recommended practice for architectural description of software-intensive systems,” *IEEE Architecture*, pp. 1471–2000, 2000.
- [34] A. Berson, *Client/server architecture (2nd ed.)*. New York, NY, USA: McGraw-Hill, Inc., 1996.
- [35] A. Delgado Ramos, M. Cabrera Hernández, A. Sánchez Rodríguez, and A. Rodríguez Díaz, “Registro informatizado de salud,” 2009.
- [36] M. Shaw and P. Clements, “The golden age of software architecture,” *IEEE Software*, vol. 23, pp. 31–39, 2006.
- [37] D. Garlan and M. Shaw, “An introduction to software architecture,” Tech. Rep. CMU-CS-94-166, Carnegie Mellon University, January 1994.
- [38] R. L. Bennett, K. S. French, R. G. Resta, and D. L. Doyle, “Standardized human pedigree nomenclature: Update and assessment of the recommendations of the National Society of Genetic Counselors,” *J. Genet. Counsel.*, vol. 17, no. 424–433, 2008.
- [39] “What is openup?,” Eclipse Procces Framework, 2010.
- [40] C. Szyperski, *Component Software: Beyond Object-Oriented Programming*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 2002.
- [41] R. Pressman, *Software Engineering: A Practitioner’s Approach*. McGraw-Hill, sixth ed., 2005.
- [42] C. Buchheim, M. Jünger, and S. Leipert, “Improving walker’s algorithm to run in linear time,” in *Revised Papers from the 10th International Symposium on Graph Drawing*, GD ’02, (London, UK), pp. 344–353, Springer-Verlag, 2002.
- [43] T. Zazelenchuk, “How to install and use the usability datalogger (v5.0),” Userfocus LTD, 2008.

REFERENCIAS BIBLIOGRÁFICAS

- [44] T. Z. Alex Genov, Mark Keavney, “Usability Testing with Real Data,” *Journal of Usability Studies*, vol. 4, 2009.

Glosario de Términos

API: Una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

JDK: Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en java. Puede instalarse en una computadora local o en una unidad de red.

JPA: Java Persistence API, es la API de persistencia desarrollada para la plataforma Java EE.

Anexos

Anexo 1: Aval del cliente

Ciudad de La Habana, 4 de febrero de 2009
"Año del 50 aniversario del Triunfo de la Revolución"



Opinión del cliente sobre el Software alasArbogen

Este producto se ha desarrollado con el objetivo de poner a disposición de la Red Nacional de Genética Médica del país una herramienta propia para la confección de árboles genealógicos y el manejo de la información relacionada.

Este software permite de un modo rápido, sencillo y en la propia consulta de genética clínica o asesoramiento genético, hacer una representación gráfica de la familia en estudio. El mismo viene a satisfacer la necesidad de contar con un sistema de este tipo en idioma español que pueda sustituir el empleo de otro más complejo y menos accesible para la mayor parte de los usuarios de la red.

Entre las ventajas de este nuevo producto puede mencionarse la interfaz amigable y fácil de entender. El mismo brinda además, la posibilidad de generar imágenes que pueden ser exportadas para ser utilizadas por otras aplicaciones y pueden ser enviadas a otros genetistas sirviendo para hacer consultas a distancia. Esta manera de representación de la información puede estar accesible al instante, solo para los especialistas autorizados aun en lugares físicamente muy distantes lo que sería impracticable de no estar informatizada. De esa manera la información relevante puede ser compartida y empleada fácilmente por todos los especialistas involucrados, lo que garantiza un mejor y más integral manejo de los casos.

Por otra parte el sistema permite almacenar toda la información relevante referente a las familias, los estudios indicados y otros aspectos de interés, todo ello adecuado a las características propias de la atención que reciben los pacientes afectados por enfermedades genéticas y defectos congénitos en nuestro país.

El software cumple con los requisitos y posee las funcionalidades que fueron definidas en los inicios de su desarrollo. Su uso se ha generalizado a todos los centros de la Red Nacional de Genética Médica y ya ha comenzado a ser utilizado de manera masiva por sus especialistas.

Dra. Elsa García Castillo
Vicedirectora de Investigaciones

FIGURA 7: Opinión del cliente acerca del sistema

Anexo 2: Especificación de los requisitos funcionales del sistema

RF1. Gestionar Gráficamente un Individuo

- RF 1.1 Insertar un individuo.
- RF 1.2 Modificar datos de un individuo.
- RF 1.3 Eliminar un individuo.
- RF 1.4 Mostrar datos de un individuo.

- RF 1.5 Trasladar gráficamente un individuo.
- RF2. Gestionar Relaciones entre Individuos.
 - RF 2.1 Crear relación entre individuos.
 - RF 2.2 Eliminar relación entre individuos.
- RF3. Gestionar Símbolos.
 - RF 3.1 Crear nuevo símbolo.
 - RF 3.2 Modificar símbolos.
 - RF 3.3 Eliminar un símbolo.
- RF4. Gestionar Enfermedades.
 - RF 4.1 Crear una nueva enfermedad.
 - RF 4.2 Eliminar una enfermedad.
- RF5. Gestionar Muestras.
 - RF 5.1 Insertar nuevas muestras.
 - RF 5.2 Modificar una muestra.
 - RF 5.3 Eliminar una muestra.
- RF6. Gestionar Familia.
 - RF 6.1 Crear una nueva familia.
 - RF 6.2 Buscar una familia.
 - RF 6.3 Actualizar datos de una familia.
 - RF 6.4 Guardar familia.
 - RF 6.5 Exportar familia.
- RF7. Gestionar Dimensiones del Árbol Genealógico.
 - RF 7.1 Aumentar tamaño del árbol.
 - RF 7.2 Disminuir tamaño del árbol.
- RF8. Buscar en el Árbol Genealógico.
- RF9. Ordenar Árbol Genealógico.
- RF10. Mostrar Vista Previa.
- RF11. Imprimir Árbol Genealógico.
- RF12. Generar Reporte del Árbol Genealógico.
 - RF 12.1 Imprimir Reporte
- RF13. Compartir árbol genealógico.
- RF14. Autenticar Usuario.
- RF15. Gestionar Usuario.
 - RF 15.1 Registrar usuario.
 - RF 15.2 Modificar usuario.
 - RF 15.3 Listar usuarios.
 - RF 15.4 Eliminar usuario.
- RF16. Gestionar Nomencladores.

Anexo 3: Diseño de clases para el sistema basado en reglas

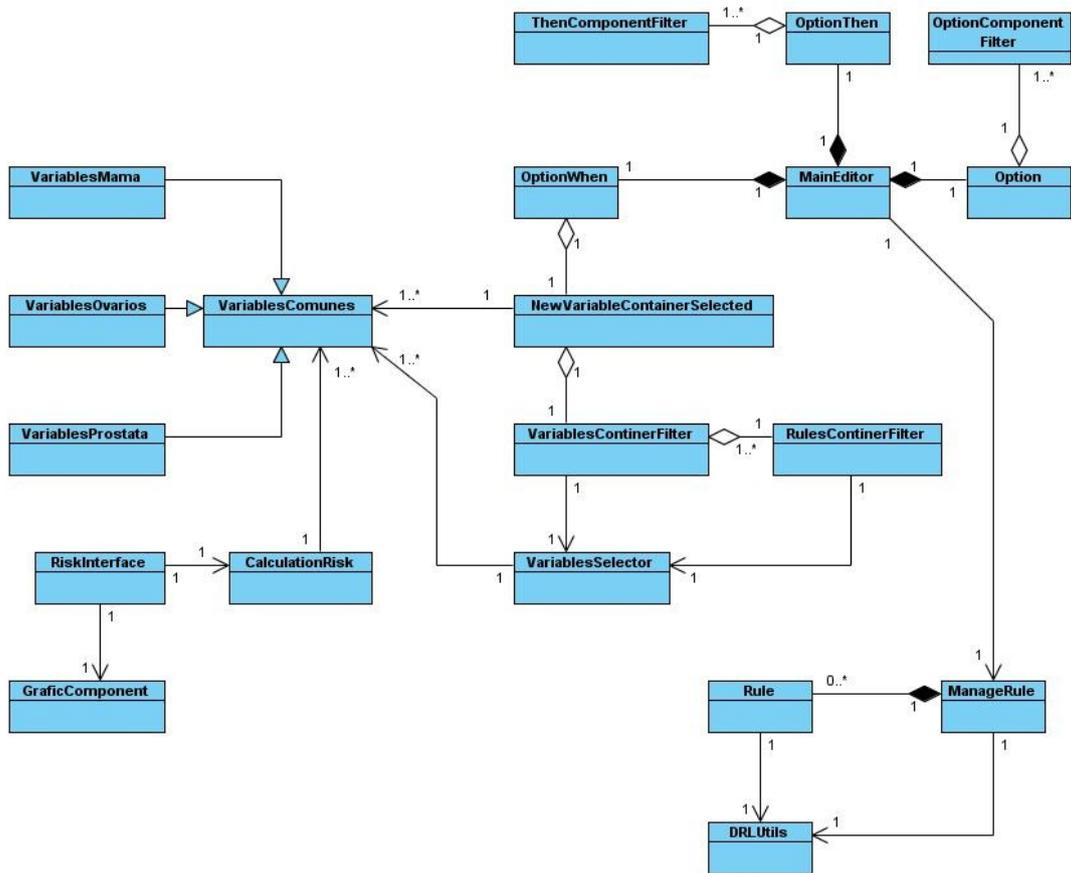


FIGURA 8: Interfaz del editor de reglas

Anexo 4: Cuestionario de Usabilidad

		alasARBOGEN 2.0						
Satisfaction Questionnaire		strongly disagree			strongly agree			
1	Using this product enables me to accomplish tasks more quickly.	1	2	3	4	5	6	7
2	Using this product improves my current performance.	1	2	3	4	5	6	7
3	Using this product increases my productivity.	1	2	3	4	5	6	7
4	Using this product makes me more effective.	1	2	3	4	5	6	7
5	Using this product makes it easier to do my work.	1	2	3	4	5	6	7
6	I find this product useful.	1	2	3	4	5	6	7
7	Learning to operate this product was easy for me.	1	2	3	4	5	6	7
8	I found it easy to get this product to do what I want it to do.	1	2	3	4	5	6	7
9	My interaction with this product was clear and understandable.	1	2	3	4	5	6	7
10	I found this product to be flexible to interact with.	1	2	3	4	5	6	7
11	It was easy for me to become skillful at using the system.	1	2	3	4	5	6	7
12	I found the system easy to use.	1	2	3	4	5	6	7

FIGURA 9: Cuestionario de Utilidad y Facilidad de Uso realizado a los especialistas del CNGM

Anexo 5: Tareas seleccionadas para medir la facilidad de uso del sistema.

Task#	Description	Score
1	Crear un árbol de 3 generaciones	Easy
2	Adicionar relaciones a un individuo seleccionado	Easy
3	Editar los datos personales de un individuo	Easy
4	Exportar reportes del árbol	Assist
5	Crear un árbol con simbología extra	Easy
6	Editar reglas para estudios de Cáncer de mama	Hard
7	Instalar y configurar el sistema	Assist
8	Unir árboles genealógicos dado un propósito	Medium

FIGURA 10: Tareas de usuario

Perceived Usefulness & Ease of Use	M	Strongly disagree Strongly agree						
		1	2	3	4	5	6	7
1 Using this product enables me to accomplish tasks more quickly.	7,0							≡≡≡
2 Using this product improves my current performance.	6,0					--	≡	--
3 Using this product increases my productivity.	6,0					--	≡	--
4 Using this product makes me more effective.	6,0					--	≡	--
5 Using this product makes it easier to do my work.	7,0							≡≡≡
6 I find this product useful.	7,0							≡≡≡
7 Learning to operate this product was easy for me.	7,0							≡≡≡
8 I found it easy to get this product to do what I want it to do.	6,6						=	≡≡
9 My interaction with this product was clear and understandable.	6,6						=	≡≡
10 I found this product to be flexible to interact with.	7,0							≡≡≡
11 It was easy for me to become skillful at using the system.	6,8						--	≡≡
12 I found the system easy to use.	6,6						=	≡≡
	USEFULNESS	6,5						
	EASE OF USE	6,8						

FIGURA 11: Resultados de los cuestionarios aplicados