



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
DEPARTAMENTO DE SISTEMAS ESPECIALIZADOS EN LA SALUD

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

**Título: Esquematización de la información familiar para la
detección de problemas hereditarios en el módulo Genética
del Sistema de Evaluación del Neurodesarrollo en Niños.**

Autor: Asiel Hernández Martínez

Alexei Pineda Caraballo

Tutor: Ing. Juan Miguel Peña Cabrera

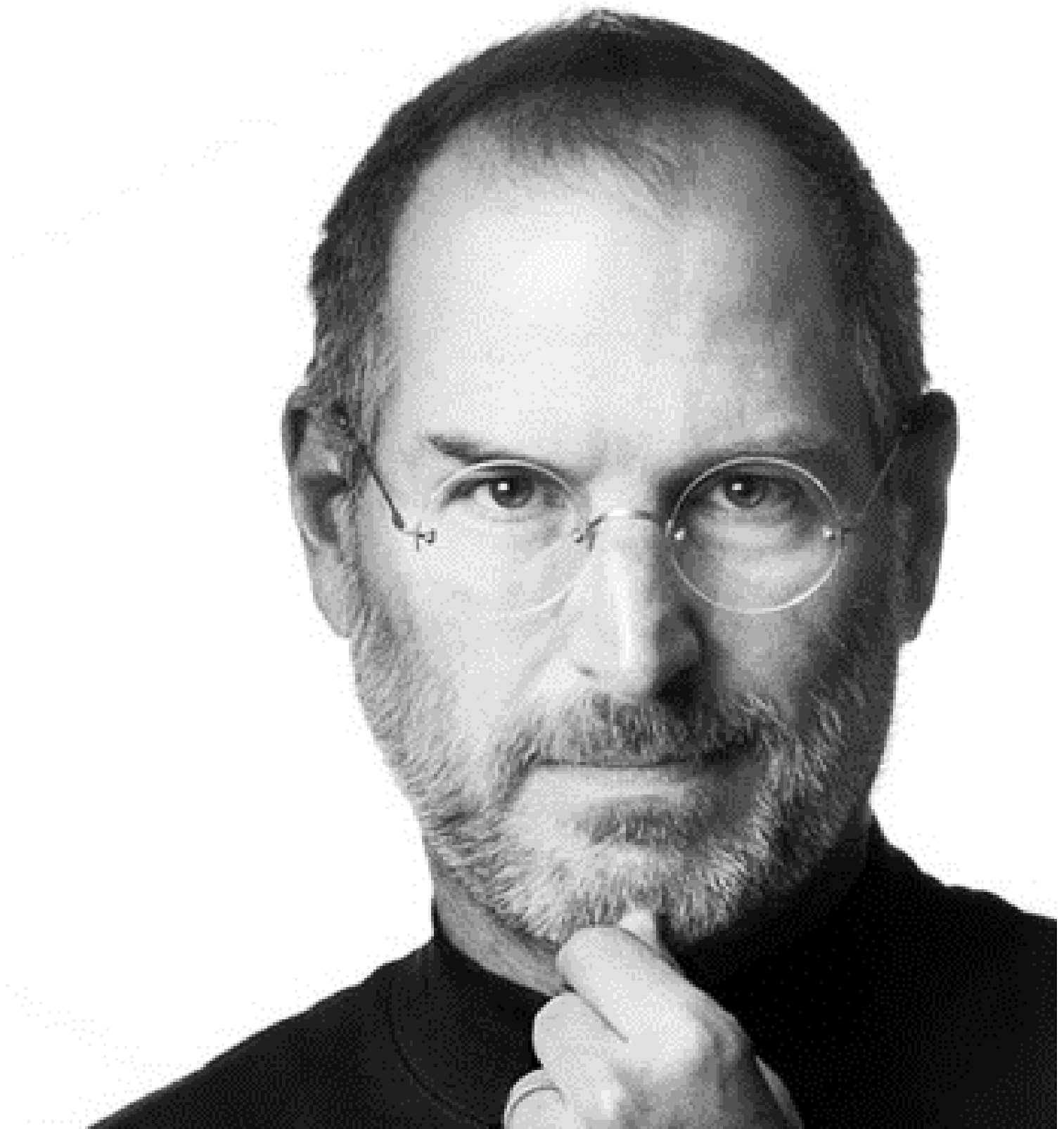
Ing. Dayanna Hernández Pérez

La Habana

Junio 2013

“Año 55 de la Revolución”

Pensamiento



“Tu tiempo es limitado, así que no lo malgastes viviendo la vida de otra persona [...] No dejes que el ruido de las opiniones de otros apague tu propia voz interior”

Steve Jobs

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los ___ días del mes de _____ del año _____.

Asiel Hernández Martínez

Firma del autor

Alexei Pineda Caraballo

Firma del autor

Ing. Juan Miguel Peña Cabrera

Firma del tutor

Ing. Dayanna Hernández Pérez

Firma del tutor

Datos de Contacto

Ing. Juan Miguel Peña Cabrera: Profesor con categoría de Instructor, imparte clases de Sistemas Operativos en la Facultad 7, a estudiantes de 3er año. Es miembro del proyecto Salud Ambiental del Dpto. Sistemas de Apoyo a la Salud del Centro de Informática Médica, donde trabaja como desarrollador del proyecto.

Correo: jmpena@uci.cu

Ing. Dayanna Hernández Pérez. Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2011. Recién graduada en adiestramiento. En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM) y específicamente trabaja en el desarrollo del proyecto SENDN donde se desempeña jefa de proyecto.

Correo: dhdez@uci.cu

Resumen:

En el Departamento Sistemas Especializados en Salud (SES) perteneciente al Centro de Informática Médica (CESIM), se encuentra el proyecto Sistema de Evaluación del Neurodesarrollo en Niños (SENDN), su principal objetivo es informatizar el programa “Renacer Contigo” (PRC) desarrollado en el Hospital William Soler. En la consulta de genética del mismo, se atienden entre otros, los niños del PRC que tienen de cero a cinco años y son egresados de la unidad de terapia intensiva.

Actualmente el SENDN no posee dentro de su módulo Genética, funcionalidades que le permitan identificar problemas de salud, genéticos y hereditarios, basados en la información recopilada del paciente. Es por ello que el objetivo de la investigación es desarrollar las funcionalidades para la esquematización de la información de familiogramas, facilitando el proceso de diagnóstico de los pacientes. Para el desarrollo del sistema se utilizó la Plataforma Java Enterprise Edition 5.0, como Sistema Gestor de Base de Datos (SGBD) PostgreSQL 8.4 y el marco de trabajo Jboss Seam 2.1.1 para la lógica del negocio.

Con el desarrollo de estas funcionalidades se espera brindarles a los genetistas del hospital William Soler, la posibilidad de facilitar tanto el proceso de confección de familiogramas, como el proceso de diagnóstico de los pacientes. También tendrán la posibilidad de desechar algunas de las hipótesis respecto a una enfermedad de un paciente en estudio, evitando así un gasto considerable de recursos en la realización de pruebas diagnósticas más caras. Las pruebas unitarias realizadas sobre el sistema arrojan un correcto funcionamiento del mismo.

Palabras claves: familiograma, genética, esquematización, gestión, sistema.

Índice

Introducción	1
Capítulo 1: Fundamentación teórica de la investigación.	6
1.1. Marco teórico.....	6
1.2. Conceptos asociados a la investigación.....	6
1.3. Descripción de los procesos de negocio	9
1.4. Análisis crítico de ejecución de los procesos actuales.....	9
1.5. Objeto de automatización.....	9
1.6. Beneficios esperados.....	10
1.7. Sistemas informáticos existentes en el ámbito internacional y nacional.	10
1.7.1. Sistemas internacionales:	10
1.7.2. Sistemas nacionales:	12
1.8. Metodología, tecnologías y herramientas.....	15
1.8.1. Metodología de desarrollo de software.....	16
1.8.2. Notación para el modelado del proceso de negocio	16
1.8.3. Marcos de trabajo y librerías.	17
1.8.4. Lenguajes.	20
1.8.5. Servidor de aplicaciones.	22
1.8.6. Sistema gestor de base de datos (SGBD).....	23
1.8.7. Herramientas.....	24
Capítulo 2. Características del sistema propuesto.	28
2.1. Modelo del negocio.....	28
2.1.1. Identificación de roles del entorno del negocio.	28
2.1.2. Diagrama de procesos del negocio.....	29
2.1.3. Descripción del proceso de negocio.	30
2.2. Especificación de requisitos del sistema.....	31
2.2.1. Requisitos funcionales.....	32
2.2.2. Requisitos no funcionales.....	33
2.3. Modelo de casos de uso del sistema.	38
2.3.1. Diagrama de casos de uso del sistema.	39
2.3.2. Descripción de casos de uso del sistema.	40
2.3.3. Especificación de casos de usos.	40
Capítulo 3 Diseño del sistema propuesto.....	43
3.1. Patrones de arquitectura y diseño.....	43
3.1.1. GRASP	44

Índice

3.2. Modelo de diseño.	45
3.2.1. Diagrama de clases del diseño.	46
3.2.2. Clases comunes. Descripción.	47
3.2.3. Descripción de caso de uso adicionar abuelos.	51
3.2.4. Descripción de las clases del diseño.	53
3.3. Modelo de datos.	54
3.4. Descripción de las tablas del modelo de datos.	54
Capítulo 4: Implementación del sistema propuesto.	57
4.1. Modelo de Implementación.	57
4.1.1. Diagrama de componentes.	58
4.1.2. Diagrama de despliegue.	60
4.2. Tratamiento de errores.	61
4.3. Seguridad.	62
4.4. Estrategias de codificación. Estándares y estilos a utilizar.	63
4.4.1. Estándares de Diseño.	65
4.5. Descripción de interfaces del sistema.	67
4.6. Pruebas de validación del sistema.	71
4.6.1 Pruebas de Caja Negra	72
4.6.1.1 Pruebas de Caja Negra aplicada a los componentes implementados.	72
Conclusiones	75
Recomendaciones	76
Referencias Bibliográficas	77
Bibliografía.	80
Anexos	83
Glosario de términos.	98

Introducción

En Cuba, en las últimas décadas la informática se ha desarrollado como uno de los fenómenos más importantes en las diferentes esferas de la sociedad, asociado directamente a la práctica masiva de las Tecnologías de la Información y las Comunicaciones (TIC). Si bien la informática ha realizado cambios revolucionarios en muchas áreas y profesiones, la medicina es una de las que más se ha visto beneficiada con los avances vinculados con la tecnología, ejemplo de esto es el uso de soluciones informáticas para apoyar la interpretación de los resultados mejorando así la calidad del diagnóstico.

Muchas de las transformaciones educacionales y sociales realizadas en Cuba formaban parte del profundo y novedoso programa de la Batalla de Ideas, a partir del cual se emprendieron tareas destinadas a elevar el nivel cultural de la población y su calidad de vida. En estas circunstancias surge la idea de convertir el territorio que ocupaba la base rusa, en la Universidad de Ciencias Informáticas (UCI), como idea del Comandante en Jefe Fidel Castro, con el objetivo de producir soluciones informáticas y brindar servicios informáticos a partir de la vinculación estudio–trabajo como modelo de formación, así como servir de soporte a la industria cubana del Software, como expresara Fidel: “La idea es convertir la informática en una de las ramas más productivas y aportadoras de recursos para la nación. Es el empleo a fondo de la inteligencia y del capital humano que tenemos y principalmente del que podemos crear casi como espina dorsal de la economía”. (1)

En la Facultad 7 de la UCI se encuentra el Centro de Informática Médica (CESIM), el cual tiene la misión de desarrollar sistemas informáticos para gestionar la información del Sistema Nacional de Salud (SNS), también productos, servicios y soluciones informáticas para la optimización del trabajo y mejoramiento de la calidad de la atención médica, contribuyendo a la formación integral de profesionales y lograr incluirse en el mercado nacional e internacional. En el Departamento Sistemas Especializados en Salud (SES), perteneciente al CESIM, existe el proyecto: Sistema de Evaluación del Neurodesarrollo en Niños (SENDN), con el objetivo de informatizar el Programa “Renacer Contigo” (PRC) desarrollado en el Hospital Pediátrico Universitario “William Soler”.

El PRC tiene como objetivos generales: evaluar la calidad del neurodesarrollo de los niños de cero a cinco años de edad egresados de las Unidades de Terapia Intensiva Polivalente y Neonatal (UCIP/UCIN), determinar grupos de tratamientos mediante la clasificación de Zuluaga, valorar la efectividad del Programa de Atención Temprana en dependencia de los

Introducción

grupos de tratamientos y estandarizar protocolos de trabajo de acuerdo a los hallazgos encontrados.

La evaluación es realizada por un equipo interdisciplinario, quienes determinan la presencia o no de afecciones del neurodesarrollo tras su período de gravedad. Este equipo está compuesto por especialistas en: Fisiatría, Neurología, Nutrición, Neurofisiología, Logopedia, Psicología y Genética. (2)

Particularmente la genética es el campo de la biología que busca comprender la herencia biológica que se transmite de generación en generación. Su estudio permite comprender qué es lo que ocurre exactamente en el ciclo celular de los seres vivos y cómo entre seres humanos se transmiten características biológicas y físicas, además de caracteres bioquímicos y de comportamiento de padres a hijos.

La genética puede aplicarse en muchos sectores, pero el más trascendente es la medicina. En la misma el campo de aplicación es amplio y variado, incluye diferentes áreas individuales tales como: la genética clínica, bioquímica, molecular y de enfermedades comunes.

En las consultas de Genética del PRC, los especialistas realizan un estudio exhaustivo del paciente compuesto por: examen físico, historia reproductiva de la madre, estudios prenatales de la madre, la historia perinatal y desarrollo psicomotor. Después de analizados los resultados obtenidos en el estudio realizado al paciente, el genetista emite su impresión diagnóstica y le indica exámenes complementarios, en caso de ser necesarios, para arribar al diagnóstico definitivo del paciente; conformando así la evaluación genética del paciente. Para facilitar el trabajo de los especialistas se desarrolló el módulo Genética.

Actualmente el sistema SENDN no cuenta dentro de su módulo Genética, con funcionalidades que le permitan identificar problemas de salud, genéticos y hereditarios, basados en la información recopilada del paciente. Debido a esto la representación gráfica del árbol genealógico de cada paciente se realiza de forma manual, siendo esto un trabajo engorroso e implicando una gran pérdida de tiempo al especialista. Al no encontrarse esquematizada la información recopilada por el SENDN de los miembros de la familia, la identificación de riesgos de carácter biológicos (problemas hereditarios) puede resultar un tanto complicada para el genetista en la elaboración de la evaluación genética del paciente.

La realización de este árbol genealógico es uno de los primeros pasos para llevar a cabo el diagnóstico de enfermedades genéticas. En este caso consiste en la representación gráfica

de los datos recogidos en la entrevista familiar. La misma facilita la identificación de síndromes genéticos y el establecimiento de diagnósticos presintomáticos, lo que permite un mejor cálculo del riesgo (recurrencia u ocurrencia) y los patrones de herencia de una enfermedad, es decir, permitirá conocer la probabilidad de tener una enfermedad o de heredarla. Solamente con esta herramienta pueden desecharse algunas de las hipótesis posibles respecto a la enfermedad del paciente en estudio, evitando así un gasto considerable de recursos en la realización de pruebas diagnósticas más caras.

Dada la situación anterior, se define el siguiente **problema a resolver** ¿Cómo sintetizar la información familiar para el apoyo al diagnóstico de enfermedades hereditarias o de aparición familiar?

En correspondencia con el problema, el **objeto de estudio** está enfocado al proceso de gestión de la información de familiogramas para la genética médica, de manera que el **campo de acción** se enmarca en el proceso de gestión de la información de familiogramas para el SENDN.

Para solucionar los problemas mencionados, se define como **objetivo general** del presente trabajo desarrollar la funcionalidad de esquematización de la información familiar a través del uso de familiogramas en el módulo de genética del Sistema de Evaluación del Neurodesarrollo en Niños.

Con la intención de alcanzar el objetivo propuesto se han trazado las siguientes **tareas a desarrollar**:

- Análisis crítico de cómo se ejecutan actualmente los procesos, las causas que originan la situación problemática y las consecuencias.
- Análisis crítico y valorativo de sistemas existentes a nivel nacional e internacional, que desarrollen estos procesos.
- Caracterización de la arquitectura definida por el Departamento Sistema de Gestión Hospitalaria.
- Elaboración del modelado del negocio.
- Especificación de los requisitos funcionales y no funcionales del sistema.
- Elaboración del modelo de casos de uso del sistema.
- Especificación de las estrategias de codificación, los estándares y estilos a utilizar.
- Elaboración del diseño de las funcionalidades.

Introducción

- Elaboración del modelo de datos.
- Elaboración del diagrama de despliegue.
- Elaboración del modelo de implementación.
- Implementación, aplicando las pautas de diseño, de los procesos para la realización del familiograma.

El conocimiento de la metodología es de gran utilidad para la investigación, pues permite seleccionar la perspectiva adecuada al problema planteado y dependerá de la finalidad y objetivos que se propongan. Los métodos son el conjunto de operaciones ordenadas con que se pretende resolver un problema y obtener un resultado. En el presente trabajo se utilizarán los métodos teóricos, que es el nivel de conocimiento que se obtiene sobre el objeto, a partir de la experiencia, mediante el pensamiento abstracto y los empíricos, que su contenido surge de la experiencia con cierta elaboración racional.

Teóricos:

- Histórico – Lógico: se realizó un estudio del surgimiento y uso de los familiogramas, en las diferentes ramas de la medicina, en especial la genética médica.
- Análisis y síntesis: al estudiar las características de la esquematización de familiogramas se analizaron las características generales de estos y las relaciones esenciales entre ellas.
- Inductivo – Deductivo: la investigación de sistemas existentes que gestionan la realización de familiogramas, determinará si estos no son suficientes para resolver el problema.
- Modelación: mediante la modelación de los artefactos correspondientes al diseño e implementación se generarán los diagramas correspondientes a dichas fases.

Empíricos:

- Entrevista: se llevarán a cabo entrevistas con los especialistas de genética del hospital “William Soler”, con el objetivo de lograr un entendimiento entre las partes médica e informática. Logrando una mayor comprensión del negocio.
- Observación: permitirá conocer la realidad mediante la percepción directa de los objetos y fenómenos existentes en la consulta de genética. La posibilidad de ver directamente cómo es que se realiza el proceso a informatizar permitirá lograr una

mejor comprensión del mismo. La observación puede utilizarse en distintos momentos de una investigación con el objetivo de acercar la parte informática a la médica.

Con el propósito de organizar y darle una estructura al trabajo se ha decidido dividir el mismo en cuatro capítulos:

Capítulo 1: Fundamentación teórica de la investigación. En este capítulo se realiza un estudio del estado del arte y del proceso de gestión de la información de familiogramas para la genética. Además, se describen la metodología, herramientas y tecnologías utilizadas en la investigación.

Capítulo 2. Características del sistema propuesto: En este capítulo se expone un marco conceptual asociado a la información que será manipulada por el sistema. Se arriba a un acuerdo sobre las funcionalidades, los requerimientos no funcionales y la estrategia de integración a otros módulos.

Capítulo 3. Diseño del sistema propuesto: En este capítulo se describe todo lo relacionado al flujo de trabajo del Diseño. Se centra en la modelación detallada y la construcción de la estructura de la aplicación; así como en la arquitectura y los patrones a utilizar para el diseño e implementación del sistema y se representa el modelo de datos.

Capítulo 4. Implementación del sistema: En este capítulo se describe, de acuerdo a la arquitectura propuesta por el diseño, el modelado de los artefactos correspondientes al flujo de implementación. Se modelan los diagramas de componentes y de despliegue. También se muestran los estándares de diseño y codificación, se explica cómo se realiza el tratamiento de errores y la seguridad del sistema.

Capítulo 1: Fundamentación teórica de la investigación.

El presente capítulo aborda los elementos que permitieron realizar una correcta caracterización de las soluciones existentes a nivel internacional, nacional y en la UCI, sustentados en un estudio de los conceptos fundamentales a los que está relacionado. Se ofrece un estudio de las principales herramientas, plataforma de desarrollo y metodología de desarrollo de software que son sugeridas. Se estudian los diferentes patrones empleados en el desarrollo de soluciones informáticas de este tipo.

1.1. Marco teórico.

El familiograma conocido también como genograma, árbol genealógico o árbol familiar fue desarrollado en 1978 por Murray Bowen, como una herramienta de evaluación para articular patrones multigeneracionales. Fue popularizado en el ámbito clínico por Monica McGoldrick y Gerson Randy con la publicación de su libro, "Genogramas: Evaluación e Intervención" en 1985. En septiembre de 1986 durante el IV Simposium Internacional de Medicina Familiar se presentó el trabajo "Familiograma" en el que se propone la simbología adoptada. Actualmente se usan en varios campos dentro de los cuales se encuentran: la medicina, la psicología, el trabajo social, la investigación genética y la educación.

Los familiogramas aportan gran beneficio para la medicina por su utilidad en la terapia familiar. Permite a los doctores identificar problemas de salud, genéticos y hereditarios, basados en la información recopilada. También ayudan al paciente a entender de una manera simple su propia dinámica familiar. Además son útiles en la identificación de patrones predominantes de comportamiento o problemas familiares como el divorcio, la alienación entre los hermanos, suicidios o alcoholismo.

1.2. Conceptos asociados a la investigación.

Familiograma: Se define al familiograma como una herramienta que permite valorar la dinámica, la composición, estructura, el tipo de familia, relaciones, los roles que asumen y el ciclo evolutivo por el cual está cursando la familia en un momento determinado. Permite a los médicos y al grupo de salud identificar problemas de salud genéticos y hereditarios, basados en la información recopilada (3). Mediante el familiograma es posible almacenar los datos de la familia, los cuales dan la oportunidad de hacer una retro-información básica (nombres, fechas de nacimiento, muerte, matrimonio, entre otros) y así mismo información compleja (problemas familiares repetitivos, triangulaciones, herencia y otros).

Capítulo 1: Fundamentación teórica de la investigación.

Árbol genealógico: Un árbol genealógico es una representación gráfica que incluye los descendientes y los antepasados de un individuo en una forma organizada y sistemática, sea en forma de árbol o tabla. Su composición puede realizarse de forma ascendente para el caso en que se vaya a representar a los antepasados de un individuo o de forma descendiente para representar todos sus descendientes.

Composición del familiograma

- Para realizar un familiograma deben seguirse tres pasos fundamentales:
 - Trazado de la estructura familiar.
 - Registro de la información de la familia.
 - Señalamiento de las relaciones familiares

El trazado de la estructura se realiza expresando a través de símbolos las relaciones biológicas y legales entre los miembros de la familia. Se utilizan círculos y cuadrados para representar a los miembros de la familia y con líneas se representan las relaciones que existen entre ellos, (ver figura 1) y (ver figura 2).



Figura. 1 Hombre



Figura. 2 Mujer

Las relaciones de parejas legalmente establecidas se trazan con líneas continuas, (ver figura 3) y las relaciones de parejas no formalizadas legalmente (Concubinato) se expresan con líneas discontinuas, (ver figura 4).

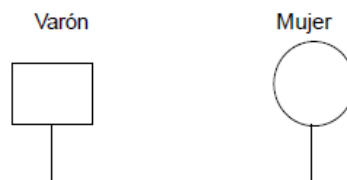


Figura. 3 Relación de pareja.

Capítulo 1: Fundamentación teórica de la investigación.

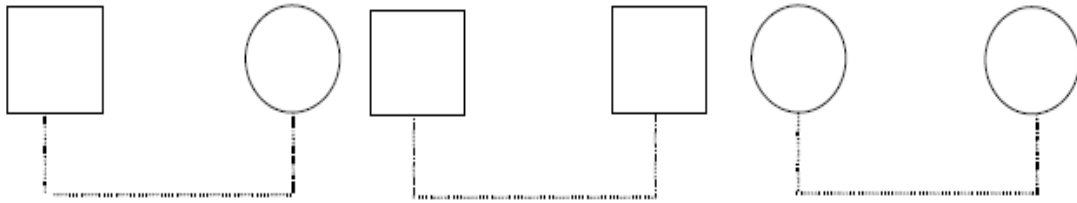


Figura. 4 Relación de parejas no formalizadas legalmente.

SENDN: El proyecto Sistema de Evaluación del Neurodesarrollo en Niños (SENDN) es un sistema basado en el programa “Renacer Contigo” del Hospital Pediátrico Universitario “William Soler”, desarrollado como proyecto de investigación y aprobado por la Dirección Nacional Materno Infantil. Este proyecto fue desarrollado por el Grupo Interdisciplinario de Atención Temprana perteneciente a dicho hospital. Estos fueron motivados por la influencia que sobre la evolución del niño en riesgo ejercen los programas de atención temprana utilizados hace ya varios años a nivel mundial; además de la pobre presencia de estos programas en los servicios de atención al grave en el país y la necesidad cada vez mayor de mejorar la calidad de vida de la población infantil en el país, ya con tasas de mortalidad comparables con los países desarrollados.

Programa Renacer Contigo: El programa “Renacer Contigo” del Hospital Pediátrico Universitario “William Soler”, desarrollado como proyecto de investigación y aprobado por la Dirección Nacional Materno Infantil. Este proyecto fue desarrollado por el Grupo Interdisciplinario de Atención Temprana perteneciente a dicho hospital. Estos fueron motivados por la influencia que sobre la evolución del niño en riesgo ejercen los programas de atención temprana utilizados hace ya varios años a nivel mundial; además de la pobre presencia de estos programas en los servicios de atención al grave en el país y la necesidad cada vez mayor de mejorar la calidad de vida de la población infantil en el país, ya con tasas de mortalidad comparables con los países desarrollados.

Se lleva a cabo mediante este programa un proceso de evaluación de la calidad del neurodesarrollo en los niños de cero a cinco años el cual tiene los siguientes objetivos:

1. Detectar posibles enfermedades que el niño puede heredar de sus antecesores.
2. Evaluar y manejar síntomas frecuentes.
3. Anticipar o prevenir problemas médicos.
4. Reconocer, diagnosticar, realizar consejería o registrar problemas genéticos.

Capítulo 1: Fundamentación teórica de la investigación.

En aras de darle cumplimiento a lo antes planteado el hospital cuenta con un equipo de profesionales de varias disciplinas que evalúa la información de la familia registrada en el familiograma y a través de esta se emite un diagnóstico de posibles padecimientos que pueden ser transmitidos al niño.

1.3. Descripción de los procesos de negocio

El médico para obtener una evaluación del paciente, primeramente analiza el estado del niño y la información que se ha registrado con anterioridad de sus antecesores hasta la tercera generación, luego identifica problemas de salud, genéticos y hereditarios, basados en la información recopilada. Cuando termina de realizar esta búsqueda valora cuáles son las enfermedades que han presentado los antecesores del niño que pueden ser transmitidas y emite su impresión médica.

1.4. Análisis crítico de ejecución de los procesos actuales.

La recogida de datos de pacientes para gestionar familiogramas en el programa “Renacer Contigo” del Hospital Pediátrico “William Soler” es realizada de forma manual. Esto puede conllevar a la pérdida de la información, su deterioro y también la introducción de errores humanos.

En cada consulta el especialista recoge un volumen de datos que en ocasiones puede llegar a ser bastante grande, por lo que el proceso de atención de un paciente puede llegar a ser lento e impedir que puedan atenderse todos los pacientes planificados para ese día. Otro elemento que puede influir en esto es el proceso de búsqueda de información relacionada con un determinado paciente para realizar alguna comparación o estudio genético, ya que también puede demorarse debido a la gran cantidad de datos recogidos y almacenados en papeles y expedientes.

1.5. Objeto de automatización.

Para darle solución a los problemas anteriormente mencionados con respecto al proceso de gestión de familiograma en la consulta de Genética, se propone desarrollar funcionalidades que permitan gestionar toda la información esquematizada en los familiogramas, posibilitando así una mayor confidencialidad, seguridad y control de la información que se manipula, así como mejorar la realización de los diagnósticos de los pacientes.

1.6. Beneficios esperados.

Con el desarrollo de la aplicación se espera solucionar los problemas anteriormente planteados e informatizar los procesos del negocio descritos. Al desplegar el sistema se persigue lograr una atención rápida y profesional a los pacientes del SENDN y la realización de familiogramas de manera automática, permitiendo consultar en tiempo real la información familiar reflejada en los mismos. El sistema al ser realizado en la Universidad de las Ciencias Informáticas, garantiza soporte técnico de forma gratuita.

1.7. Sistemas informáticos existentes en el ámbito internacional y nacional.

Con el desarrollo de la informática, las comunicaciones y las nuevas tecnologías, el sector de la salud no se ha quedado ajeno al proceso de incorporación de las TIC, con el objetivo de mejorar la atención médica a la población. En el mundo son múltiples las empresas que desarrollan soluciones informáticas para gestionar toda la información procedente desde los diferentes niveles de atención médica. La realización de familiogramas constituye una de las principales vías en la medicina familiar para el estudio de la familia, esto ayuda a comprender tanto aspectos estructurales como las relaciones de la familia en estudio. Actualmente existen algunos programas que ayudan a realizar estos gráficos con mayor facilidad. A continuación se analizan algunos de los sistemas existentes tanto a nivel internacional como nacional que están orientados a la creación de familiogramas.

1.7.1. Sistemas internacionales:

Dentro de los sistemas estudiados en el ámbito internacional podemos citar dentro de los más usados actualmente los siguientes:

GenoPro: Se trata de una aplicación avanzada que ofrece varias funcionalidades válidas para el campo de la genealogía. Permite visualizar un gráfico completo de un árbol genealógico, y ver individualmente el registro de cada persona, tanto los antecesores como los descendientes, (ver figura 5)

Los informes se pueden editar y modificar una vez que se hayan generado. También es posible recopilar estadísticas para encontrar posibles errores en algunos registros de un árbol genealógico. Brinda soporte para relaciones personales más complejas (no solamente parentescos directos) (4).

Los autores de la investigación determinan que una de las principales desventajas que presenta este sistema, está precisamente en que ha sido diseñado únicamente para funcionar

Capítulo 1: Fundamentación teórica de la investigación.

en plataforma Windows además, de que es privativo. Este sistema solamente permite darle seguimiento a un pequeño grupo de las enfermedades que puede presentar un paciente, desechando información valiosa para un posterior análisis de la familia.

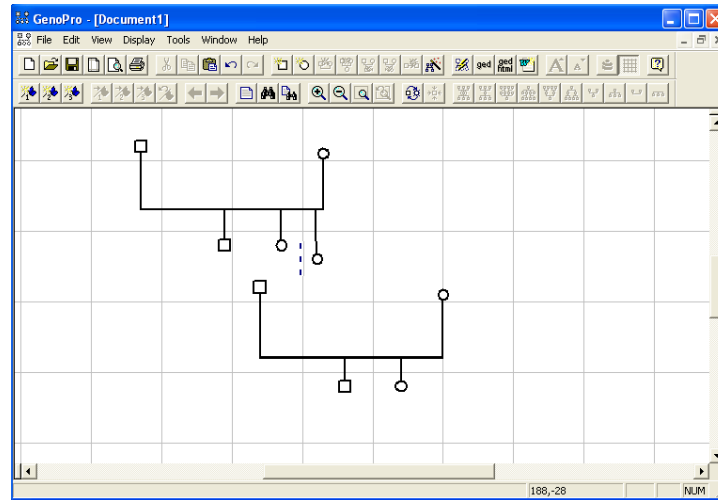


Figura. 5 Sistema GenoPro

WinGeno: es una solución informática utilizada en la realización de diagramas de las relaciones familiares en forma de genogramas donde la representación de los datos de cada miembro de la familia se realiza con símbolos establecidos, (ver figura 6).

Permite a los usuarios representar a un miembro de su familia, masculino o femenino y a ese miembro añadirle los padres o hijos. Con la ayuda del "Instrumento de conexión" se pueden dibujar alianzas entre los miembros de la familia incluidos en el genograma. (5)

Esta aplicación a consideración de los autores de la investigación no es la más acertada para el problema presentado ya que no permite ver los padecimientos de los pacientes ni darle seguimiento a las enfermedades predominante en una determinada familia, lo que dificulta la toma de decisiones de los profesionales de la salud que la utilicen.

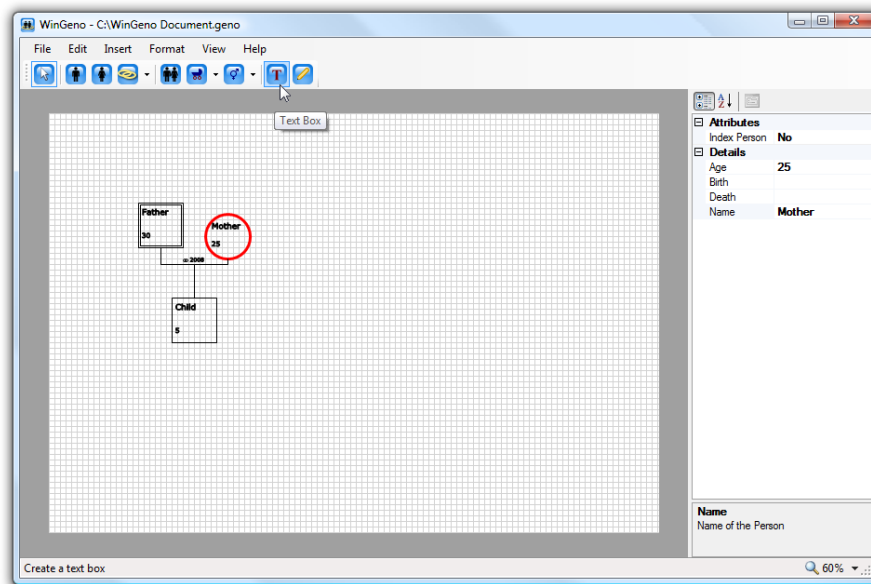


Figura. 6 Sistema WinGeno

1.7.2. Sistemas nacionales:

AlasARBOGEN: Se trata de una aplicación desarrollada sobre el marco de trabajo .Net en la Universidad de las Ciencias Informáticas en conjunto con el Centro Nacional de Genética Médica (CNGM), (ver figura 7). Permite la representación del árbol genealógico de un paciente. También recoge los datos del paciente y de sus familiares. Esta aplicación facilita la realización de operaciones como inserción y modificación de datos y de individuos, las relaciones entre cada uno de los individuos del árbol y las enfermedades que padecen. (6)

Utiliza el software Microsoft Word o el Bloc de notas para exportar un reporte con los datos de todos los individuos representados en el árbol genealógico. Dentro de las principales funcionalidades del sistema se encuentran.

- Gestionar gráficamente un individuo.
- Gestionar relaciones entre individuos.
- Gestionar símbolos.
- Gestionar enfermedades.
- Gestionar familia.

Uno de los aspectos que limita su aplicación como solución al problema de la investigación es que está implementada sobre .Net no es multiplataforma, por lo que solo funciona en el sistema

Capítulo 1: Fundamentación teórica de la investigación.

operativo Windows. Además que no utiliza base de datos para gestionar la información asociada a los árboles genealógicos.

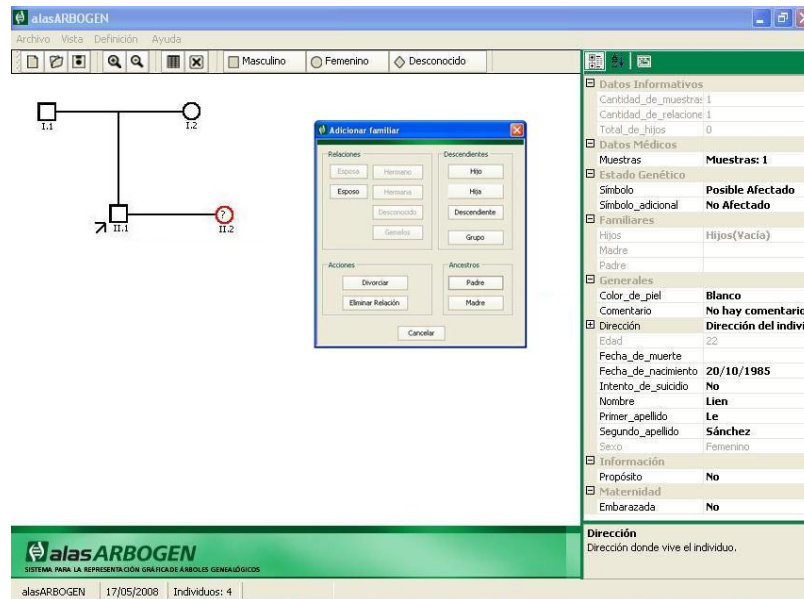


Figura. 7 Aplicación AlasARBOGEN

AlasARBOGEN 2.0: Es una aplicación de escritorio multiplataforma destinada a la representación gráfica de los árboles genealógicos, (ver figura 8). Uno de sus principales objetivos es continuar optimizando las funcionalidades de la aplicación AlasARBOGEN. La aplicación permite hacer una representación del árbol genealógico de un paciente. Recoger la información de interés para el estudio genético perteneciente al individuo, a la familia, así como a las relaciones de pareja y datos de interés que puedan aportar información útil para el estudio del caso. Presenta una mejora en la gestión visual de los individuos, así como en las operaciones de inserción y actualización de la información representada en el árbol genealógico. El manejo de las relaciones existentes entre los individuos del árbol genealógico y la simbología para la representación de las enfermedades genéticas se realiza de una forma más cómoda, al igual que la gestión de las muestras de laboratorio.

Como nuevas funcionalidades automatizadas con respecto a AlasARBOGEN se encuentran que presenta una vista previa de impresión y un servicio de reporte en el formato pdf. Se garantiza la persistencia de los datos en una BD centralizada, así como la persistencia de los datos en archivos binarios en la PC. Se informatiza el proceso de exportación del árbol genealógico que se está estudiando en formato jpg. Permitirá realizar búsquedas internas en el árbol genealógico atendiendo a diferentes criterios según sea de interés para el genetista y el

Capítulo 1: Fundamentación teórica de la investigación.

ordenamiento gráfico del árbol genealógico atendiendo a la organización generacional que este presenta. (7)

A pesar de tener nuevas funcionalidades automatizadas con respecto a las otras aplicaciones, no incluye la opción de "Rehacer" y "Deshacer" los cambios que se realicen en los árboles genealógicos por lo que una vez realizada una acción en el sistema no se puede revertir, esto implica tener que comenzar nuevamente la construcción del árbol genealógico.

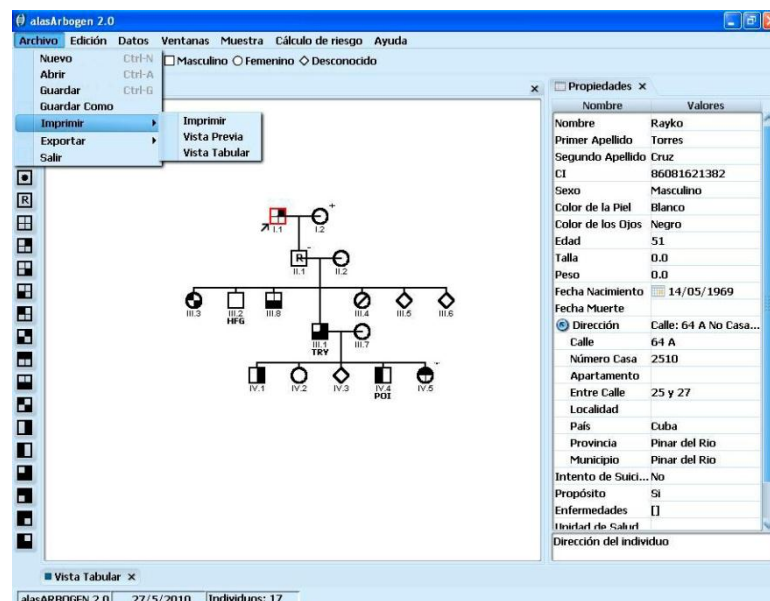


Figura. 8 Aplicación AlasARBOGEN 2.0

Otra aplicación es el Sistema Integral para la Atención Primaria de Salud (SIAPS) desarrollado por el Departamento de Atención Primaria de Salud (APS) del Centro de Informática Médica (CESIM) de la UCI donde se desarrolló en el 2011 un componente web que facilita la gestión de la información de familiogramas.

Como características generales se identifican que está basado en tecnologías libres, es multiplataforma y de código abierto. Se utiliza la Plataforma Java Enterprise Edition 5.0, además como Sistema Gestor de Base de Datos (SGBD) se tiene PostgreSQL 8.4, Hibernate 3.3 como herramienta ORM para la persistencia de los datos y el framework Jboss Seam 2.1.1

Como características específicas de este componente se identifican que todos los miembros de una familia son pacientes del Sistema de Información Hospitalaria (HIS), se genera una representación esquemática de la familia, que provee información de los miembros de esta en cuanto a su estructura y relaciones. A pesar de que en materia de funcionalidades automatizadas se considera que en este componente son más generales que en los anteriores

Capítulo 1: Fundamentación teórica de la investigación.

mencionados, los autores de la investigación consideran que no es del todo acertado la aplicación de este para la solución que requiere el problema presentado, ya que solo se limita a establecer relaciones de consanguinidad, adopción, matrimonio, afinidad u otra relación estable de afectividad. Se trata, por tanto, de vínculos que pueden ser biológicos o no y que se organizan de acuerdo a líneas que permiten medir diversos grados de parentesco, no se tiene en cuenta las tres generaciones de parentesco de una persona como: abuelos maternos y paternos, tíos, hermanos, primos, por lo que no se permite darle seguimiento a enfermedades que se transmiten de una generación a otra.

Valoración de las soluciones existentes

En la valoración realizada por los autores de la investigación de las soluciones existentes a nivel internacional para gestión de familiogramas, identifican que no cumplen en su totalidad con las normas establecidas para la informatización del SENDN. Las soluciones existentes en el ámbito nacional no tienen incorporado una herramienta que facilite el análisis de la estructura familiar, ni tampoco permiten darle seguimiento a enfermedades que se transmiten de una generación a otra. Por estas razones surge la necesidad de incorporar funcionalidades nuevas para gestionar el familiograma del módulo Genética de este departamento y facilitar a los profesionales de la salud la toma de decisiones y el intercambio de información.

A pesar de las desventajas que poseen las soluciones anteriormente mencionadas, se pudo observar que estas ofrecen varias funcionalidades que pueden ser utilizadas en el desarrollo de la solución que se propone en esta investigación para la realización de un familiograma que cumpla con todos los requerimientos necesarios para el módulo Genética del SENDN, dentro de las que se proponen incorporara se la que gestiona los datos de los familiares y el uso de símbolos para representar los miembros de la familia.

1.8. Metodología, tecnologías y herramientas.

En la actualidad, las diferentes tecnologías y herramientas de la informática han evolucionado considerablemente, por lo que es de vital importancia para el desarrollo de un sistema informático escoger las tecnologías adecuadas para garantizar que el mismo se realice con una buena calidad y con la mayor facilidad por parte de los desarrolladores. Teniendo en cuenta lo antes mencionado, a continuación se expone la metodología de desarrollo utilizada, así como las herramientas y tecnologías utilizadas para la implementación de las funcionalidades correspondientes para el módulo Genética.

1.8.1. Metodología de desarrollo de software.

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares, que ayudan a los desarrolladores de soluciones informáticas en sus esfuerzos por implementar nuevas aplicaciones. En un proyecto de desarrollo, define quién debe hacer qué, cuándo y cómo hacerlo (8).

En la realización del presente trabajo se utiliza RUP como metodología de desarrollo debido a que está preparada para desarrollar grandes y complejos proyectos, además de ser orientada a objetos. Esta metodología es la utilizada por el SENDN para la generación de todos los documentos, ya que en muchas ocasiones son únicos para todo el proyecto. RUP no es solo un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos (9).

El ciclo de vida de RUP se caracteriza por ser:

Dirigido por casos de uso: en RUP los casos de uso, no son solo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Estos constituyen un elemento integrador y una guía de trabajo. No solo inician el proceso de desarrollo, sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Centrado en la arquitectura: presta especial atención al establecimiento temprano de una buena arquitectura que no se encuentre fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los casos de uso y la forma la proporciona la arquitectura (10).

Iterativo e incremental: el trabajo se divide en partes más pequeñas o mini proyectos, permitiendo que el equilibrio entre casos de uso y arquitectura, se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales), del cual se obtiene un incremento que produce un crecimiento en el producto.

RUP agrupa las actividades en grupos lógicos, definiéndose nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo basándose en UML (11).

1.8.2. Notación para el modelado del proceso de negocio

Capítulo 1: Fundamentación teórica de la investigación.

En el desarrollo de un sistema informático es muy importante realizar un buen modelo de procesos de negocio independientemente de la notación con la que se exprese, en la realización del presente trabajo se utilizó la Notación para el Modelado de Procesos de Negocio (BPMN, por sus siglas en inglés) y a continuación se explica en que consiste así como algunas de sus principales características.

BPMN es un nuevo estándar para modelar flujos de procesos de negocio y servicios web. Su meta principal es definir una notación entendible para todos los usuarios del negocio, desde los analistas del negocio que modelan los procesos hasta los desarrolladores técnicos responsables de implementarlos, garantizando la mejora de los mismos y finalmente a los usuarios del negocio quienes monitorearán y administrarán dichos procesos. BPMN crea un puente estándar entre el diseño del proceso de negocio y su implementación.

BPMN es un facilitador de fondo para una nueva iniciativa en el mundo de la arquitectura empresarial, la Administración de Procesos de Negocio (BPM, por sus siglas en inglés). Esta permite gestionar el cambio para mejorar los procesos de negocio y unifica las disciplinas de Modelación de Procesos, Simulación, Flujo de trabajo, Integración de Aplicaciones Empresariales (EAI, por sus siglas en inglés) y la integración Business-to-Business (B2B) en un único estándar.

Características de BPMN:

- Proporciona un lenguaje gráfico común, con el fin de facilitar su comprensión a los usuarios de negocios.
- Integra las funciones empresariales.
- Utiliza una Arquitectura Orientada por Servicios (SOA, por sus siglas en inglés), con el objetivo de adaptarse rápidamente a los cambios y oportunidades del negocio.
- Combina las capacidades del sistema informático y la experiencia de negocio para optimizar los procesos y facilitar la innovación del negocio (12).

1.8.3. Marcos de trabajo y librerías.

En la actualidad, cada día son más las instituciones y empresas que desean expandirse a Internet, debido a esto los desarrolladores de aplicaciones tienen como reto implementar aplicaciones de calidad en un tiempo reducido. Mediante el uso de marcos de trabajo y librerías es posible la reutilización a gran escala de componentes de software para el desarrollo rápido de aplicaciones. A continuación se muestran cuáles fueron los principales marcos de trabajo y librerías utilizados en la realización del presente trabajo.

Capítulo 1: Fundamentación teórica de la investigación.

Java Server Faces (JSF) 1.2: JSF es un marco de trabajo que permite crear Interfaces de Usuario (UI, por sus siglas en inglés) para aplicaciones web, mediante componentes reutilizables. Permite el manejo de estados y eventos, así como la asociación entre los datos de la interfaz y los datos de la aplicación web. Permite desarrollar rápidamente aplicaciones de negocio dinámicas en las que toda la lógica de negocio se implementa en Java, o es llamada desde Java, creando páginas para las vistas muy sencillas. Resuelve validaciones, conversiones, mensajes de error e internacionalización y permite introducir JavaScript en la página, para acelerar la respuesta de la interfaz en el cliente (13).

JBoss Seam 2.1.1: Seam es un marco de trabajo de código abierto desarrollado por la empresa JBoss, con el fin de unir diferentes tecnologías y estándares de Java, a la vez que añade algunas funcionalidades no contempladas por ellos. Permite usar Enterprise JavaBeans (EJB3) y JSF de una forma muy sencilla, además añade herramientas muy útiles para el desarrollo de aplicaciones web, todo bien acoplado y basado en estándares ampliamente utilizados y probados, escalables, portables y reusables (14).

Facelets 1.1: es un marco de trabajo para plantillas centrado en la tecnología JSF, por lo cual se integran de manera muy fácil (15).

Características de Facelets

- Facilidad en la creación de plantillas para los componentes y páginas.
- Un buen sistema de reporte de errores.
- No es necesaria la configuración del Lenguaje Extensible de Etiquetado (XML).

Hibernate 3.3: es un marco de trabajo de consultas y mapeos objeto-relacional muy potente y de alto rendimiento. Tiene como objetivo facilitar la persistencia de objetos Java en bases de datos relacionales y obtener información de estas. Facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos XML. Proporciona además, un lenguaje para el manejo de consultas a la base de datos similar a SQL y es utilizado para obtener objetos de la BD. Es un modelo de programación natural y una capa de persistencia objeto/relacional que permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, así como un gran número de tipos de datos.

Principales características de Hibernate:

Capítulo 1: Fundamentación teórica de la investigación.

- Facilidades en metadatos: soporta el formato del mapeado de XML, diseñado para ser editado a mano y el mapeado basado en anotaciones. Además de validación basada en anotaciones.
- Puede configurarse y ejecutarse en la mayoría de aplicaciones Java y entornos de desarrollo. Generalmente, se utiliza en aplicaciones cliente-servidor de dos y tres capas, y se despliega únicamente en el servidor.
- Gran escalabilidad: es muy eficiente, tiene una arquitectura de caché de doble capa y podría ser usado en un clúster.
- Software libre: está bajo licencia Lesser GNU Public License (LGPL).
- Su característica más importante es que implementa varios dialectos (16).

Los dialectos en Hibernate son los encargados de encapsular las diferentes formas en que el marco de trabajo puede comunicarse con una base de datos, para lograr tareas como obtener un valor de secuencia o estructurar una petición SELECT.

Ajax4JSF: es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript.

Mediante esta librería se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas y controlar cualquier evento de usuario. Ajax4JSF permite dotar a la aplicación JSF de contenido mucho más profesional con muy poco esfuerzo (17).

Richfaces 3.3.1: es una biblioteca de componentes para JSF para la integración de Ajax con facilidad en la capacidad de desarrollo de aplicaciones de negocio. Richfaces permite definir por medio de etiquetas de JSF diferentes partes de una página JSF que se desee actualizar con una solicitud Ajax, proporcionando así varias opciones para enviar peticiones Ajax al servidor. Además aprovecha al máximo los beneficios de JSF incluyendo, la validación y conversión de instalaciones, junto con la gestión de estática y dinámica de los recursos (18).

Java Persistence API (JPA) 1.0: es la Interfaz de Programación de Aplicaciones (API, por sus siglas en inglés) para la persistencia de objetos Java a cualquier base de datos relacional. Esta API fue desarrollada para la plataforma Java EE e incluida en el estándar de EJB 3.0 (19). Busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la

Capítulo 1: Fundamentación teórica de la investigación.

orientación a objetos al interactuar con una base de datos, como sí ocurría con EJB2 y permitir usar objetos regulares.

Enterprise JavaBeans (EJB) 3.0: EJB es la arquitectura de componentes del lado del servidor para Java Platform Enterprise Edition (Java EE). Permite el desarrollo rápido y sencillo de aplicaciones distribuidas, transaccionales, seguras y portátiles basado en tecnología Java (20).

Los EJB proporcionan un modelo distribuido y estándar de componentes que se ejecutan en el servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

Java Platform Enterprise Edition (JavaEE5): Es una plataforma de programación para desarrollar y ejecutar sistemas informáticos implementados con el lenguaje de programación Java, preferiblemente del lado del servidor. Se caracteriza por tener una arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares, ejecutándose sobre un servidor de aplicaciones. Esta plataforma proporciona a los desarrolladores un potente conjunto de API que reducen el tiempo de desarrollo y la complejidad, al tiempo que mejoran el rendimiento de las aplicaciones. La implementación Java EE puede ser descargada desde la web gratuitamente, y existen disímiles herramientas de código abierto, disponibles para extender la plataforma o para simplificar el desarrollo (21).

Java Runtime Environment (JRE6): Se corresponde con un conjunto de utilidades que permite la ejecución de programas Java sobre todas las plataformas soportadas. El entorno en tiempo de ejecución Java está constituido por una máquina virtual y además, por las librerías de clases estándar que implementan el API asociado. Ofrece la funcionalidad de intermediario entre una aplicación programada en este lenguaje y el sistema operativo que se esté usando. Sirve para que cualquier aplicación pueda funcionar en diferentes sistemas operativos, lo cual es significativo para el desarrollo del sistema en dicho lenguaje (22).

1.8.4. Lenguajes.

En el siguiente epígrafe se muestran cuáles fueron los lenguajes utilizados para el desarrollo de las funcionalidades necesarias. Estos están divididos en diferentes grupos, dígame el lenguaje de programación por parte del servidor, el lenguaje de marcado por parte del cliente y también el lenguaje utilizado para construir y documentar los artefactos del sistema.

Capítulo 1: Fundamentación teórica de la investigación.

Lenguaje Extensible de Marcado de Hipertexto (XHTML) 1.1: XHTML es un lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. Es la versión XML de HTML con las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML. Su objetivo es lograr una web semántica, donde la información y la forma de presentarla estén claramente separadas (23).

Java 1.6: Es un lenguaje de objetos desarrollado por un grupo de ingenieros de Sun Microsystems en 1991, utilizado por Netscape posteriormente como base para JavaScript. A diferencia de los lenguajes convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado a un código intermedio o bytecode, el cual es interpretado por una máquina virtual de Java (24).

Tiene mucho en su sintaxis de C y C++, pero posee un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir muchos errores, como la manipulación directa de punteros a memoria.

Java es un potente y versátil lenguaje de programación que puede trabajar en todo tipo de entornos, desde servidores de aplicaciones middle-tier hasta clientes web. Independientemente del tipo de aplicación que se desarrolle y del tipo de máquina en la que se ejecute el código, la aplicación seguramente tendrá que acceder a datos almacenados en algún tipo de base de datos. Las bases de datos relacionales son la elección obvia a la hora de trabajar con Java debido a sus características.

Una de las principales características de este lenguaje, es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que un mismo programa escrito para Linux puede ser ejecutado en Windows sin ningún problema. Además, es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas.

Partes de la plataforma Java

- Lenguaje: un lenguaje de propósito general que emplea el paradigma de programación orientado a objetos.
- La Máquina Virtual: los programas Java se compilan a un código intermedio interpretado por una máquina virtual Java Virtual Machine (JVM, por sus siglas en inglés) lo que permite su portabilidad.
- Las bibliotecas: conjunto de API que proporcionan herramientas para el desarrollo.

Lenguaje Unificado de Modelado (UML) 2.1: UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de soluciones informáticas (25). Permite la modelación de sistemas con tecnología orientada a objetos. Se puede aplicar en el desarrollo de sistemas informáticos entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso utilizar.

Ventajas de UML

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de diseño e implementación, construyendo así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo.
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

1.8.5. Servidor de aplicaciones.

Un servidor de aplicaciones es un sistema que proporciona los servicios de una aplicación a las computadoras cliente. Este se encarga de gestionar las funciones de lógica del negocio y permite el acceso a los datos de la aplicación.

JBoss Server 4.2.2: JBoss es el servidor de aplicaciones de código abierto más utilizado actualmente. Este servidor de aplicaciones se encuentra certificado con J2EE y soporta sistemas de gran complejidad y alta concurrencia. Al estar basado en Java, puede ser utilizado en cualquier sistema operativo que lo soporte. Implementa todo el paquete de servicios de J2EE.

Características de JBoss

- Producto de licencia de código abierto sin coste adicional.
- Confiable a nivel de empresa.

- Orientado a arquitectura de servicios.
- Flexibilidad consistente.
- Servicios del middleware para cualquier objeto de Java.
- Ayuda profesional disponible. Soporte completo para Java Management Extensions (JMX).

Es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business. Combinando una arquitectura orientada a servicios revolucionaria con una licencia de código abierto, puede ser descargado, utilizado, instalado y distribuido sin restricciones por la licencia. Por este motivo es la plataforma más popular de middleware para desarrolladores, vendedores independientes, así como para grandes empresas.

1.8.6. Sistema gestor de base de datos (SGBD).

Los SGBD son un tipo de sistema informático que sirven de interfaz entre el usuario, la aplicación y la base de datos. El propósito general es gestionar de manera organizada, sencilla y eficaz, la información almacenada en una base de datos para posteriormente ser utilizada por el usuario en un determinado fin. Las operaciones fundamentales son: crear, modificar, eliminar y obtener la estructura asociada al esquema lógico de una BD (26).

PostgreSQL 8.4: es un potente Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS, por sus siglas en inglés) de código abierto. Tiene más de 15 años de estar activo el proceso de desarrollo a nivel mundial y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de los datos, y la corrección. Es multiplataforma y funciona en los principales sistemas operativos, como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. También apoya el almacenamiento de grandes objetos binarios, imágenes, sonidos o vídeo.

PostgreSQL materializa la seguridad en tres aspectos:

- Seguridad en la manipulación de los ficheros.
- Seguridad en el acceso de los clientes.
- Definición de los privilegios para acceder a los objetos de la base de datos a los usuarios.

Características de PostgreSQL:

Capítulo 1: Fundamentación teórica de la investigación.

- Objeto-relacionales: en cada tabla define una clase que permite implementar la herencia entre tablas o clases, funciones y operadores polimórficos.
- Implementación del estándar SQL92/SQL99.
- Código abierto: debido a la licencia liberal, puede ser usado, modificado y distribuido por todo el mundo de forma gratuita para cualquier fin, ya sea de datos, comerciales o académicas.
- Múltiples lenguajes de procedimientos: los disparadores y otros procedimientos pueden ser escritos en varios lenguajes de procedimientos. El código del lado del servidor es comúnmente escrito en PL / PostgreSQL, un lenguaje de procedimiento similar al de Oracle PL / SQL. También se puede desarrollar código del lado del servidor en Tcl, Perl, incluso bash (el de código abierto Linux / Unix shell).
- Múltiples-cliente API: soporta el desarrollo de aplicaciones cliente en varios lenguajes, interfaz para PostgreSQL desde C, C ++, ODBC, Perl, PHP, Tcl / Tk, y Python.
- Tipos de datos: integer, string, numeric, boolean, char, varchar, date, interval, y timestamp y tipos de datos diseñados específicamente para hacer frente a las direcciones de red.
- La extensibilidad: es una de las características más importantes ya que puede ser ampliado, se pueden añadir nuevos tipos de datos, nuevas funciones y operadores, e incluso nuevos lenguajes de procedimiento y de cliente.
- Tiene un soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados.
- Control de Concurrencia Multi-Versión (MVCC), es la tecnología que usa para evitar bloqueos innecesarios, es decir, mientras un proceso escribe no se bloquean las tablas, ni siquiera las tuplas. Mediante el uso de MVCC, se evita el problema de que procesos lectores esperen a que se termine de escribir por completo. MVCC está considerado mejor que el bloqueo a nivel de tupla porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.
- Integridad Referencial: soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos (27).

1.8.7. Herramientas.

Las herramientas de desarrollo son medios mediante los cuales se pueden llevar a cabo las tareas necesarias para la realización de un determinado sistema. Es importante seleccionar las

Capítulo 1: Fundamentación teórica de la investigación.

herramientas correctas ya que de esta forma se ahorra tiempo y recursos en todo el proceso de desarrollo.

Visual Paradigm (VP) 6.4: Es una herramienta CASE que utiliza el UML. Está orientada a la creación de diseños usando el paradigma de Programación Orientada a Objetos (POO). Es una poderosa herramienta para visualizar y diseñar elementos de software. Visual Paradigm incluye una herramienta llamada Visual Architect que permite la generación de código para el manejo de la base de datos.

Con esta herramienta se puede generar código para los lenguajes PHP, JAVA y C# y para los gestores de base de datos DB2, Informix, SQL Server, MySQL, Oracle y PostgreSQL.

Beneficios de Visual Paradigm para UML

- Persistencia de forma fácil.

Los desarrolladores emplean mucho esfuerzo en salvar y cargar objetos entre la memoria y la base de datos lo que hace que el programa sea complicado y difícil de mantener. Visual Paradigm simplifica estas tareas mediante la generación de una capa de persistencia entre objeto y modelos de datos.

- Generador de mapeo objeto-relacional sofisticado.

La capa de mapeo objeto-relacional que se genera incorpora características como soporte de transacciones, capa de conectar en caché, agrupación de conexiones y personalización de sentencias SQL.

- Amplia cobertura para bases de datos.

Soporta una amplia gama de base de datos, incluidos Oracle, DB2, Cloudscape / Derby, Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Microsoft SQL Server, PostgreSQL, MySQL y otros.

- Base de datos de ingeniería inversa.

Permite la ingeniería inversa existente en una base de datos a través JDBC en el modelo entidad-relación. Los desarrolladores pueden transformar el modelo entidad-relación al modelo de objetos y rediseñar la base de datos para un mayor desarrollo.

Capítulo 1: Fundamentación teórica de la investigación.

- Integración con IDE.

VP-UML no sólo es una aplicación independiente, se puede integrar a los principales Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés): Eclipse, WebSphere y Borland (28).

Entorno de Desarrollo Integrado (IDE) Eclipse 3.5.2: Eclipse es un IDE para todo tipo de aplicaciones de código abierto, independiente de una plataforma para desarrollar lo que el proyecto llama Aplicaciones de Cliente Enriquecido, opuesto a las aplicaciones Cliente-liviano basadas en navegadores. La característica clave de Eclipse es la extensibilidad. Es una gran estructura formada por un núcleo y muchos plugins que van conformando la funcionalidad final. La forma en que los plugins interactúan es mediante interfaces o puntos de extensión; así, los nuevos aportes se integran sin dificultad ni conflictos. Es una herramienta que necesita de mucha ayuda del hardware para realizar la compilación del código fuente escrito, además de ser multiplataforma (29).

PgAdmin III 1.10.5: Es una herramienta para la administración de la base de datos, de código abierto que tiene una interfaz gráfica que soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux (30).

Subversion 1.6.12: es un sistema de control de versiones diseñado específicamente para reemplazar al popular Concurrent Versions System (CVS), el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo (31).

Ventajas de Subversion.

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Puede ser servido mediante Apache. Esto permite que los clientes utilicen Subversion en forma transparente.
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.

Capítulo 1: Fundamentación teórica de la investigación.

- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar.

En este capítulo se profundizó en los conceptos relacionados con el familiograma y SENDN lo que permitió conocer más a fondo las actividades que se realizan en los mismos. Los sistemas analizados en el estado del arte no ofrecen una solución viable al problema planteado. El estudio de las tecnologías y herramientas propuestas por el departamento de gestión hospitalaria permitió asimilar y aplicar las mismas, pues estas ofrecen ventajas para la implementación de las funcionalidades debido a que se integran fácilmente entre sí.

Capítulo 2. Características del sistema propuesto.

En este capítulo se explican los conceptos fundamentales abordados en el desarrollo del Modelo de Negocio facilitando la comprensión de estos. Se describen las características del sistema a través de los requisitos no funcionales y se especifican los requisitos funcionales que deben ser automatizados. Estas deben ser lo suficientemente buenas como para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre qué debe hacer y qué no debe hacer el sistema.

2.1. Modelo del negocio.

El modelo de negocio es una representación simplificada de la lógica de negocio. Permite realizar una descripción detallada de los requisitos que se incluirán en el sistema. Como finalidad persigue describir cada proceso del negocio, especificando sus datos, actividades, roles y las reglas del mismo. Mediante este modelo los clientes y los desarrolladores dejan definidos lo relacionado con la estructura y la dinámica de cómo se va a desarrollar el sistema (32).

2.1.1. Identificación de roles del entorno del negocio.

Una vez identificados los procesos del negocio, se pasa a definir quienes se van a encargar de su realización. Donde cada uno, desempeña un rol específico. A continuación se verán cuáles son estos roles.

Actor del negocio: Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se tiene cuando se interactúa con el negocio para beneficiarse de sus resultados. El actor identificado en el desarrollo de este trabajo fue el paciente que asiste a la consulta de genética en el hospital “William Soler”, (ver tabla 1).

Tabla 1: Actores del negocio.

Actores del negocio	Descripción
Paciente	Niños de cero a cinco años de edad atendidos en la consulta de Genética del PRC del Hospital “William Soler”.

Capítulo 2: Características del sistema propuesto.

Trabajador del negocio: representa un rol que juega una persona (o grupo de personas), una máquina o un sistema automatizado; actuando en el negocio, (ver tabla 2).

Tabla 2: Trabajadores del negocio.

Trabajador del negocio	Descripción
Genetista	Es el especialista que recoge los datos del paciente y realiza el familiograma.

2.1.2. Diagrama de procesos del negocio.

Proceso: Se puede definir como una serie de actividades que se relacionan entre sí, que tiene como finalidad dar lugar a una variedad de salidas, materiales o información con valor añadido, a partir de diferentes entradas de materiales o información, (ver figura 9).

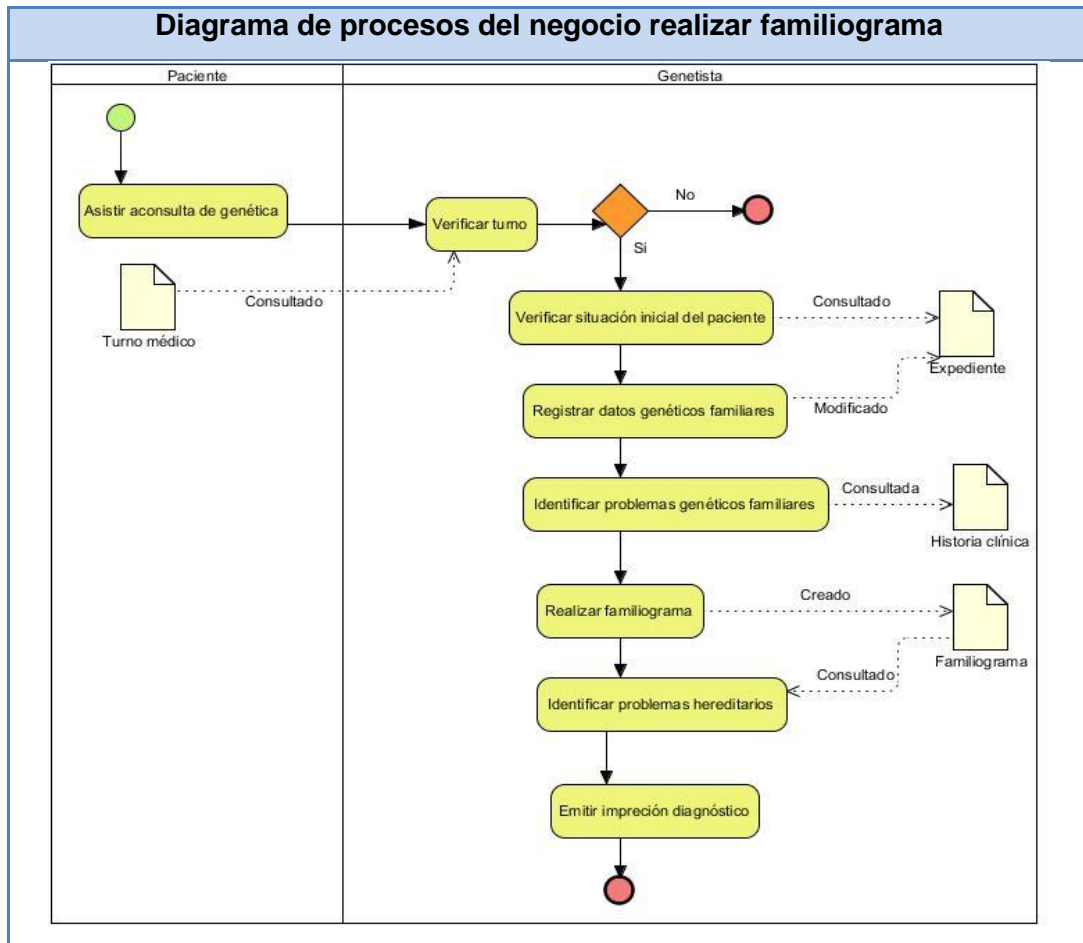


Figura 9: Diagrama de procesos del negocio.

2.1.3. Descripción del proceso de negocio.

Para un mayor entendimiento del proceso de negocio se hace una descripción más detallada, de cada uno de dichos procesos, (ver tabla 3).

Tabla 3: Descripción de los procesos del negocio

Objetivo	Realizar familiograma del paciente en la especialidad de genética.
Evento que lo genera	Cuando el paciente es atendido en la consulta de genética.
Precondiciones	El paciente no tiene realizado el familiograma.
Reglas del negocio	El paciente tiene una cita o turno previo, se le revisa si coincide con la fecha del día, de lo contrario no se atiende el paciente.
Cliente interno	Genetista

Capítulo 2: Características del sistema propuesto.

Cliente externo	Paciente
Entrada	Turno.
Flujo de eventos	
Flujo básico de Realizar familiograma del paciente	
Verificar turno: El genetista comprueba que la fecha del turno que tiene el paciente es la misma que la fecha del día que se está realizando la consulta.	
Verificar la situación inicial del paciente: El genetista consulta el expediente del paciente para verificar si se le ha realizado un familiograma.	
Registrar datos genéticos familiares: El genetista registra los datos genéticos de los familiares del paciente.	
Identificar problemas genéticos familiares: El genetista identifica enfermedades y síntomas que comparten otros miembros de la familia.	
Realizar familiograma: Una vez recopilada toda la información necesaria el genetista procede a conformar el familiograma.	
Identificar problemas hereditarios: Con el familiograma ya conformado el genetista reconoce cuáles son los problemas genéticos que pueden afectar al niño.	
Emitir impresión diagnóstico: El genetista realiza un diagnóstico a partir de lo observado en el paciente.	
Poscondiciones	
Se ha realizado la evaluación del paciente.	
Se crea un nuevo Familiograma.	
Salidas	
Se realizó el familiograma.	
Seguir con la consulta de genética.	
Flujos alternos	
Verificar turno: si la fecha del turno médico no coincide con la fecha de la consulta, el paciente no es atendido y el proceso termina.	

2.2. Especificación de requisitos del sistema.

Para poder obtener éxitos y resultados relevantes en la realización de un sistema informático es muy importante que se establezca una comunicación entre los clientes o usuarios que van a utilizar el mismo y el equipo de desarrollo. Es importante que el cliente mencione las funcionalidades que desea que tenga el producto pero también que los desarrolladores definan

otras que quizás el cliente no conoce que se pueden incluir. Para complementar todo esto es necesario que se realice la identificación de los requisitos que se llevarán a cabo en el proceso de desarrollo del sistema.

2.2.1. Requisitos funcionales.

Los requisitos funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera que este debe reaccionar a entradas particulares y cómo se debe comportar en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (33). Para el levantamiento de los mismos en el sistema, fue fundamental la entrevista con los genetistas del hospital “William Soler”, aportando la base fundamental para el desarrollo de la aplicación.

RF 1 – Listar pacientes sin familiograma.

RF 2 – Buscar pacientes sin familiogramas realizados.

RF 3 – Crear familiograma.

RF 4 – Insertar datos de los abuelos.

RF 5 – Listar abuelos.

RF 6 – Ver datos de abuelos.

RF 7 – Modificar datos de los abuelos.

RF 8 – Eliminar datos de abuelos.

RF 9 – Insertar datos de los tíos.

RF 10 – Listar tíos.

RF 11 – Ver datos de tíos.

RF 12 – Modificar datos de los tíos.

RF 13 – Eliminar datos de tíos.

Capítulo 2: Características del sistema propuesto.

RF 14 – Insertar datos de los padres.

RF 15 – Listar padres.

RF 16 – Ver datos de padres.

RF 17 – Modificar datos de los padres.

RF 18 – Eliminar datos de padres.

RF 19 – Insertar datos de los primos.

RF 20 – Listar primos.

RF 21 – Ver datos de primos.

RF 22 – Modificar datos de los primos.

RF 23 – Eliminar datos de primos.

RF 24 – Insertar datos de los hermanos.

RF 25 – Listar hermanos.

RF 26 – Ver datos de hermanos.

RF 27 – Modificar datos de los hermanos.

RF 28 – Eliminar datos de los hermanos.

RF 29 – Dibujar familiograma.

2.2.2. Requisitos no funcionales.

Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de

Capítulo 2: Características del sistema propuesto.

comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Son importantes para que los clientes y usuarios puedan valorar las características no funcionales del producto, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

RNF Usabilidad

La usabilidad se refiere a la capacidad de un sistema informático de ser comprendido, aprendido, usado y atractivo para el usuario, en condiciones específicas de uso. Estos requerimientos describen los niveles apropiados de usabilidad, dados los usuarios finales del producto. Los requerimientos de usabilidad se derivan de la combinación de lo que el cliente está tratando de lograr con el producto y lo que los usuarios finales esperan del mismo.

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

- Para alcanzar un nivel elemental asociado al dominio del sistema y el uso eficiente del mismo, serán necesarios 20 días de preparación, obteniendo la categoría de usuarios normales.
- Para alcanzar un nivel avanzado asociado al dominio del sistema y el uso eficiente del mismo, serán necesarios 30 días de preparación, obteniendo la categoría de usuarios avanzados.

RNF Fiabilidad

La fiabilidad es la característica de los sistemas informáticos por la que se mide el tiempo de funcionamiento sin fallos. En la investigación se propone garantizar la misma de diversas formas:

- En los servidores de los hospitales se deberá proporcionar una arquitectura de máxima disponibilidad, tanto de servidores de aplicación como de base de datos. Se plantea el empleo de políticas de respaldo a toda la información, para evitar pérdidas en caso de desastres ajenos al sistema.
- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la base de datos, independientemente de su inexistencia para el sistema. Permitirá la recuperación de la información de la base de datos, a partir de los respaldos, o salvadas realizadas.

Capítulo 2: Características del sistema propuesto.

RNF Eficiencia

Imponen condiciones a los RF, por ejemplo, para una acción específica pueden definirse parámetros tales como: velocidad de procesamiento o cálculo, rendimiento, disponibilidad, precisión, tiempo de respuesta, tiempo de recuperación y aprovechamiento de los recursos.

- El Centro de Datos permitirá agregar recursos para aumentar el poder de procesamiento y almacenamiento sin afectar los sistemas, garantizando expansiones motivadas por futuros requerimientos.
- El sistema minimizará el volumen de datos en las peticiones y además, optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla, guardar en la memoria caché datos y recursos de alta demanda.
- Se respetarán las buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos. Se deberá usar siempre que sea posible, el patrón Singleton (instancia única), destruir referencias que ya no estén siendo usadas, y optimizar el trabajo con cadenas, entre otras buenas prácticas, que ayudan a mejorar el rendimiento.

RNF Soporte

Abarcan todas las acciones que se llevarán a cabo una vez que se ha terminado el desarrollo del sistema, con motivos de asistir a los clientes de este, logrando así su mejora progresiva y evolución en el tiempo. Pueden incluir: pruebas, extensibilidad, adaptabilidad, mantenimiento, configuración, compatibilidad, servicios e instalación.

- Las notificaciones de las deficiencias detectadas en la aplicación desplegada deberán realizarse por escrito.
- Una vez notificada por la entidad médica, la deficiencia detectada en la aplicación desplegada, el equipo de desarrollo deberá solucionarla en el menor tiempo posible de acuerdo a la complejidad de la misma.
- La capacitación y entrenamiento del profesional de salud para el uso del sistema, se realizará en un período de 6 meses.

RNF Réplica

Capítulo 2: Características del sistema propuesto.

Se permitirá realizar réplica de la base de datos de los hospitales con el Centro de Datos. Esta réplica se podrá hacer de forma manual y automatizada, a través de la red.

RNF Restricciones de diseño

El sistema estará dividido en las siguientes capas.

Capas físicas

Cliente: computadora con cualquier tecnología o sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se recomienda Internet Explorer 7 o superior o Firefox 3.0 o superior).

Servidor de Aplicaciones: servidor con cualquier tecnología o sistema operativo que soporte Java Runtime Environment (JRE) 6 y al JBoss AS 4.2.

Servidor de Base de Datos: servidor con cualquier tecnología o sistema operativo que soporte a PostgreSQL Server 8.4 en los servidores de base de datos de cada hospital.

Capas lógicas

Presentación: contiene todas las vistas y la lógica de la presentación. El flujo web se maneja de forma declarativa y basándose en definiciones de procesos del negocio.

Negocio: mantiene el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario. En los casos de que algún objeto del negocio tenga una interfaz externa, siendo accesible la misma desde sistemas legados o directamente del cliente, se garantiza la seguridad a nivel de objeto y métodos.

Acceso a Datos: contiene las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

Interfaz

Interfaces de usuario

Capítulo 2: Características del sistema propuesto.

- Las ventanas del sistema contendrán bien estructurados los datos, además de permitir la interpretación correcta de la información.
- La entrada de datos incorrecta será detectada claramente, e informada al usuario.
- Todos los textos y mensajes en pantalla aparecerán en idioma español.
- El diseño de la interfaz del sistema responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.

RNF Rendimiento

La eficiencia del sistema estará dada por el aprovechamiento de los recursos que se disponen en el modelo usuario. El sistema debe ser rápido y el tiempo de respuesta debe ser el mínimo posible.

- Se minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria.
- Respetará buenas prácticas de programación, para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

RNF Seguridad

Este es quizás el tipo de requisito más difícil, que provocará los mayores riesgos si no se maneja correctamente, debido a la cantidad de personas dedicadas a encontrar fallas de seguridad en las aplicaciones y a explotarlas con fines malintencionados. Por ello, se hace necesario establecer una serie de políticas de seguridad tendentes a garantizar la continuidad del sistema en el caso que se produzcan incidencias, fallos, actuaciones malignas por parte de terceros, pérdidas accidentales o desastres que afecten los datos almacenados.

- La información estará protegida contra accesos no autorizados utilizando mecanismos de validación que puedan garantizar el cumplimiento de: usuario, contraseña y nivel de acceso, de manera que, cada uno pueda tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios, garantizando así la confidencialidad.
- Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando solo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.

Capítulo 2: Características del sistema propuesto.

- El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.
- Se usarán mecanismos como Md5¹ de encriptación para el envío de las contraseñas al servidor de base de datos.
- Las contraseñas podrán cambiarse únicamente por el propio usuario, o por el administrador del sistema.

RNF Hardware

Los requisitos de hardware estarán en correspondencia con la plataforma específica que se utilice para la instalación del sistema, en cuanto a sistema operativo, servidor de aplicaciones y gestor de bases de datos.

Estaciones de trabajo: las estaciones de trabajo deben tener como mínimo 256 Mb de memoria RAM y un microprocesador a 2.0 Hz.

Servidores: Los servidores de aplicación deberán tener como mínimo un procesador Intel(R) Xeon(R) 5140 Dual-Core, 4GB de memoria RAM y 1 TB de disco duro mientras que los servidores de base de datos deberán tener como mínimo un procesador Intel(R) Xeon(R) 5140 Dual-Core, 8GB de memoria RAM y 1 TB de disco duro.

2.3. Modelo de casos de uso del sistema.

El modelo de casos de uso contiene las relaciones existentes entre los diferentes tipos de usuarios que interactúan con el sistema con el objetivo de darle solución a un determinado problema. Tiene la función de ser un elemento común para los desarrolladores en pos de unificar cada uno de sus trabajos. Se usa en diferentes etapas del desarrollo del sistema, en la especificación de los requisitos funcionales, para el diseño y la documentación del usuario.

¹**MD5** es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (*Massachusetts Institute of Technology*, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4 después de que Hans Dobbertin descubriese su debilidad.

Capítulo 2: Características del sistema propuesto.

Actor del sistema: Se le llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo. El conjunto de casos de uso al que un actor tiene acceso define su rol global en el sistema y el alcance de su acción, (ver tabla 4) (34).

Tabla 4: Actores del sistema.

Actor	Descripción
Genetista	Es el especialista que realiza la valoración inicial del paciente y realiza el famiogramas

2.3.1. Diagrama de casos de uso del sistema.

Un diagrama de Casos de Uso del Sistema refleja la relación existente entre los actores y los casos de uso o entre los mismos casos de usos. Se representan las funcionalidades del sistema de una manera gráfica. Permiten modelar el comportamiento del sistema, un subsistema o una clase, (ver figura 10).

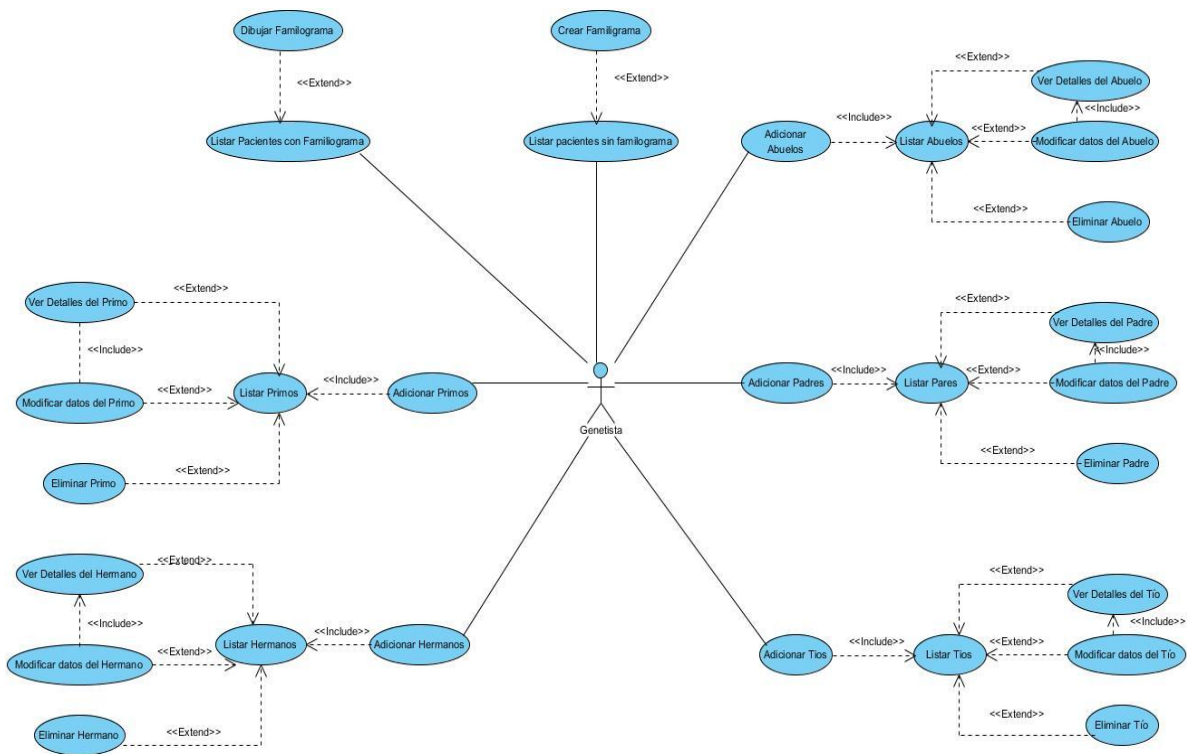


Figura 10: Diagrama de casos de uso del sistema.

2.3.2. Descripción de casos de uso del sistema.

Un caso de uso (CU) describe lo que el sistema debe hacer desde el punto de vista del usuario (35). A continuación se muestra el caso de uso Adicionar Abuelos, (ver tabla 5). Los demás casos de uso se pueden consultar en el [Anexo 1](#).

Tabla 5: Casos de uso del sistema.

CU – 4	Adicionar Abuelos.
Actor	Genetista
Descripción	El caso de uso inicia cuando el actor accede a la opción insertar datos de abuelos, el sistema brinda la posibilidad de introducir los datos correspondientes a los abuelos del paciente, el actor introduce los datos, el sistema inserta los datos, el caso de uso termina.
Referencia	RF 4

2.3.3. Especificación de casos de usos.

A continuación se muestra la especificación del caso de uso Adicionar Abuelos, (ver tabla 6). Las demás especificaciones de casos de uso se pueden consultar en el [Anexo 2](#).

Adicionar Abuelos

Tabla 6: Especificación de casos de uso del sistema.

Caso de uso	
CU-4	Adicionar Abuelos.
Propósito	Insertar la información correspondiente a los abuelos.
Actores	Genetista
Resumen: El caso de uso inicia cuando el actor accede a la opción Insertar datos de abuelos, el sistema brinda la posibilidad de introducir la información correspondiente a los abuelos del paciente, el sistema inserta los datos, el caso de uso termina.	
Precondiciones	Debe haberse creado el familiograma asociado al paciente a atender en el día, para posteriormente insertar la información de los abuelos.

Capítulo 2: Características del sistema propuesto.

Referencias	RF 4
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor accede a la opción insertar datos de abuelos.	
	<p>2. Brinda la posibilidad de introducir los datos de los abuelos.</p> <ul style="list-style-type: none"> • Aceptar Insertar datos del abuelo. • Cancelar operación. Ver Alternativa 1: “Cancelar operación.”
3. Introduce la información de los abuelos y selecciona la opción Insertar.	
	4. Valida los datos. Si hay datos incompletos, ver Alternativa 2: “Existen datos incompletos.”
	5. Inserta los datos.
	6. Muestra el mensaje de información “Se han insertado los datos de los abuelos.”
	7. El caso de uso termina.
Flujo alternativo	
Alternativa 1. “Cancelar operación.”	
Acción del actor	Respuesta del sistema
1. Selecciona la opción de Cancelar operación.	
	2. Regresa a la vista anterior.
Alternativa 2. “Existen datos incompletos.”	

Capítulo 2: Características del sistema propuesto.

	1. Muestra el mensaje de error “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
	2. Muestra un indicador sobre los campos vacíos.

La realización del presente capítulo permitió obtener una descripción del negocio mediante el modelado del mismo, relacionado a la consulta de genética, especialmente cuando se adiciona un abuelo. Fueron definidos cada uno de los requisitos a cumplir por parte del sistema ya sean funcionales como no funcionales tomando como referencia las necesidades del cliente, con el objetivo primordial de que el sistema tenga éxito cuando esté terminado. A partir de los requisitos funcionales se especificaron los actores y casos de usos del sistema, así como las funcionalidades a implementar.

Capítulo 3 Diseño del sistema propuesto.

En este capítulo se presentan los aspectos fundamentales del diseño de las funcionalidades para la gestión de los familiogramas. Se analizará la arquitectura utilizada para el desarrollo de las mismas y también se muestran los diagramas de clases del diseño realizados.

3.1. Patrones de arquitectura y diseño.

Los patrones de diseño especifican la estructura y el comportamiento de una sociedad de clases. Estos son directrices y principios estructurados que describen un problema común y presentan soluciones simples. Están basados en la experiencia y se ha demostrado que funcionan. Los Patrones de Diseño empleados se concretaron durante la elaboración del diseño de las clases y las relaciones entre ellas, con el propósito de garantizar un bajo acoplamiento y alta cohesión entre las clases, así como un correcto encapsulamiento de las clases y la adecuada distribución de responsabilidades entre ellas (36).

Patrones arquitectónicos: especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes del sistema. Para la realización del familiograma para el módulo de genética del SENDN se propone el uso de los patrones arquitectónicos Modelo Vista Controlador (MVC) y en capas, debido que es un estilo de arquitectura de software que separa los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, o sea que dicho patrón se ve frecuentemente en aplicaciones Web, donde la vista es la página XHTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la lógica de negocio, permitiendo mayor independencia, mantenimiento y reutilización.

Modelo: representa los datos del programa y controla todas sus transformaciones.

Vista: genera la presentación visual de los datos representados por el Modelo y muestra los datos al usuario.

Controlador: maneja las entradas del usuario, actuando sobre los datos representados por el Modelo.

Este modelo de arquitectura presenta varias ventajas entre las cuales se destacan:

- Clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.

- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

3.1.1. GRASP

GRASP: (General Responsibility Assignment Software Patterns) se encargan de asignar responsabilidades, las principales responsabilidades son conocer atributos y relaciones con otros objetos y hacer tareas que debe cumplir cada objeto (37).

Los principales patrones GRASP utilizados fueron:

Experto

Indica cual es la clase que debe asumir una determinada responsabilidad (clase experta), teniendo en cuenta la información que contiene para cumplirla. Este se evidencia en la definición de las clases según las funcionalidades que realizan, como por ejemplo la clase CCaddAbuelos que es la encargada de realizar las funcionalidades de insertar, listar, actualizar y eliminar abuelos, entre otras.

Creador

El patrón Creador recomienda que una clase B debe crear una instancia de A si:

- B agrega los objetos de A.
- B contiene a los objetos de A.
- B registra las instancias de los objetos de A.
- B utiliza específicamente los objetos de A.
- B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado.

Este patrón se evidencia en la clase CCFamiliograma, la cual es la responsable de crear instancias de la clase Fam_General cada vez que desee utilizar funcionalidades de esta.

Bajo Acoplamiento

Este patrón resuelve el problema de cómo lograr baja dependencia, alta reutilización entre las clases y minimizar el impacto de los cambios. La figura 11 ejemplifica la forma de empleo del mismo.



Figura 11: Patrón bajo acoplamiento ejemplificado.

En este caso la responsabilidad de crear objetos de Fam_General la tiene la clase CcaddAbuelos disminuyendo el número de recurrencias de la clase CCFamiliograma a Fam_General.

Alta Cohesión

Este patrón se aplica para agrupar las clases en dependencia de los requerimientos y los procesos que se necesitan informatizar, de modo que cada clase implementa las operaciones que se encuentran en su misma área funcional, de esta forma CcaddAbuelos solamente se encarga de crear abuelos, así como otras operaciones que se realizan sobre esta área, limitándola a otras operaciones de otra naturaleza.

Controlador

Este patrón define quién responde ante determinados eventos. En este caso la clase controladora de cada componente, aparte de separar la lógica de negocios de la capa de presentación, delega en otras clases (los modelos de negocio y dominio) la responsabilidad de efectuar operaciones complejas sobre el área, funcionando como agente de tráfico para las operaciones, donde recibe peticiones y devuelve respuestas, mientras que otras clases se encargan de gestionarlas realmente. CCFamiliograma es ejemplo de cómo se aplica este patrón.

3.2. Modelo de diseño.

El diseño tiene como objetivo crear modelos centrados en los requisitos no funcionales y en el dominio de la solución. Se presenta como antesala a la implementación del sistema, mediante el modelo de diseño se puede verificar el cumplimiento de los requisitos del sistema sin tener un prototipo funcional de este. Es usado como una entrada inicial en las actividades de implementación y prueba. Es un modelo o guía para realizar la implementación del sistema que se va a desarrollar.

Capítulo 3: Diseño del sistema propuesto.

Para el modelado del sistema se utilizaron los estereotipos definidos para este tipo de aplicaciones por la extensión de UML, lo que proporciona un mayor entendimiento de diseño de la aplicación. A continuación se brinda una breve explicación de cada uno de ellos.

<<Client Page>>: Es una página web que se ejecuta del lado del cliente sobre un navegador. Su función fundamental es permitir la interacción entre el sistema y el usuario. Solo pueden ser creadas por una única Server Page y pueden presentar algún vínculo con otras páginas clientes.

<<HTML Form>>: Es una colección de elementos de entrada que están contenidos en la página cliente. Su función fundamental es enviar los datos mediante submit para las páginas servidoras para que sean procesados.

<<Server Page>>: Es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Además se encarga de construir las páginas HTML

3.2.1. Diagrama de clases del diseño.

Los diagramas de clases del diseño se encargan de describir gráficamente un conjunto de interfaces, colaboraciones y sus relaciones para satisfacer los detalles de la implementación. Modelan la vista de diseño estática de un sistema. Contienen información como clases, asociaciones y atributos, interfaces con sus operaciones y constantes, métodos, información sobre los tipos de los atributos, navegabilidad y dependencias. Cada caso de uso cuenta con un diagrama de clases del diseño y pueden organizarse en paquetes o subsistemas para hacer más fácil su comprensión, (ver figura 12).

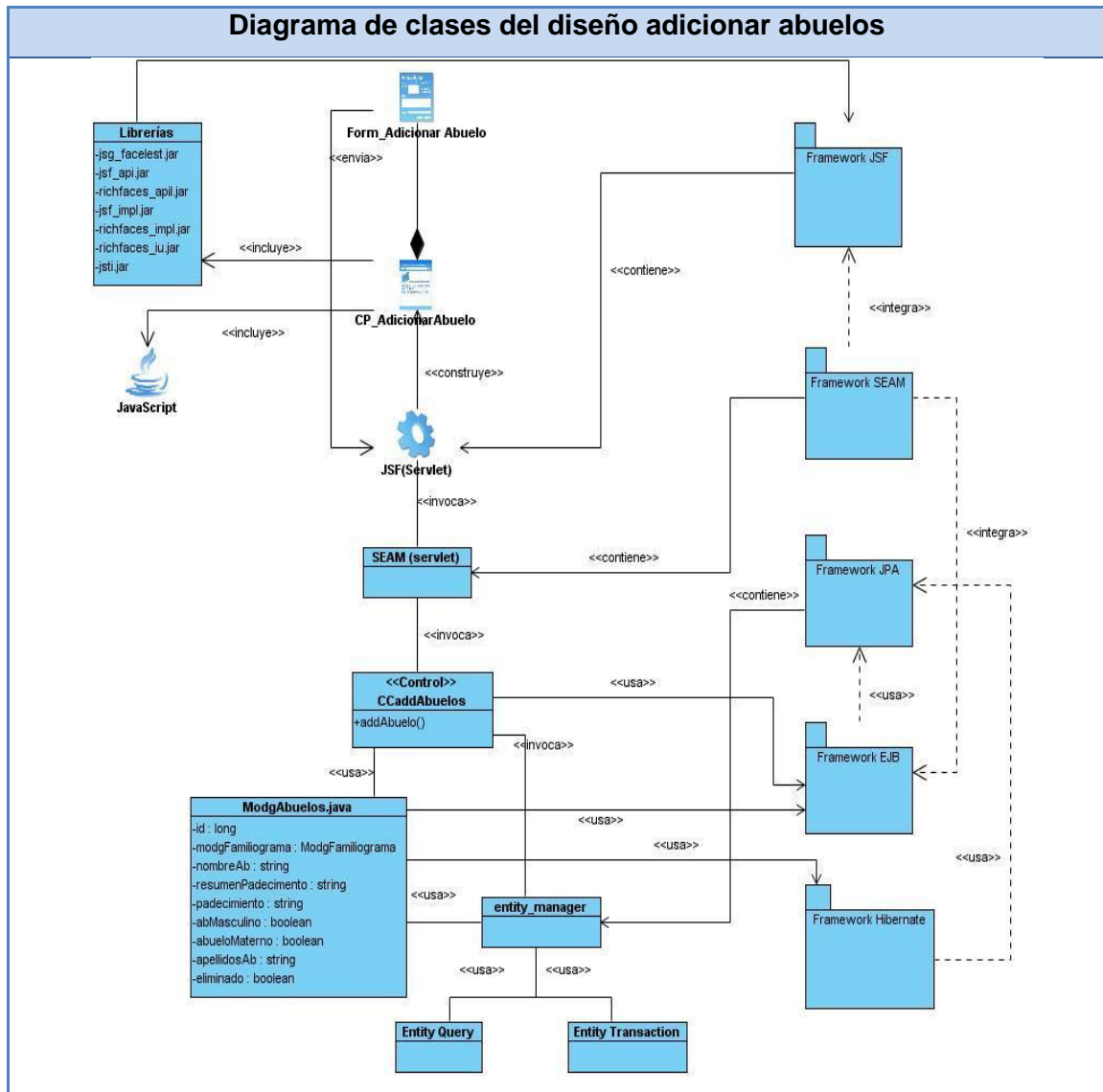





Figura 12: Diagrama de clases del diseño adicionar abuelos.

3.2.2. Clases comunes. Descripción.



Mediante la descripción de las clases del diseño se puede conocer cómo están conformadas estas clases con sus atributos, (ver tabla 7). A continuación se muestra la descripción de estas clases las demás se pueden consultar en el [Anexo 3](#).

Capítulo 3: Diseño del sistema propuesto.

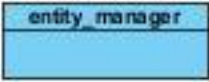
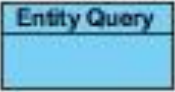
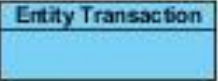
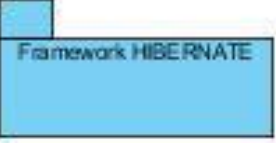
Tabla 7: Descripción de clases comunes.

Capa de Presentación	
Nombre:	Propósito:
 <p>Figura 12.1 Clase JavaScript</p>	<p>Crear interfaces de usuario mejoradas y páginas web dinámicas mediante componentes reutilizables.</p>
Descripción:	
<p>Permite validar la información que se maneja por parte del cliente y también la construcción de páginas de una manera más dinámica integradas a un navegador web.</p>	
Nombre:	Propósito:
 <p>Figura 12.2 Clase Librerías</p>	<p>Conjunto de librerías para la creación de interfaces.</p>
Descripción:	
<p>Contiene un conjunto de librerías que permiten construir interfaces de usuario mediante componentes reutilizables y extensibles. Disminuyen en gran manera las operaciones de construir y mantener aplicaciones web con componentes de interfaz del lado del servidor.</p>	
Nombre	Propósito:
 <p>Figura 12.3 Clase JSF Servlet</p>	<p>Permite la obtención de las peticiones provenientes de las páginas JSF.</p>

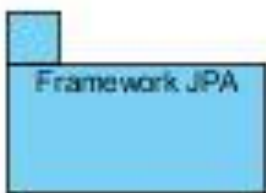
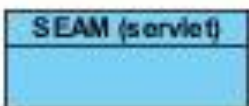
Capítulo 3: Diseño del sistema propuesto.

Descripción:	
<p>Consiste en el controlador para JSF el cual es el encargado de interceptar las peticiones de las páginas clientes, asociándoles a estas páginas, clases java que recopilan la información introducida y a su vez cuentan con un grupo de métodos que dan respuestas a las acciones del usuario.</p>	
Nombre	Propósito
 <p>Figura 12.4 Paquete Marco de trabajo JSF</p>	<p>Simplificar el desarrollo de interfaces de usuario en aplicaciones java basadas en el patrón MVC.</p>
Descripción:	
<p>Consiste en un marco de trabajo de los componentes de la interfaz de usuario y es válido para todas aquellas aplicaciones web basadas en la tecnología JAVA, está basado en el patrón MVC. Forma parte de la especificación JEE 5.</p>	
Capa de Negocio	
Nombre:	Propósito
 <p>Figura 12.5 Paquete Marco de trabajo EJB</p>	<p>Encapsular la lógica de negocio que da cumplimiento el propósito de la aplicación.</p>
	Observaciones:
	<p>EJB se incluye en las capas de Negocio y Persistencia.</p>
Descripción:	
<p>Enterprise JavaBeans consiste en una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables mediante el lenguaje de programación Java. Persigue como objetivo simplificar el desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java.</p>	
Capa de Datos	
Nombre:	Propósito:

Capítulo 3: Diseño del sistema propuesto.

 <p>Figura 12.6 Clase entityManager</p>	<p>Gestionar las entidades proveyendo servicios de persistencia.</p>
<p>Descripción</p>	
<p>Se encarga de que se realicen las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en las cuales intervienen entidades.</p>	
<p>Nombre:</p>	<p>Propósito:</p>
 <p>Figura 12.7 Clase EntityQuery</p>	<p>Agregar consultas que pueden ser aplicadas a las entidades del modelo.</p>
<p>Descripción</p>	
<p>Permite encontrar objetos persistentes manejando cierto criterio de búsqueda. También permite realizar peticiones a la base de datos y controla cómo se ejecuta dicha petición. Se utiliza para enlazar los parámetros de la petición, limitar el número de resultados devueltos por la petición y para ejecutar dicha petición.</p>	
<p>Nombre:</p>	<p>Propósito</p>
 <p>Figura 12.8 Clase EntityTrasaction</p>	<p>Agrupar las operaciones sobre datos persistentes en una unidad transaccional.</p>
<p>Descripción</p>	
<p>Permite realizar operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.</p>	
<p>Nombre:</p>	<p>Propósito</p>
 <p>Figura 12.9 Paquete Framework HIBERNATE</p>	<p>Herramienta de mapeo objeto/relacional con la base de datos.</p>

Capítulo 3: Diseño del sistema propuesto.




Descripción	
<p>Consiste en varias clases agrupadas en componentes que constituyen una herramienta de mapeo objeto/relacional u ORM de código abierto y además brinda la posibilidad de generar sentencias SQL. Permite el diseño de objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. Permite generar Base de Datos en cualquiera de los entornos soportados: Oracle, PostgreSQL, DB2 y MySQL, de una manera muy rápida y optimizada.</p>	
Nombre	Propósito
 <p>Figura 12.10 Paquete Framework JPA</p>	<p>Unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.</p>
Descripción	
<p>Consiste en un conjunto de clase agrupadas en componentes que constituyen la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB 3.0 como parte de JSR 220, su uso no se limita a los componentes de software EJB. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos. Permite unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.</p>	
Nombre	Propósito
 <p>Figura 12.11 Clase SEAM Servlet</p>	<p>Permite la interacción de la capa de Negocio con la de Presentación.</p>
	Observaciones:
	<p>Se encuentra ubicado entre la capa de Presentación y la de Negocio y no en una capa en específico.</p>
Descripción	
<p>Es el controlador de SEAM que capta las peticiones derivadas de la interacción del usuario después de interactuar con el Servlet de JSF. Persigue como objetivo enrutar las peticiones hacia los Beans que posibilitarán darle respuesta a la petición solicitada. Interviene en la integración de las capas de presentación y negocio.</p>	

3.2.3. Descripción de caso de uso adicionar abuelos.

Capítulo 3: Diseño del sistema propuesto.

A continuación se describe el diagrama del caso de uso Adicionar Abuelos, (ver tabla 8). En dicha descripción se encuentran cada uno de los elementos que intervienen y el propósito que persiguen, también se incluye una breve descripción de estos elementos para que, de esta manera se comprenda mejor el funcionamiento del caso de uso.

Tabla 8: Descripción de caso de uso Adicionar Abuelos.

Capa de Presentación	
Nombre:	Propósito:
 CP_AdicionarAbuelo	Posibilita la interacción del usuario con la interfaz del cliente para adicionar los datos de un abuelo.
Descripción:	
Es una página web que se ejecuta del lado del cliente sobre un navegador. Incluye varias validaciones en JavaScript para evitar las peticiones innecesarias al servidor. Permite insertar y gestionar la información referente a los abuelos.	
Nombre:	Propósito:
 Form_AdicionarAbuelos	Enviar los datos introducidos en la vista CP_AdicionarAbuelo a la controladora CC_addAbuelos.
Descripción:	
La clase Form_AdicionarAbuelos contiene varios elementos de entrada que están contenidos en la página cliente para que el usuario introduzca los datos de los abuelos. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros).	
Nombre:	Propósito:
 CC_addAbuelos.java	Permite darle respuesta a las peticiones realizadas desde la vista CP_AdicionarAbuelo.
Descripción:	

La clase CC_addAbuelos.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones realizadas desde la vista CP_AdicionarAbuelo a través de los métodos que contiene. Se encarga de gestionar la acción de mostrar la información de los abuelos sobre la página cliente correspondiente. Hace uso del marco de trabajo EJB que encapsula la lógica de negocio, integrándose con la vista a través de SEAM.

Nombre:	Propósito:
<div style="border: 1px solid black; padding: 5px; background-color: #e0f0ff;"> <p style="margin: 0;">ModgAbuelos.java</p> <p style="margin: 0;">-id : long</p> <p style="margin: 0;">-modgFamiliograma : ModgFamiliograma</p> <p style="margin: 0;">-nombreAb : string</p> <p style="margin: 0;">-resumenPadecimiento : string</p> <p style="margin: 0;">-padecimiento : string</p> <p style="margin: 0;">-abMasculino : boolean</p> <p style="margin: 0;">-abueloMaterno : boolean</p> <p style="margin: 0;">-apellidosAb : string</p> <p style="margin: 0;">-eliminado : boolean</p> </div> <p>ModgAbuelos.java</p>	<p>Proveer el mapeo con la base de datos.</p>
Descripción:	
<p>La clase ModgAbuelos.java es una clase que se ejecuta del lado del servidor. Es utilizada para almacenar los datos de los abuelos del paciente. Está asociada a una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso de los marcos de trabajo Hibernate y JPA.</p>	

3.2.4. Descripción de las clases del diseño.

Mediante la descripción de las clases del diseño se puede conocer cómo están conformadas estas clases con sus atributos, (ver tabla 9). A continuación se muestra la descripción de la clase ModgAbuelos.

Tabla 9: Descripción de clase ModgAbuelos.

Nombre de la clase: ModgAbuelos	
Tipo de Clase: Modelo	
Atributo	Tipo
id	Long
modgFamiliograma	ModgFamiliograma
nombreAb	String
padecimiento	String
resumenPadecimiento	String
abMasculino	Boolean

Capítulo 3: Diseño del sistema propuesto.

abueloMaterno	Boolean
apellidosAb	String
eliminado	Boolean

3.3. Modelo de datos.

El Modelo de Datos describe la representación lógica y física de la información persistente manejada por el sistema, tipo de datos que incluye la base de datos y la forma en que se relacionan. Controla que la información básica almacenada para cada atributo o propiedad de un concepto sea válida. Está formado por objetos (entidades que existen y que se manipulan), atributos (características básicas de estos objetos) y relaciones (forma en que enlazan los distintos objetos entre sí). Puede ser inicialmente creado a través de la ingeniería inversa de un almacenamiento de datos persistentes que ya exista o puede ser inicialmente creado a partir de un conjunto de clases del diseño persistentes en el modelo de diseño. Es usado para definir el mapeo entre las clases del diseño y las estructuras de datos (38).

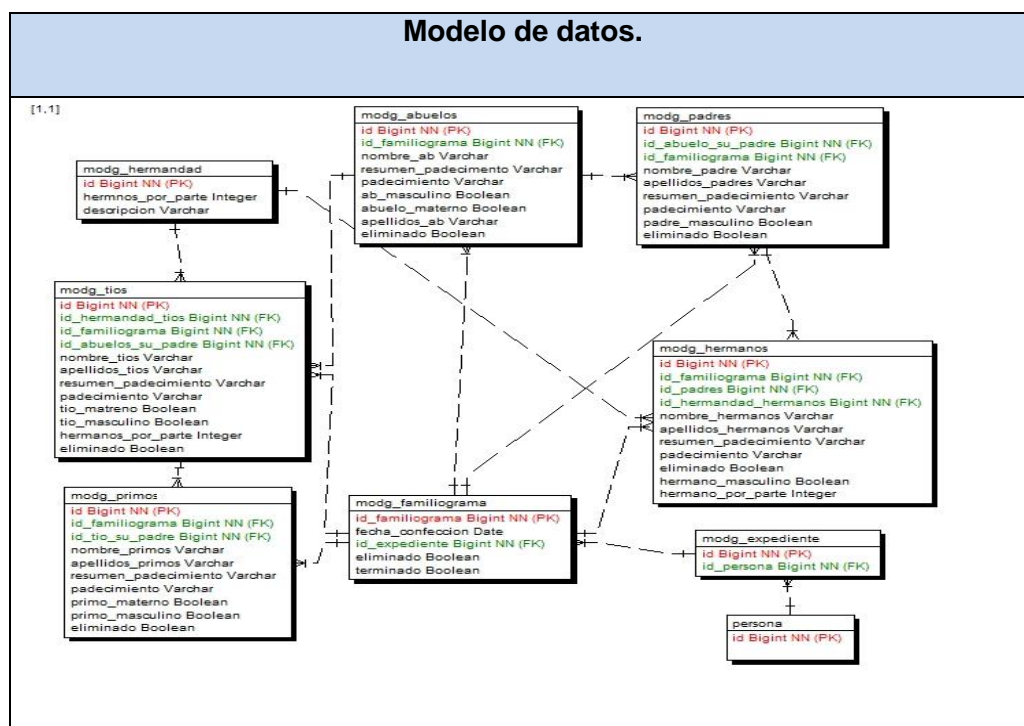


Figura 13: Modelo de datos.

3.4. Descripción de las tablas del modelo de datos.

A continuación se muestra la descripción de la tabla modg_abuelos, (ver tabla10), representada en el modelo de datos, (ver figura 13: Modelo de datos), las demás descripciones se pueden consultar en el [Anexo 4](#).

Capítulo 3: Diseño del sistema propuesto.

Tabla 10: Descripción de las tablas del modelo de datos.

Nombre de la clase: modg_abuelos		
Descripción: Tabla que almacena la información de los abuelos.		
Atributo	Tipo	Descripción
id	serial	En este atributo se almacena el identificador del abuelo.
id_familiograma	long	Identifica a que familiograma pertenece el abuelo. Este atributo es llave foránea que se absorbe de la tabla modg_familiograma.
nombre_ab	string	Guarda el nombre del abuelo.
apellidos_ab	string	Guarda los apellidos del abuelo.
ab_masculino	boolean	Identifica si el género del abuelo/a es masculino o femenino, si es verdadero es masculino.
padecimiento	string	Guarda el nombre del padecimiento del abuelo.
resumen_padecimiento	string	Guarda el resumen del padecimiento del abuelo.
abuelo_materno	boolean	Identifica si el abuelo es materno o paterno. Si es verdadero es porque es materno.
eliminado	boolean	Dice si el abuelo está eliminado o no.

Capítulo 3: Diseño del sistema propuesto.

El presente capítulo se definió de la arquitectura y el diseño de los requisitos funcionales señalados en el capítulo anterior. Se fundamentaron los patrones de diseño empleados, se definió la arquitectura basada en el patrón Modelo-Vista-Controlador. Fue especificada la estructura y comportamiento de las clases, donde se atienden los niveles de reutilización y mantenimiento, garantizando presentar una solución que cumpla con las necesidades del cliente.

Capítulo 4: Implementación del sistema propuesto.

En el presente capítulo se describe, de acuerdo a la arquitectura propuesta por el diseño, el modelado de los artefactos correspondientes al flujo de implementación. Se modelan los diagramas de componentes y de despliegue. Además se muestran los estándares de diseño y codificación, se explica cómo se realiza el tratamiento de errores y la seguridad del sistema. También se da una breve descripción de las interfaces principales del mismo

4.1. Modelo de Implementación.

El Modelo de Implementación consiste en obtener una visión general de lo que tiene que ser implementado. Describe los componentes a construir, su organización y la dependencia entre nodos físicos, en los que funcionará el sistema. Los diagramas de despliegue y de componente conforman lo que se conoce como modelo de implementación.

Para el modelo de implementación del familiograma del módulo de Genética del SENDN se expone una organización en capas, jerarquías de paquetes y subsistemas de implementación. Los mismos contienen componentes y sus relaciones, dividiendo el sistema en partes más manejables. Esto posibilita la reutilización, que se pueda implementar por separado y disminuye el impacto que pueda traer consigo un cambio, (ver figura 14).

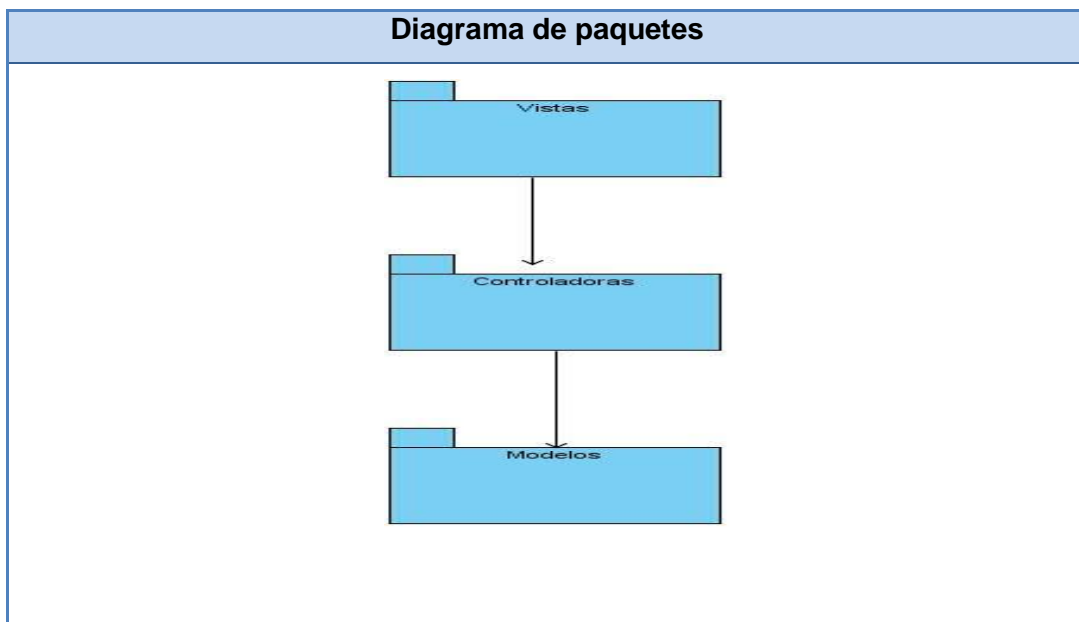


Figura 14: Diagrama de paquetes.

4.1.1. Diagrama de componentes.

El diagrama de componentes tiene como objetivo mostrar las organizaciones y dependencias lógicas entre componentes de sistemas informáticos, sean éstos componentes de código fuente, binarios o ejecutables. Permiten visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación.

A continuación se muestra el diagrama de componentes del familiograma para el módulo Genética del SENDN, conformado por el paquete Vistas, (ver figura 15), el paquete Controlador, (ver figura 16) y el paquete Modelo, (ver figura 17).

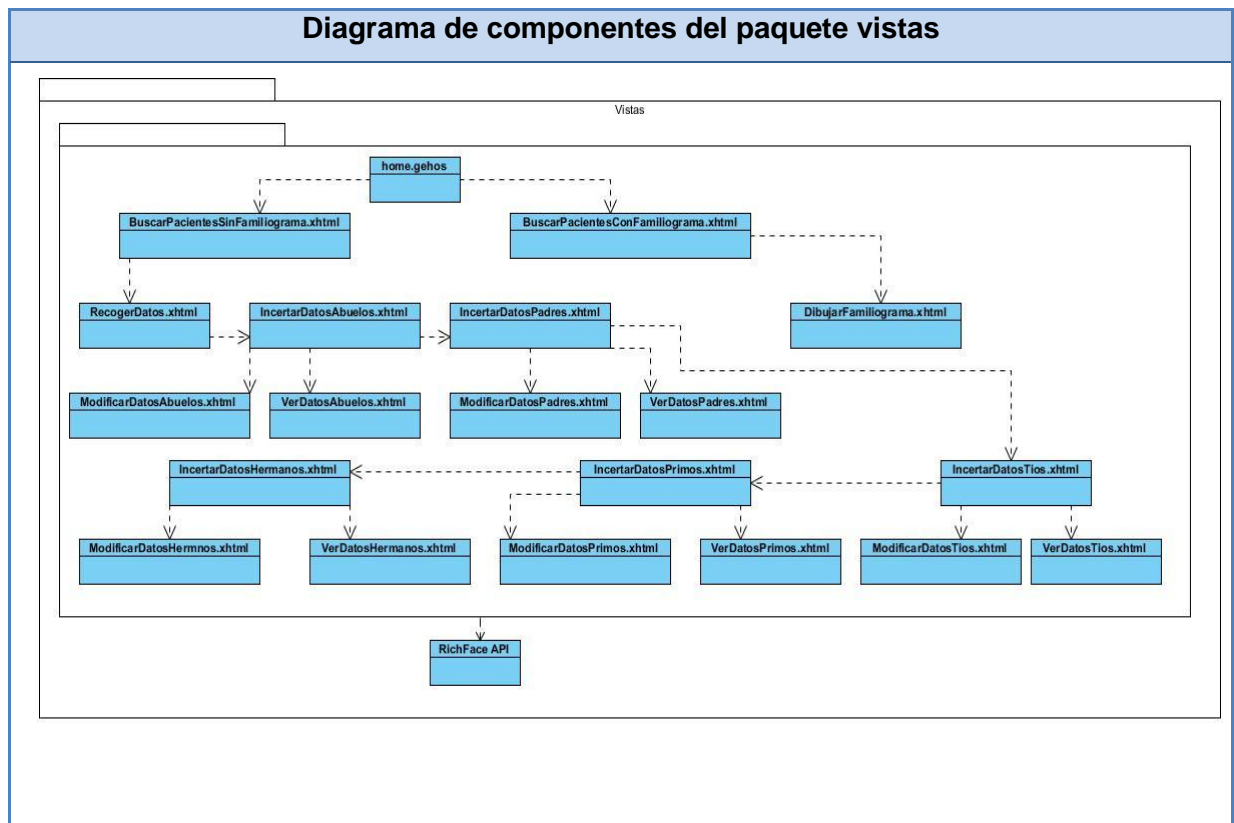


Figura 15: Diagrama de componentes del paquete vistas.

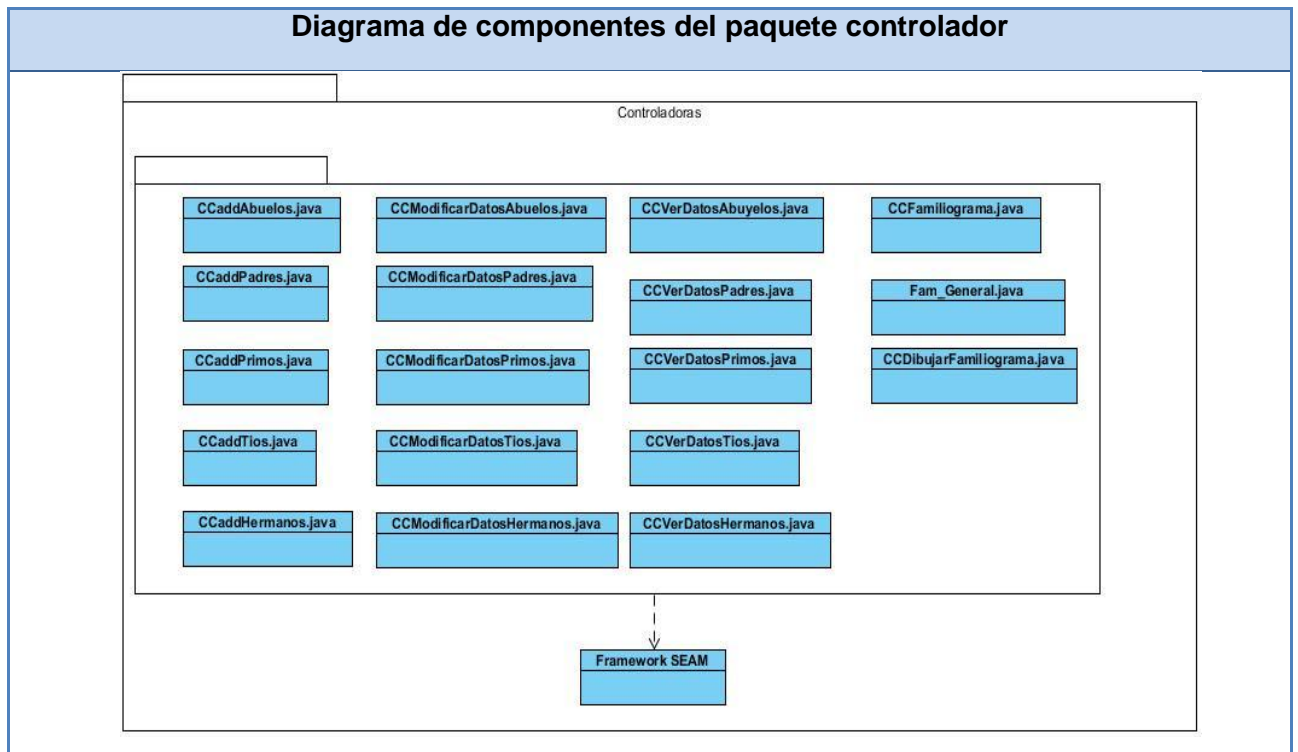


Figura 16: Diagrama de componentes del paquete controlador.

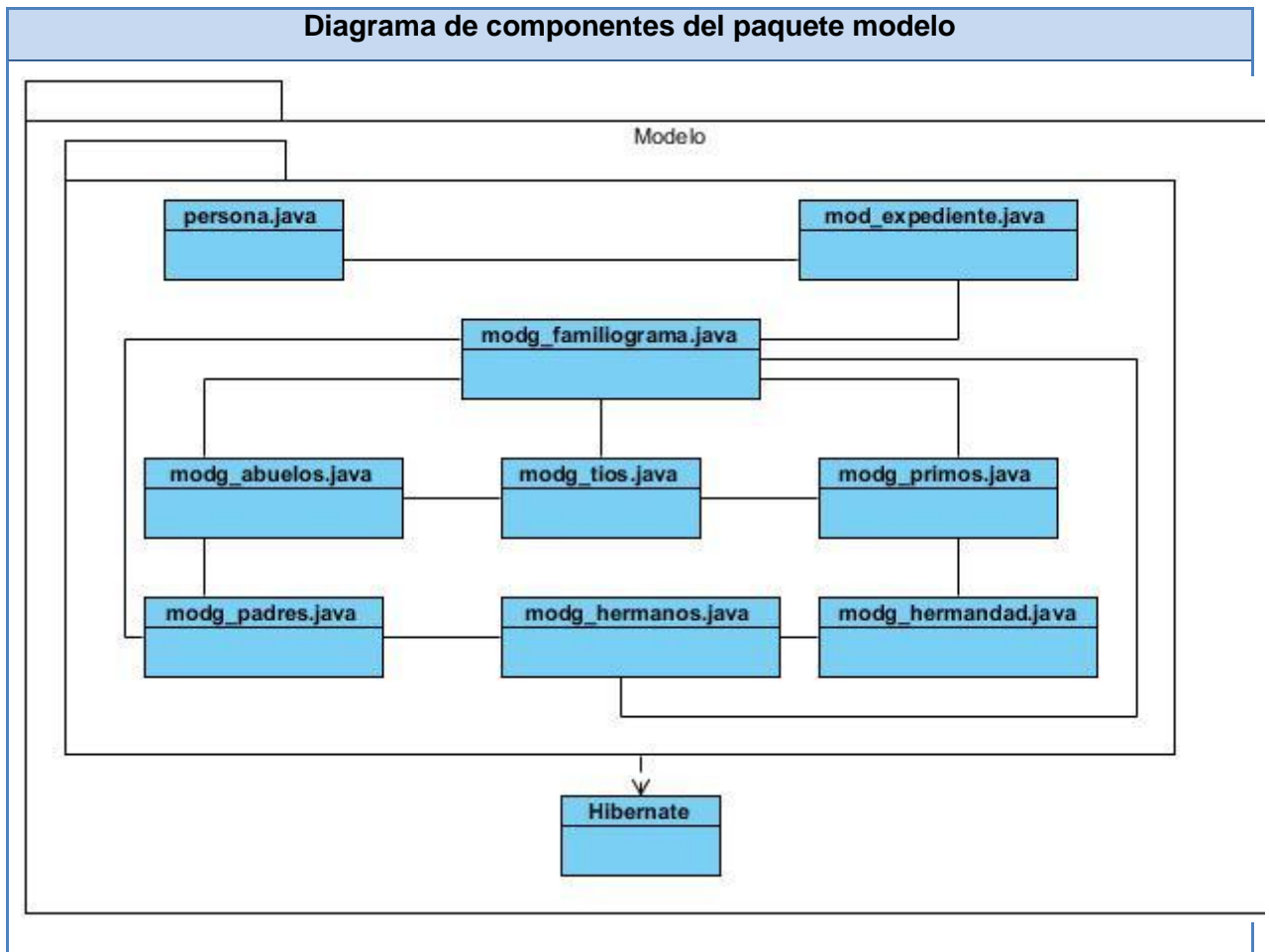


Figura 17: Diagrama de componentes del paquete modelo.

4.1.2. Diagrama de despliegue.

El diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Para el despliegue se deben tomar decisiones sobre los parámetros de la configuración, funcionamiento, asignación de recursos, distribución y concurrencia, (ver figura 18).

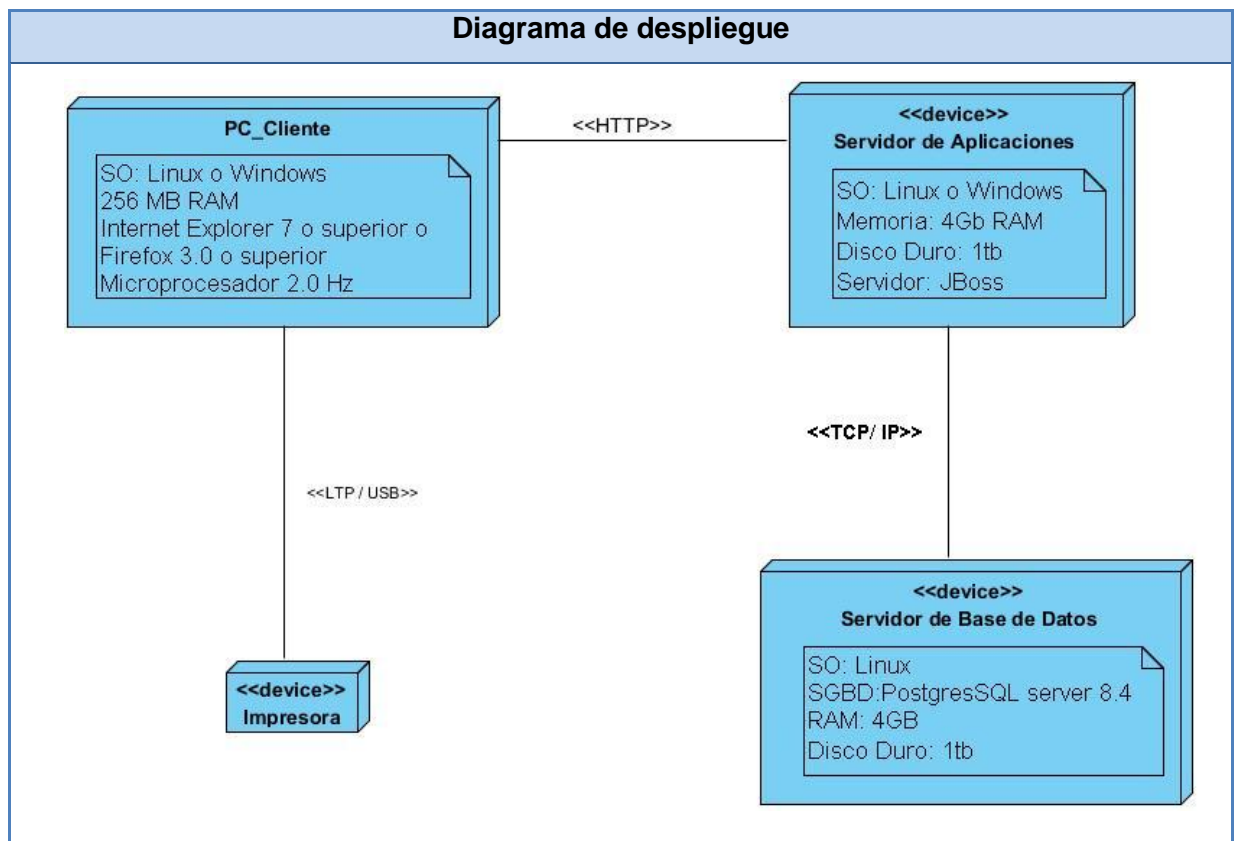


Figura 18: Diagrama de despliegue.

4.2. Tratamiento de errores.

Para lograr una aplicación Web bien diseñada sus formularios deben incluir validaciones de los datos que introducirá el usuario antes de enviar la información al servidor. Una de las formas más viables es realizar tantas comprobaciones como sea posible del lado del cliente, así se mejora el tiempo de respuesta y se reduce la carga del servidor.

En el desarrollo del sistema se validaron todos los campos de entrada al sistema del lado del cliente mediante código JavaScript y del lado del servidor se utilizó la librería Ajax. Las advertencias al usuario ante alguna acción incorrecta se gestionaron a través de un mensaje de alerta que no permite que los datos pasen al servidor. De producirse un error en la base de datos, este es capturado y se especifica al usuario la naturaleza del mismo para que conozca en todo momento cual es la fuente del error. Para la modificación de algún dato del sistema que se encuentre almacenado, mediante un mensaje de confirmación se preguntará al usuario si realmente desea realizar dicha acción, dando la posibilidad de corregir algún error o que se realice dicha operación.

4.3. Seguridad.

La seguridad es uno de los aspectos fundamentales en el desarrollo de un sistema informático, de ella depende la integridad, autenticidad y confiabilidad de su información. En el caso del sistema en cuestión, la seguridad es llevada a cabo por el proyecto alas-HIS. Para esto definieron diferentes tipos de seguridad: acceso al sistema, registro de trazas, administración de seguridad (vista lógica y vista física) y configuración de funcionalidades (39).

Características de la seguridad informática:

Integridad: los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada (40).

Confidencialidad: la información o los activos informáticos son accedidos solo por las personas autorizadas para hacerlo (41).

Disponibilidad: los activos informáticos son accedidos por las personas autorizadas en el momento requerido (42).

Irrefutabilidad: El uso y modificación de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar dicha acción (43).

Acceso al sistema: Se definen roles que permiten un nivel de acceso distinto para cada uno de ellos, con los que se puede acceder a los diferentes módulos y podrán usar las funcionalidades de acuerdo a los permisos de su rol, todo este proceso se realiza a través de usuario y contraseña. El sistema permite: cerrar sesión y salir del módulo.

Registro de trazas: Al usuario realizar alguna acción ya sea inicio o cierre de sesión, acceso a algún módulo en específico, modificación de datos o cualquier otra operación sobre el sistema, este registra una traza en la base de datos.

Administración de seguridad: El sistema da la posibilidad de definir los permisos ya sea dándolos o denegándolos en dependencia de los usuarios o roles en los diferentes módulos o funcionalidades dentro de estos. También permite la eliminación de roles y usuarios de las listas de los que se le negó o permitió algún permiso. Todos estos permisos son registrados por el sistema.

Configuración de funcionalidades: Los usuarios del sistema pueden adicionar o eliminar las diferentes funcionalidades y categorías de un módulo en específico.

4.4. Estrategias de codificación. Estándares y estilos a utilizar.

Para escribir el código fuente de un sistema informático los desarrolladores deben seguir un conjunto de reglas que son establecidas por los estándares de codificación. Esto tiene como objetivo que dicho código sea entendible por cualquier desarrollador del grupo de trabajo, no solo por el que lo creó; garantizando un mantenimiento del sistema más rápido y eficiente, ya sea creando nuevas funcionalidades o modificando las ya existentes.

Para el desarrollo del sistema propuesto se utiliza la siguiente notación:

Notación Camell: se utiliza para denotar variables, parámetros y métodos. Si el identificador es una palabra simple se escribe todo con minúscula, en caso de que sea compuesta, se escribe con minúscula la primera letra del identificador y la que viene a continuación con mayúscula.

A continuación se muestran algunas excepciones para la nomenclatura basadas en los estándares a utilizar:

Idioma: se debe utilizar como idioma el español, las palabras no se acentuarán.

Identación: El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la PC o la configuración de dicha tecla. Los inicios ({} y ()) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.

Comentarios, separadores, líneas, espacios en blanco y márgenes:

Ubicación de comentarios: Se debe comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de datos, y objetivo del parámetro).

Líneas en blanco: Se debe dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

Espacios en blanco: Se deben usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código.

Capítulo 4: Implementación del sistema propuesto.

Aspectos generales: Se debe evitar comentar cada línea de código; cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción. Con respecto a los espacios en blanco, no se deben usar: después del corchete abierto y antes del cerrado de un arreglo, después del paréntesis abierto y antes del cerrado ni antes de un punto y coma.

Bases de Datos, Tablas, Esquemas y Campos

Apariencia de la Base de Datos: Los nombres de las Bases de Datos deben comenzar con mayúscula y con underscore entre palabras.

Ejemplo: "Nueva_Linea_Base".

Apariencia de las tablas: Todas las letras en minúscula. El nombre a emplear para las tablas en caso de que sea un nombre compuesto se utilizará underscore para separarlo.

Ejemplo: "modg_abuelos".

Nombre de los campos: en caso de identificadores. Todos los campos identificadores van a comenzar con el identificador id seguido de underscore y posteriormente el nombre del campo. ejemplo: "id_abuelo".

Sentencias SQL: todas las letras en mayúscula. Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

Aspectos generales: el nombre empleado para las Bases de Datos, las vistas, las tablas, los campos y los procedimientos almacenados, deben permitir que con solo leerlos se conozca el propósito de los mismos.

Nomenclatura

- Los nombres de cada uno de los elementos del programa deben ser significativos; su nombre debe explicar en lo posible el uso del elemento.
- La mayoría de los elementos se deben nombrar usando sustantivos (posiblemente compuestos), o formas verbales en imperativo.
- La forma de construir los nombres será colocando primero el verbo o el sustantivo, seguido de cada uno de sus complementos con la primera letra en mayúscula.

4.4.1. Estándares de Diseño.

En el diseño de una interfaz para un sitio web es necesario tomar en cuenta la interacción de un usuario con un objetivo determinado y un espacio virtual, el sitio web, donde se encuentra el objeto buscado por el usuario. Deben evitarse problemas comunes como el exceso de entretenimiento e interactividad innecesarias, éstos influyen negativamente en el acceso del usuario a la información ofrecida y/o buscada dentro del sitio. Algunos efectos multimedia pueden ser muy atractivos, pero dentro de un sitio público traen consigo problemas de carga lenta del sitio web, poca compatibilidad en diferentes navegadores y versiones de los mismos.

El módulo de Genética del SENDN y por consiguiente las funcionalidades para la gestión de la información genética familiar y posterior esquematización de familiogramas de dicho módulo, sigue las pautas de diseño del HIS. Estas pautas permiten lograr una mayor efectividad en el proceso de trabajo al existir una mayor coherencia formal entre los módulos y páginas del sistema, y que estos sean identificados como parte de un todo. Se han pautado una serie de elementos comunes que facilitarán su reconocimiento y el uso que se haga de ellos.

General




- El género de todos los actores que aparezcan en el prototipo será masculino.
- En el título de los formularios se pondrá el nombre de la acción en infinitivo, la capitalización es igual a la de las etiquetas. Ejemplo: crear familiograma, adicionar abuelo.
- Todos los datos de muestra que se pongan en el prototipo cumplirán con las mismas reglas de capitalización que las etiquetas.

Mensajería

- En las ventanas de Información y Error se mostrará solo un botón de “Aceptar”, centrado en la parte inferior.
- En las ventanas de Advertencia se utilizarán dos botones centrados en la parte inferior y contendrán los textos: “Si” y “No”. El botón “Si” estará a la izquierda y el “No” a la derecha.
- A la izquierda del mensaje aparecerá una imagen que será el ícono que indica el tipo de mensaje (advertencia (⚠), información (💡) y error (❌)).
- En el título de la ventana aparecerá el tipo de mensaje que se muestra.

Capítulo 4: Implementación del sistema propuesto.

Tablas

- Las tablas aparecerán centradas con los colores definidos en el estilo CCS.
- El nombre de la tabla aparecerá en texto alineado a la izquierda y en negrita. Cumpliendo las mismas reglas de capitalización que las etiquetas.
- Las tablas que representen listados de pacientes su nombre comenzará precedido por el texto Listado de <entidad>.
- El nombre de la columna aparecerá alineado a la izquierda y en negro (R0 G0 B0 - #000000). Cumpliendo las mismas reglas de capitalización que las etiquetas. Se le ubicará el título de "Foto" a la columna en la que se muestre la foto del paciente, no así en las columnas donde se ubiquen los íconos de: "Ver" () , "Modificar" () , "Eliminar" () , donde el encabezado de la columna quedará en blanco.
- El contenido de la tabla será alineado a la izquierda. Excepto en las columnas que se muestren datos numéricos que el contenido se alineará a la derecha.

Botones

- Los colores están definidos en el estilo CSS.
- Aparecerán alineados en la parte inferior a la derecha y el orden será de tal forma que las acciones positivas al flujo sea de izquierda a derecha.
- El orden de los botones será de máxima a mínima prioridad de izquierda a derecha, es decir a la izquierda siempre estará el botón de mayor prioridad.

Etiquetas.

- El tamaño y el color están definidos en el estilo CSS.
- Todas las etiquetas estáticas irán en negro (R0 G0 B0 - #000000). La clase de estilo a utilizar es la siguiente:

```
.normalText  
  
{  
  
font-family:Verdana !important;  
  
font-size:10px !important;  
  
color:#000000 !important;
```

```
}
```

- En caso de mostrar información de solo lectura será mostrada de otro color (R92 G92 B92 - #5c5c5c) y se pondrá al lado de la etiqueta que lo identifica la cual estará acompañada de dos puntos. La clase de estilo a utilizar es la siguiente:

```
.dataText  
  
{  
  
font-family:Verdana !important;  
  
font-size:10px !important;  
  
color:#5c5c5c !important;  
  
}
```

- Todas las etiquetas estáticas terminarán con dos puntos (:).
- El texto se escribirá de la forma que sigue la primera letra de la primera palabra en mayúscula el resto del texto de la etiqueta en minúscula, excepto para aquellos atributos específicos que requieran otra capitalización.

Selectores

- Los componentes de selección o combobox visualizarán el texto por defecto: <Selecione>.
- El texto cumplirá con las mismas reglas de capitalización que las etiquetas Verdana 10px.

4.5. Descripción de interfaces del sistema.

La interfaz de usuario (IU) es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. El sistema estará formado por un conjunto de IU amigables y de fácil uso para los usuarios. Las mismas permitirán al genetista ejecutar cada una de las funcionalidades del sistema, donde este tendrá la posibilidad de realizar las operaciones necesarias, para la

Capítulo 4: Implementación del sistema propuesto.

recogida de la Información familiar del paciente y la posterior esquematización del famioliograma. A continuación se describen algunas de las IU.

El sistema brindará la posibilidad de seleccionar el paciente a atender en el día, partiendo de dos listados: paciente sin y con famioliograma realizado, (ver figura 19). Al seleccionar un paciente sin famioliograma realizado, de dicho listado, se comienza con la entrevista del mismo.

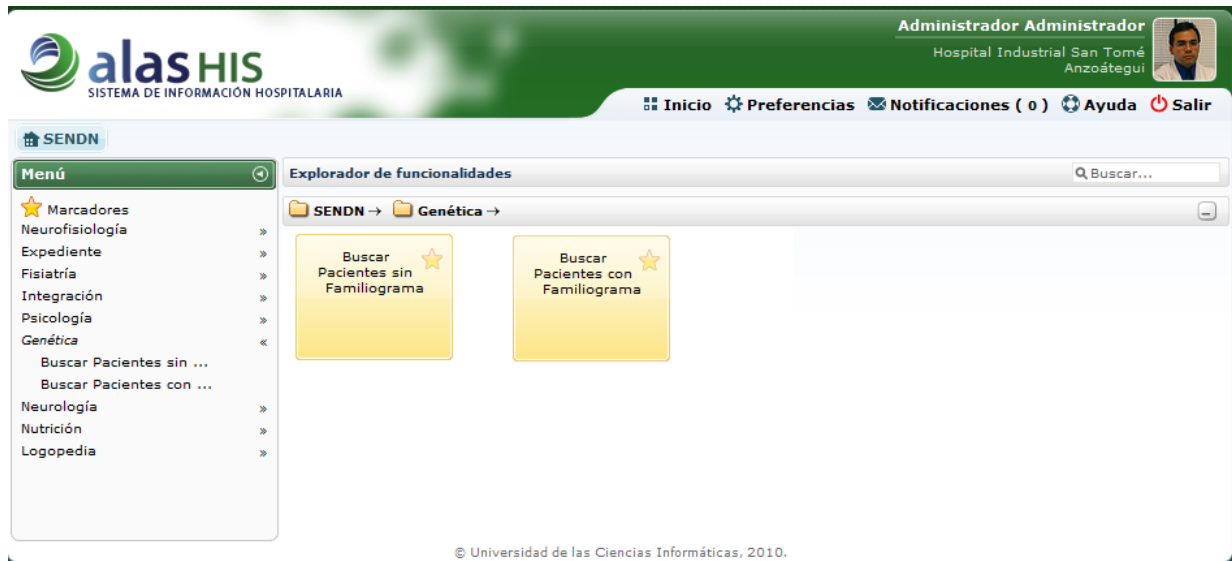


Figura 19: Listado de pacientes a atender en el día sin y con famioliograma realizado.

Una vez seleccionado el paciente comienza la entrevista familiar. En esta se gestiona la información correspondiente a los padecimientos genéticos de sus antecesores. Cuando se gestiona un pariente se ponen de manifiesto disimiles funcionalidades, como son adicionar, listar, ver datos, modificar y eliminar; (ver figura 20).

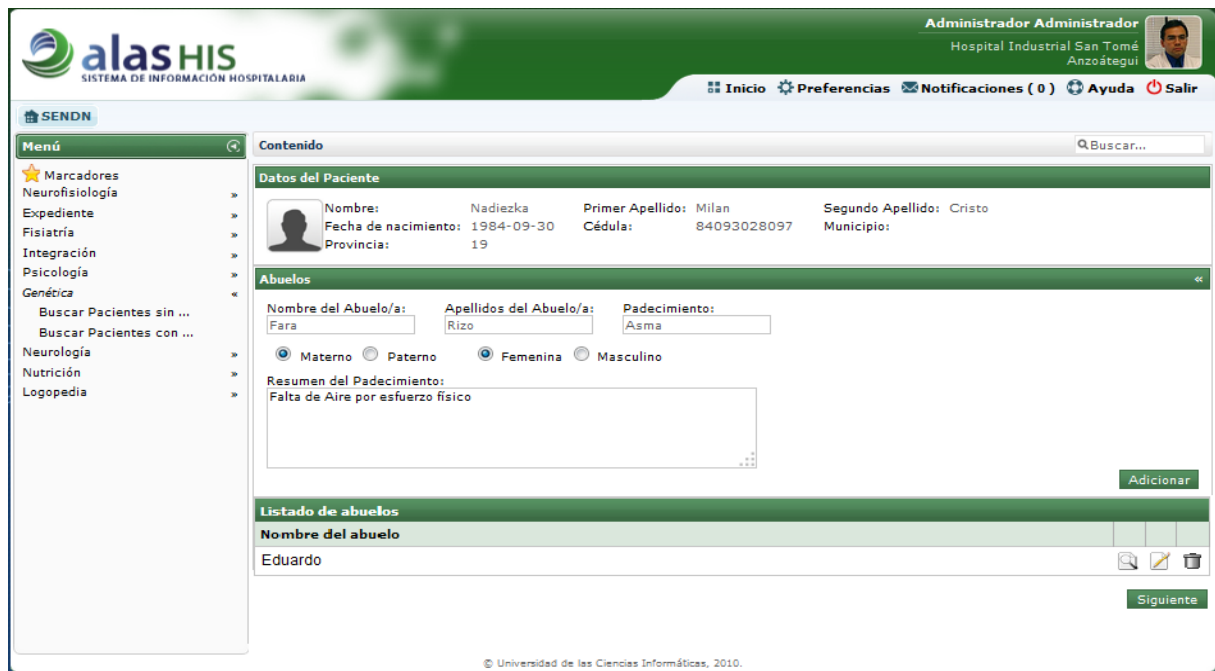


Figura 20: Gestionar abuelos.

Cuando se accede al listado de pacientes con familiograma realizado, ya sea, una vez terminada la entrevista familiar o directamente a través del propio listado, el usuario tiene la posibilidad de generar el familiograma, modificar la información anteriormente recogida en la entrevista y eliminarlo si el especialista considera que el mismo no tiene la validez necesaria para formar parte del estudio. También en estos listados se pueden hacer búsquedas por diferentes criterios, (ver figura 21).

The screenshot displays the 'alasHIS' web application interface. At the top left is the logo and name 'alasHIS SISTEMA DE INFORMACIÓN HOSPITALARIA'. At the top right, it shows the user role 'Administrador Administrador' and the hospital name 'Hospital Industrial San Tomé Anzoátegui'. A navigation bar includes 'Inicio', 'Preferencias', 'Notificaciones (0)', 'Ayuda', and 'Salir'. A left sidebar menu lists various medical specialties like Neurofisiología, Fisiatría, etc. The main content area is titled 'Buscar pacientes a atender' and contains a search form with fields for 'Nombre', 'Apellido1', 'Apellido2', and 'Cédula'. Below the search form is a table titled 'Listado de Pacientes con Familiograma creado.' with columns for 'Foto', 'Nombre', 'Apellido1', 'Apellido2', and 'Carné Identidad'. The table lists three patients: LOLA PEREZ CARBALLO, Maria Gonzalez Ramirez, and Javier Villares Arias. Each row has icons for photo, edit, and delete. The footer of the interface reads '© Universidad de las Ciencias Informáticas, 2010.'

Foto	Nombre	Apellido1	Apellido2	Carné Identidad
	LOLA	PEREZ	CARBALLO	123432567
	Maria	Gonzalez	Ramirez	87314543413145
	Javier	Villares	Arias	85112324826

Figura 21: Listado de pacientes a atender en el día con familiograma realizado.

Una vez generado el familiograma de un determinado paciente, se esquematiza la información genética correspondiente a los padecimientos de sus familiares. Al hacerlo mediante un gráfico, el especialista tendrá la posibilidad de apreciar e interpretar la información de una manera más rápida, (ver figura 22).

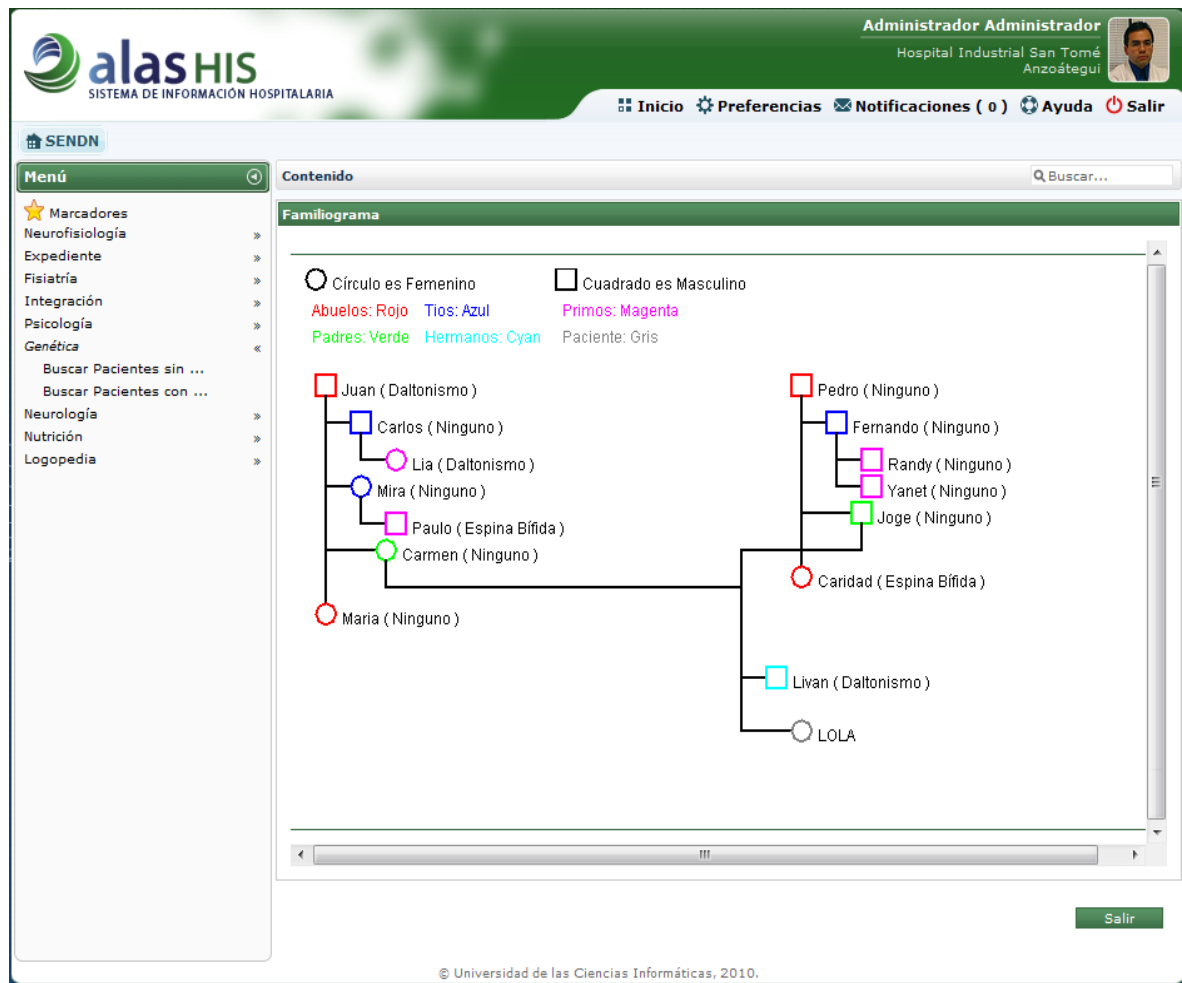


Figura 22: Esquematzación del familiograma.

En este capítulo se analizaron los principales elementos que intervienen en la implementación del sistema; además se mostraron los diagramas que intervienen en cada uno de los modelos para garantizar una mayor comprensión de la estructura del mismo. También se tratan los diferentes errores que pueden ocurrir, para evitar resultados inesperados. Se realizó una descripción de cuáles fueron los estándares de codificación utilizados con el objetivo de que otros desarrolladores puedan asimilar y comprender la implementación realizada. Asimismo se hizo una breve descripción de algunas de las IU del sistema.

4.6. Pruebas de validación del sistema.

Las pruebas de validación se definen como una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan y registran, y se realiza una evaluación de algún aspecto. La prueba es un elemento

crítico para la calidad del sistema. Estas permiten validar y verificar el mismo, determinando si los productos de una fase satisfacen las condiciones de la misma (44).

Estas técnicas se pueden aplicar en cualquiera de los niveles de pruebas (unitarias, integración, aceptación, sistema) con diferentes grados de abstracción en la definición de los casos de prueba.

4.6.1 Pruebas de Caja Negra

Las pruebas de Caja Negra se centran en los requisitos funcionales del sistema. Es decir, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Se trata de un enfoque que intenta descubrir diferentes tipos de errores que no se encuentran con los métodos de caja blanca.

La prueba de caja negra pretende encontrar errores de las siguientes categorías:

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a bases de datos externas.
- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y de terminación.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas fueron usadas:

- ✓ Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- ✓ Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

4.6.1.1 Pruebas de Caja Negra aplicada a los componentes implementados.

Se realizó un caso de prueba para cada requisito implementado. A continuación se muestra el caso de prueba realizado al requisito Adicionar abuelos, (ver tabla 11) que describe cada uno de los escenarios que pueden existir ante las posibles acciones realizadas por el usuario. De manera similar se realizaron los casos de prueba a los requisitos restantes. Para ver los mismos dirigirse al departamento SES, específicamente en el proyecto SENDN, donde se encuentra documentado el expediente de proyecto.

Capítulo 4: Implementación del sistema propuesto.

Tabla 11: Prueba de Caja Negra realizada al requisito Adicionar abuelos.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Adicionar abuelos	Se adiciona un abuelo con los atributos: Nombre Apellidos Padecimiento Materno Masculino	EP 1.1: Adicionar un nuevo abuelo introduciendo datos válidos.	<ul style="list-style-type: none">– Se presiona el botón Adicionar.– Se insertan todos los datos.– Se actualiza el listado de abuelos.
		EP 1.2: Adicionar un nuevo abuelo dejando campos requeridos en blanco.	<ul style="list-style-type: none">– Se presiona el botón Adicionar.– Se no se introducen los datos dejando campos requeridos en blanco.– Se muestra la información correspondiente al error.

Capítulo 4: Implementación del sistema propuesto.

		EP 1.3: Adicionar un nuevo abuelo introduciendo datos inválidos y valores que se encuentran fuera de los límites aceptables.	<ul style="list-style-type: none">– Se presiona el botón Adicionar.– Se intenta introducir los datos inválidos sin éxito alguno.– Se muestra la información correspondiente al error.
		EP 1.4: Adicionar un nuevo abuelo que ya se encuentra registrado en el sistema	<ul style="list-style-type: none">– Se presiona el botón Adicionar.– Se intenta introducir los datos sin éxito alguno.– Se muestra la información correspondiente al error. Reflejando mediante un mensaje la duplicidad de la información.

Como resultado de las pruebas al sistema, del tipo caja negra, donde se aplicaron los diseños de casos de pruebas diseñados, se encontraron por parte del equipo de calidad del proyecto SENDN un número de 16 no conformidades, agrupadas en seis no conformidades de tipo validación, cuatro en opciones que no funcionan y seis en errores de interfaz, las cuales fueron resueltas en su totalidad.

Conclusiones

Luego de la realización del presente trabajo se arribó a las siguientes conclusiones:

- El estudio bibliográfico realizado permitió sustentar los presupuestos que desde la teoría justifican los conceptos que conforman el cuerpo del trabajo, estos permitieron profundizar en el tema de esquematización de familiogramas, para agilizar el proceso de diagnóstico de los pacientes del programa “Renacer Contigo”.
- Con el estudio realizado se demostró que las principales herramientas existentes para la gestión de familiogramas no suplen completamente las necesidades de los genetistas del hospital “William Soler”, por lo que se hizo necesario desarrollar nuevas funcionalidades para permitir la gestión de estos para el módulo Genética del SENDN.
- Las funcionalidades creadas permitirán a los usuarios del módulo Genética del SENDN, la realización de familiogramas de una manera más eficiente evitando la introducción de errores o la pérdida de la información, validándose todas las entradas de datos al sistema.

Recomendaciones

- Incorporar técnicas de Inteligencia Artificial a la esquematización de familiogramas, para facilitar al especialista la obtención del diagnóstico de cada paciente.
- Perfeccionar la visualización de los problemas de salud de los miembros de la familia en la funcionalidad Crear familiograma.

Referencias Bibliográficas

1. Ruz, Fidel Castro. Ideas del Proyecto Futuro. *Juventud Revelde*. 2012.
2. Villavicencio, Dr. Pedro Mestre. *Proyecto Renacer Contigo*. s.l.: FACULTAD “ENRIQUE CABRERA”, 2012.
3. PEDAGOGIA Y ANTROPOLOGIA. [En línea] [Citado el: 10 de 12 de 2012.] http://pedagogiayantropologia.blogspot.com/2011/09/familiograma_01.html.
4. GenoPro - Programa de árboles familiares. [En línea] [Citado el: 12 de 12 de 2012.] <http://www.genopro.com/es/>.
5. WinGeno. [En línea] [Citado el: 12 de 12 de 2012.] <http://wingeno.org>.
6. Rosado Roselló, Reynaldo y Le Sánchez, Lien. *AlasARBOGEN aplicación informática para la representación de árboles genealógicos*. 2008. Tesis.
7. Arrastia Machin, Yosvany y Torres Cruz, Rayko Emilio. *Aplicacion de escritorio para la representacion grafica de arboles genealogicos del sistema alasARBOGEN 2.0*. 2010.
8. Middleware. [En línea] [Citado el: 15 de 12 de 2012.] <http://www.rfidpoint.com/fundamentos/middleware/>.
9. Iqingenieros. [En línea] [Citado el: 15 de 12 de 2012.] <http://iqingenieros.com/inicio/101-p/321-proceso-unificado-de-desarrollo-de-oftware.html>.
10. Ivanex. Sistema de selección de rutas óptimas. [En línea] [Citado el: 15 de 12 de 2012.] <http://ivanex.wikidot.com/metodologia>.
11. Eumed. [En línea] [Citado el: 10 de 1 de 2013.] <http://www.eumed.net/libros/2009c/584/Caracterizacion%20detallada%20de%20la%20metodologia%20de%20desarrollo%20de%20software%20RUP.htm>.
12. Colectivo de autores. *Manual de diagramación de procesos bajo estándar BPMN*.
13. Cristóbal González Almirón. Adictosaltrabajo. [En línea] [Citado el: 10 de 1 de 2013.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava#_Toc225422690.
14. Jboss Enterprise Middleware. [En línea] [Citado el: 15 de 1 de 2013.] <http://www.jboss.com/products/seam/>.
15. AMEBase. *Facelets*. 2009. pág. 1.
16. Basurto, Cristhian Kirs Herrera. *¿Qué ofrece Autentia?* Quito : s.n., 2007.
17. Ramos, Juan Alonso. *Introducción a Ajax4jsf*. 2007.

Referencias Bibliográficas

18. IBM. International Business Machines Corporation. *WebSphere Application Server Community Edition*. [En línea] [Citado el: 14 de 1 de 2013.] <http://publib.boulder.ibm.com/wasce/V2.1.0/es/java-ee-5-certified.html>.
19. Comunidad de Programadores de Software Libre del Ecuador. C O P L E C. [En línea] [Citado el: 16 de 1 de 2013.] <http://www.coplec.org/?q=book/export/html/240>.
20. Oracle Technology Network. [En línea] [Citado el: 16 de 1 de 2013.] <http://www.oracle.com/technetwork/java/javaee/ejb/>.
21. Colectivo de autores. *The Java EE 5Tutorial. For Sun Java System Application Server 9.1*. s.l. 2010. págs. 40-41.
22. Ayuda BitTorrent. [En línea] [Citado el: 17 de 1 de 2013.] <http://www.ayudabittorrent.com/jre>.
23. Web Taller. [En línea] [Citado el: 17 de 1 de 2013.] <http://www.webtaller.com/construccion/lenguajes/html/lecciones/que-es-xhtml.php>.
24. Java. [En línea] [Citado el: 17 de 1 de 2013.] <http://java.com/es/about/>.
25. Scribd. [En línea] [Citado el: 18 de 1 de 2013.] <http://es.scribd.com/doc/2080534/UML>.
26. Capote, Olga Pons. *Introducción a las Bases de Datos. El Modelo Relacional*. s.l. : Thomson Learning Ibero, 2005.
27. Llegó la hora del PostgreSQL. [En línea] [Citado el: 20 de 1 de 2013.] www.http-peru.com/postgresql1.php.
28. Visual Paradigm. [En línea] [Citado el: 21 de 1 de 2013.] <http://www.visual-paradigm.com/product/vpuml>.
29. Eclipse. [En línea] [Citado el: 21 de 1 de 2013.] <http://www.eclipse.org>.
30. Guía documentada para Ubuntu. [En línea] [Citado el: 22 de 1 de 2013.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.
31. Manuel Resinas de Reyna, Manuel J. Recena Soto. *Introducción a los Sistemas de Control de Versiones*. Universidad de Sevilla : s.n., 2005.
32. Modelo de negocio. [En línea] [Citado el: 30 de 1 de 2013.] <http://manuelgross.bligoo.com/que-es-un-modelo-de-negocio-la-fuente-de-tu-competitividad>.
33. Requerimientos funcionales. [En línea] [Citado el: 1 de 2 de 2013.] <http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>.
34. Modelo de Caso de Uso. [En línea] [Citado el: 3 de 2 de 2013.] http://www.sparxsystems.com.ar/resources/tutorial/use_case_model.html.
35. Universidad de Alcalá. [En línea] [Citado el: 3 de 2 de 2013.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.

Referencias Bibliográficas

36. Comunidad Virtual del Departamento de Sistemas Universidad del Cauca - Popayán - Colombia. [En línea] [Citado el: 13 de 2 de 2013.] http://pis.unicauca.edu.co/moodle/file.php/291/Patron_Disenio_MVC.pdf.
37. El mundo informático. [En línea] 17 de 8 de 2008. [Citado el: 5 de 2 de 2013.] <http://jorgesaavedra.wordpress.com>.
38. GONZÁLEZ, RAÚL. Modelo de datos. [En línea] 5 de 2009. [Citado el: 6 de 2 de 2013.] <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos/>.
39. Colectivo de autores. *Seguridad en los Sistemas Operativos*. 2007. págs. 35-38.
40. —. *Seguridad en los Sistemas Operativos*. 2007. pág. 38.
41. —. *Seguridad en los Sistemas Operativos*. 2007. págs. 38-39.
42. —. *Seguridad en los Sistemas Operativos*. 2007. pág. 40.
43. —. *Seguridad en los Sistemas Operativos*. 2007. pág. 41.
44. Cortés Guzmán, Oscar Hernando. Willydev. [En línea] 20 de 4 de 2004. [Citado el: 15 de 3 de 2013.] http://www.willydev.net/descargas/oguzman-diseno_pruebas.pdf.

Bibliografía

AMEBase. Facelets. 2009. pág. 1.

Arrastia Machin, Yosvany y Torres Cruz, Rayko Emilio. Aplicación de escritorio para la representación gráfica de árboles genealógicos del sistema alasARBOGEN 2.0. 2010.

Ayuda BitTorrent. [En línea] [Citado el: 17 de 1 de 2013.] <http://www.ayudabittorrent.com/jre>.

Basurto, Cristhian Kirs Herrera. ¿Qué ofrece Autentia? Quito: s.n., 2007.

Capote, Olga Pons. Introducción a las Bases de Datos. El Modelo Relacional. s.l. : Thomson Learning Ibero, 2005.

Colectivo de autores. Manual de diagramación de procesos bajo estándar BPMN.

Colectivo de autores. Seguridad en los Sistemas Operativos. 2007. pág. 38.

Colectivo de autores. Seguridad en los Sistemas Operativos. 2007. pág. 40.

Colectivo de autores. Seguridad en los Sistemas Operativos. 2007. pág. 41.

Colectivo de autores. Seguridad en los Sistemas Operativos. 2007. págs. 38-39.

Colectivo de autores. Seguridad en los Sistemas Operativos. 2007. págs. 35-38.

Colectivo de autores. The Java EE 5Tutorial. For Sun Java System Application Server 9.1. s.l. 2010. págs. 40-41.

Comunidad de Programadores de Software Libre del Ecuador. C O P L E C. [En línea] [Citado el: 16 de 1 de 2013.] <http://www.coplec.org/?q=book/export/html/240>.

Comunidad Virtual del Departamento de Sistemas Universidad del Cauca - Popayán - Colombia. [En línea] [Citado el: 13 de 2 de 2013.] http://pis.unicauca.edu.co/moodle/file.php/291/Patron_Disenio_MVC.pdf.

Cortés Guzmán, Oscar Hernando. Willydev. [En línea] 20 de 4 de 2004. [Citado el: 15 de 3 de 2013.] http://www.willydev.net/descargas/oguzman-diseno_pruebas.pdf.

Cristóbal González Almirón. Adictosaltrabajo. [En línea] [Citado el: 10 de 1 de 2013.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava#_Toc225422690.

Eclipse. [En línea] [Citado el: 21 de 1 de 2013.] <http://www.eclipse.org>.

El mundo informático. [En línea] 17 de 8 de 2008. [Citado el: 5 de 2 de 2013.] <http://jorgesaavedra.wordpress.com>.

Entorno Virtual de Aprendizaje. BPMN. [En línea] 2013. http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_2/Otros_materiales/2_Obtencion_de_Requisitos/BPMN/BPMN.pdf.

Eumed. [En línea] [Citado el: 10 de 1 de 2013.] <http://www.eumed.net/libros/2009c/584/Caracterizacion%20detallada%20de%20la%20metodologia%20de%20desarrollo%20de%20software%20RUP.htm>.

GenoPro - Programa de árboles familiares. [En línea] [Citado el: 12 de 12 de 2012.] <http://www.genopro.com/es/>.

GONZÁLEZ, RAÚL. Modelo de datos. [En línea] 5 de 2009. [Citado el: 6 de 2 de 2013.] <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos/>.

Guía documentada para Ubuntu. [En línea] [Citado el: 22 de 1 de 2013.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.

IBM. International Business Machines Corporation. WebSphere Application Server Community Edition. [En línea] [Citado el: 14 de 1 de 2013.] <http://publib.boulder.ibm.com/wasce/V2.1.0/es/java-ee-5-certified.html>.

Iqingenieros. [En línea] [Citado el: 15 de 12 de 2012.] <http://iqingenieros.com/inicio/101-p/321-proceso-unificado-de-desarrollo-de-oftware.html>.

Ivanex. Sistema de selección de rutas óptimas. [En línea] [Citado el: 15 de 12 de 2012.] <http://ivanex.wikidot.com/metodologia>.

Java. [En línea] [Citado el: 17 de 1 de 2013.] <http://java.com/es/about/>.

Jboss Enterprise Middleware. [En línea] [Citado el: 15 de 1 de 2013.] <http://www.jboss.com/products/seam/>.

Llegó la hora del PostgreSQL. [En línea] [Citado el: 20 de 1 de 2013.] www.http-peru.com/postgresql1.php.

Manual de diagramación de procesos bajo estándar BPMN.

Manuel Resinas de Reyna, Manuel J. Recena Soto. Introducción a los Sistemas de Control de Versiones. Universidad de Sevilla: s.n., 2005.

Middleware. [En línea] [Citado el: 15 de 12 de 2012.] <http://www.rfidpoint.com/fundamentos/middleware/>.

Modelo de Caso de Uso. [En línea] [Citado el: 3 de 2 de 2013.] http://www.sparxsystems.com.ar/resources/tutorial/use_case_model.html.

Modelo de negocio. [En línea] [Citado el: 30 de 1 de 2013.] <http://manuelgross.bligoo.com/que-es-un-modelo-de-negocio-la-fuente-de-tu-competitividad>.

Oracle Technology Network. [En línea] [Citado el: 16 de 1 de 2013.] <http://www.oracle.com/technetwork/java/javaee/ejb/>.

PEDAGOGIA Y ANTROPOLOGIA. [En línea] [Citado el: 10 de 12 de 2012.] http://pedagogiayantropologia.blogspot.com/2011/09/familiograma_01.html.

Pressman, Roger. Ingeniería del Software, Un Enfoque Práctico. 2005.

Ramos, Juan Alonso. Introducción a Ajax4jsf. 2007.

Requerimientos funcionales. [En línea] [Citado el: 1 de 2 de 2013.] <http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>.

Rosado Roselló, Reynaldo y Le Sánchez, Lien. AlasARBOGEN aplicación informática para la representación de árboles genealógicos. 2008. Tesis.

Ruz, Fidel Castro. Ideas del Proyecto Futuro. Juventud Revelde. 2012.

Scribd. [En línea] [Citado el: 18 de 1 de 2013.] <http://es.scribd.com/doc/2080534/UML>.

Sommerville, Ian. Ingeniería del Software. 2005. http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_2/Sommerville/Sommerville_Parte_II_Requerimientos.pdf.

Universidad de Alcalá. [En línea] [Citado el: 3 de 2 de 2013.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.

Villavicencio, Dr. Pedro Mestre. Proyecto Renacer Contigo. s.l. : FACULTAD “ENRIQUE CABRERA”, 2012.

Visual Paradigm. [En línea] [Citado el: 21 de 1 de 2013.] <http://www.visual-paradigm.com/product/vpuml>.

Web Taller. [En línea] [Citado el: 17 de 1 de 2013.] <http://www.webtaller.com/construccion/lenguajes/html/lecciones/que-es-xhtml.php>.

WinGeno. [En línea] [Citado el: 12 de 12 de 2012.] <http://wingeno.org>.

Anexos

Anexo 1: Descripción de casos de uso del sistema.

Caso de uso Listar pacientes sin familiograma.

CU - 2	Listar pacientes sin familiograma
Actor	Genetista
Descripción	El caso de uso inicia cuando el actor accede a la opción Realizar familiograma, el sistema muestra el criterio de búsqueda, el actor introduce dicho criterio, el sistema muestra el listado de los pacientes con familiogramas realizados según el criterio de búsqueda, el caso de uso termina.
Referencia	RF 2

Caso de uso Ver datos de abuelos.

CU - 6	Ver datos de abuelos
Actor	Genetista
Descripción	El caso de uso inicia cuando el actor inserta los datos de los abuelos, al dar en la opción Aceptar se muestra los datos de los abuelos insertados.
Referencia	RF 6

Caso de uso Modificar datos de los abuelos.

CU - 7	Modificar datos de los abuelos
Actor	Genetista
Descripción	El caso de uso inicia cuando el actor accede a la opción Modificar datos de los abuelos. El sistema muestra los datos de los abuelos y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes. El actor modifica los datos que necesita, el sistema actualiza los datos de los abuelos y el caso de uso termina.

Referencia	RF 7
-------------------	------

Caso de uso Insertar datos de los tíos.

CU – 9	Insertar datos de los tíos
Actor	Genetista
Descripción	El caso de uso inicia cuando el actor accede a la opción insertar datos de tíos, el sistema brinda la posibilidad de introducir los datos correspondientes a los tíos del paciente, el actor introduce los datos, el sistema inserta los datos, el caso de uso termina.
Referencia	RF 9

Caso de uso Eliminar datos de los tíos.

CU – 13	Eliminar datos de los tíos
Actor	Genetista
Descripción	El caso de uso inicia cuando el actor accede a la opción Eliminar datos de los tíos. El sistema pregunta si se desean eliminar los datos. El sistema elimina los datos de los tíos y el caso de uso termina.
Referencia	RF 13

Anexo 2: Especificación de los casos de uso.

Caso de uso Listar pacientes sin familiograma.

CU-2	Listar pacientes sin familiograma
Propósito	Listar los pacientes registrados a los cuales no se les ha realizado aun el familiograma.
Actores	Genetista

<p>Resumen: El caso de uso inicia cuando el actor accede a la opción Realizar familiograma, el sistema muestra el criterio de búsqueda, el actor introduce dicho criterio, el sistema muestra el listado de los pacientes con familiogramas realizados según el criterio de búsqueda, el caso de uso termina.</p>	
Precondiciones	Debe haber pacientes insertados en el sistema.
Referencias	RF 2
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor accede a la opción Realizar Familiograma.	
	<p>2. Brinda la posibilidad de introducir los criterios elementales de búsqueda. Busca los posibles pacientes a realizarles un familiograma.</p> <p>Cancelar operación. Ver Alternativa 1: “Cancelar operación.”</p>
3. Introduce o selecciona los datos que considera como criterios para realizar una búsqueda y selecciona la opción de Buscar.	
	<p>4. Si no se encuentra ningún resultado de la consulta que cumpla con los criterios de búsqueda. Ver Alternativa 2: “No se encuentra información que cumpla con los criterios de búsqueda.”</p>
	<p>5. Muestra un listado de todos los pacientes que cumplen</p>

	con los criterios de búsqueda.
Flujo alternativo	
Alternativa 1. "Cancelar operación."	
Acción del actor	Respuesta del sistema
1. Selecciona la opción de Cancelar operación.	
	2. Regresa a la vista anterior.
Alternativa 2. "No se encuentra información que cumpla con los criterios de búsqueda."	
	1. Muestra el mensaje de información "No se encontró información que cumpla con los criterios de búsqueda."
	2. Regresa al paso 2 del Flujo Normal de Eventos.

Caso de uso Ver detalles de abuelos.

CU-6	Ver detalles de abuelos	
Propósito	Ver los datos de un abuelo insertado.	
Actores	Genetista	
Resumen: El caso de uso inicia cuando el actor inserta los datos de los abuelos, al dar en la opción Aceptar se muestra los datos de los abuelos insertados.		
Precondiciones	Debe haberse insertado el abuelo.	
Referencias	RF 6	
Acción del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el actor acepta Insertar los datos de los abuelos,		

accediendo a la opción de Ver detalles de abuelos.	
	<p>2. Muestra los datos de los abuelos correspondientes.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Modificar Datos de Abuelos. Ver alternativa 1: “Modificar datos de los Abuelos.” • Salir de la vista actual.
3. Selecciona la opción de salir de la vista actual.	
	4. Muestra la vista anterior.
	5. El caso de uso termina.
Flujo alternativo	
Alternativa 1. “Cancelar operación.”	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Modificar datos de los Abuelos.	
	2. Modifica datos de los Abuelos. Ver caso de uso: Modificar datos de los Abuelos.

Caso de uso Modificar datos de los abuelos.

CU-7	Modificar datos de los abuelos
Propósito	Modificar los datos insertados correspondientes a los abuelos.
Actores	Genetista
Resumen: El caso de uso inicia cuando el actor accede a la opción Modificar datos de los abuelos. El sistema muestra los datos de los abuelos y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes. El actor modifica los datos que necesita, el sistema actualiza los datos de los abuelos y el caso de uso termina.	

Precondiciones	Debe haberse insertado el abuelo.	
Referencias	RF 7	
Acción del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el actor accede a la opción Modificar datos de los abuelos.		
	<p>2. Brinda la posibilidad de cambiar los valores existentes e introducir nuevos.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar las modificaciones. • Cancelar operación. Ver Alternativa 1: “Cancelar operación.” 	
3. Modifica los datos que necesita y selecciona la opción de aceptar las modificaciones.		
	4. Valida los datos. Si hay datos incompletos, ver Alternativa 2: “Existen datos incompletos.”	
	5. Actualiza los datos de los abuelos.	
	6. Muestra el mensaje de información “Se han actualizado los cambios sobre los datos de los abuelos.”	
	7. El caso de uso termina.	
Flujo alternativo		
Alternativa 1. “Cancelar operación.”		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Cancelar operación.		

	2. Regresa a la vista anterior.
Alternativa 2. “Existen datos incompletos.”	
	1. Muestra el mensaje de error “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
	2. Muestra un indicador sobre los campos vacíos.

Caso de uso Eliminar datos de los abuelos.

CU-8	Eliminar datos de los abuelos	
Propósito	Eliminar los datos insertados correspondientes a los abuelos.	
Actores	Genetista	
Resumen: El caso de uso inicia cuando el actor accede a la opción Eliminar datos de los abuelos. El sistema pregunta si se desean eliminar los datos. El sistema elimina los datos de los abuelos y el caso de uso termina.		
Precondiciones	Debe haberse insertado el abuelo.	
Referencias	RF 8	
Acción del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el actor accede a la opción Eliminar datos de los abuelos.		
	2. Brinda la posibilidad de eliminar los datos existentes de los abuelos. Y permite: <ul style="list-style-type: none"> • Eliminar los datos. • Cancelar operación. Ver Alternativa 1: “Cancelar operación.” 	
3. Acepta la eliminación de los datos.		

	4. Elimina los datos de los abuelos.
	5. Muestra el mensaje de información "Se han eliminado los datos de los abuelos."
	6. El caso de uso termina.
Flujo alternativo	
Alternativa 1. "Cancelar operación."	
Acción del actor	Respuesta del sistema
1. Selecciona la opción de Cancelar operación.	
	2. Regresa a la vista anterior.

Anexo 3: Diagrama de la clases del diseño.

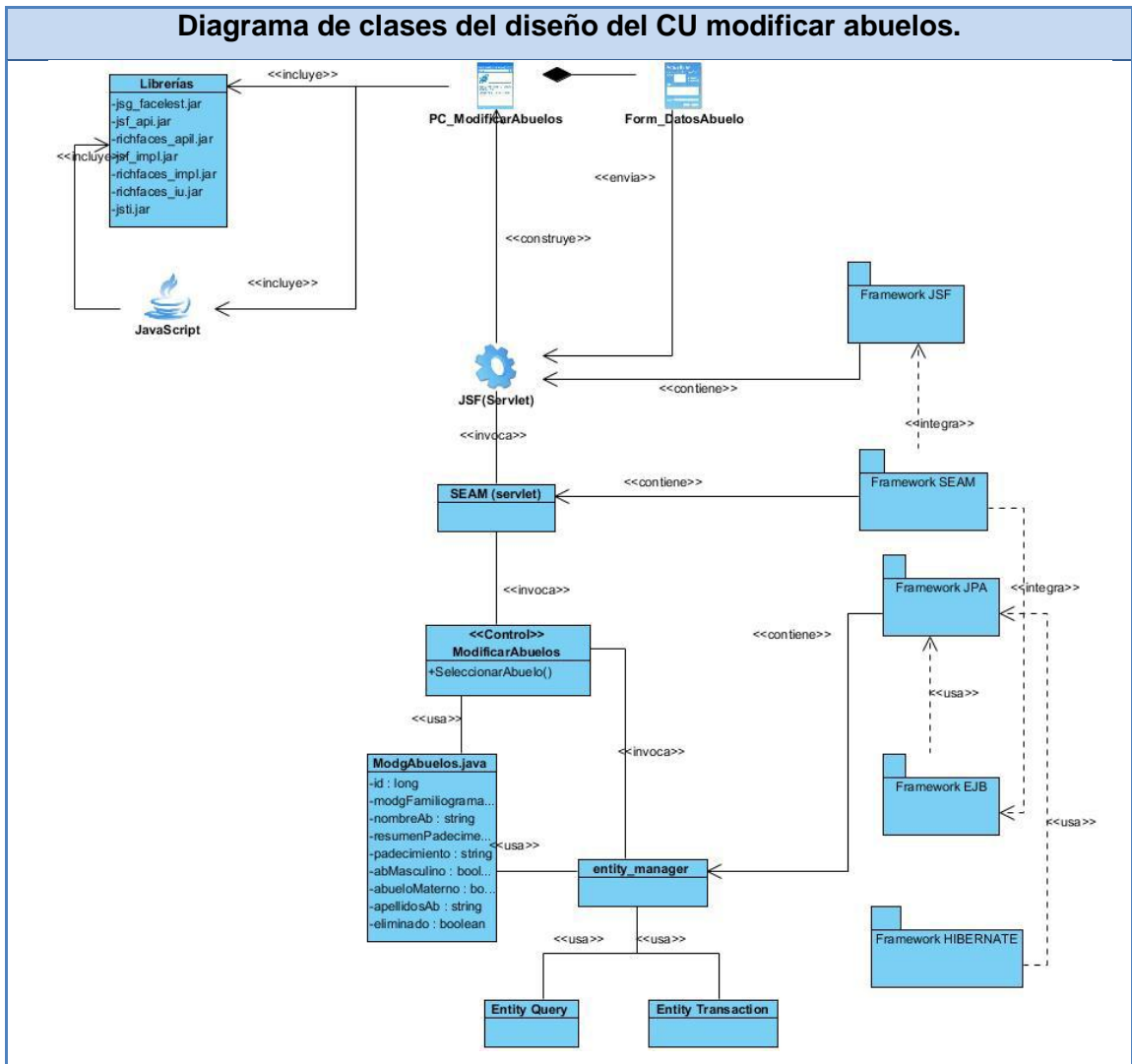
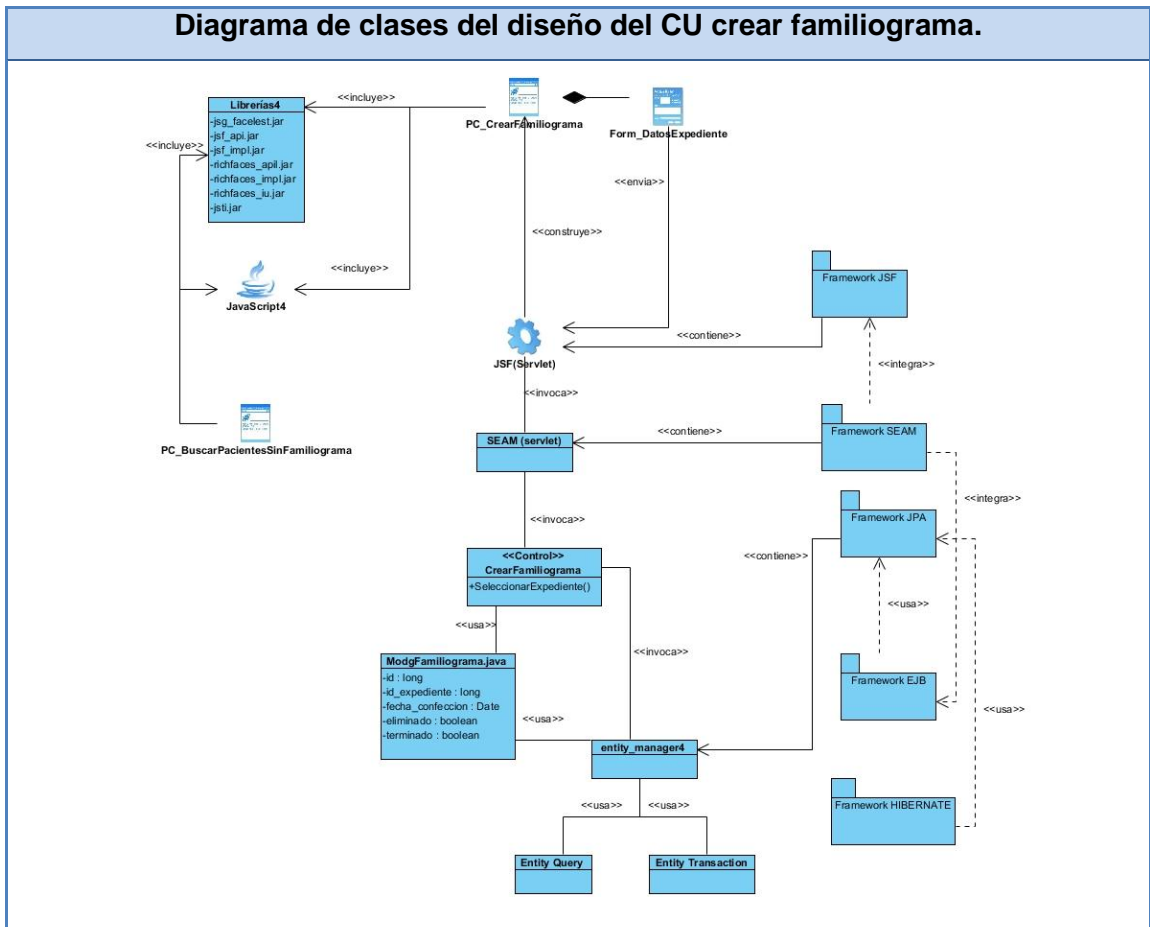
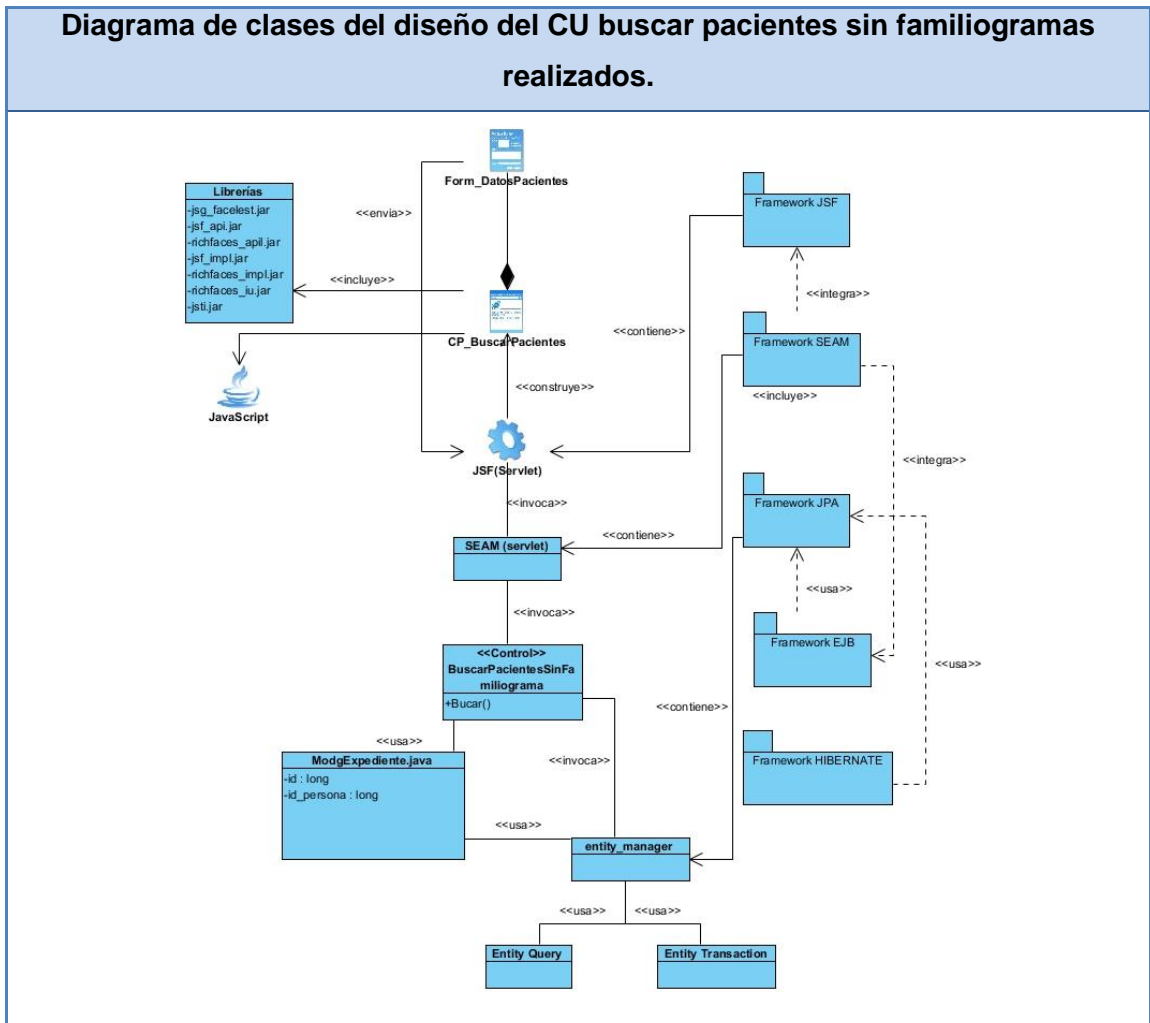


Diagrama de clases del diseño del CU crear familiograma.





Anexo 4: Modelo de datos.

Tabla modg_abuelos.

Nombre de la clase: modg_abuelos		
Descripción: Tabla que almacena la información de los abuelos.		
Atributo	Tipo	Descripción
id	serial	En este atributo se almacena el identificador del abuelo.
id_familiograma	long	Identifica a que familiograma pertenece el

		abuelo. Este atributo es llave foránea que se absorbe de la tabla modg_familiograma.
nombre_ab	string	Guarda el nombre del abuelo.
apellidos_ab	string	Guarda los apellidos del abuelo.
ab_masculino	boolean	Identifica si el género del abuelo/a (masculino o femenino), si es verdadero es masculino.
padecimiento	string	Guarda el nombre del padecimiento del abuelo.
resumen_padecimiento	string	Guarda el resumen del padecimiento del abuelo
abuelo_materno	boolean	Identifica si el abuelo es materno o paterno. Si es verdadero es porque es materno.
eliminado	boolean	Dice si el abuelo está eliminado o no.

Tabla modg_hermanos.

Nombre de la clase: modg_hermanos		
Descripción: Tabla que almacena la información de los hermanos.		
Atributo	Tipo	Descripción
id	serial	En este atributo se almacena el identificador del hermano.
id_familiograma	long	Identifica a que familiograma pertenece el

		hermano. Este atributo es llave foránea que se absorbe de la tabla modg_familiograma.
id_padres	long	Identifica a los padres del hermano del niño. Este atributo es llave foránea que se absorbe de la tabla modg_padres.
id_hermandad_hermanos	long	Identifica a los otros hermanos del niño. Este atributo es llave foránea que se absorbe de la tabla modg_hermandad.
nombre_hermano	string	Guarda el nombre del hermano.
apellidos_hermano	string	Guarda los apellidos del hermano.
padecimiento	string	Guarda el nombre del padecimiento del hermano.
resumen_padecimiento	string	Guarda el resumen del padecimiento del hermano
hermano_masculino	boolean	Identifica si el hermano es de sexo masculino o femenino. Si es verdadero es porque es masculino.
hermanos_por_parte	Integer	Especifica si son hermanos carnales o por parte de madre o de padre.
eliminado	boolean	Dice si el hermano está eliminado o no.

Tabla modg_primos.

Nombre de la clase: modg_primos		
Descripción: Tabla que almacena la información de los primos.		
Atributo	Tipo	Descripción
id	serial	En este atributo se almacena el identificador del primo.
id_familiograma	long	Identifica a que familiograma pertenece el primo. Este atributo es llave foránea que se absorbe de la tabla modg_familiograma.
id_tio_su_padre	long	Identifica a los padres del primo del niño. Este atributo es llave foránea que se absorbe de la tabla modg_padres.
nombre_primo	string	Guarda el nombre del primo.
apellidos_primo	string	Guarda los apellidos del primo.
padecimiento	string	Guarda el nombre del padecimiento del primo.
resumen_padecimiento	string	Guarda el resumen del padecimiento del primo
primo_materno	boolean	Identifica si el primo es por parte de madre o de padre. Si es verdadero es porque es por parte de madre.
primo_masculino	boolean	Identifica si el primo es de

		sexo masculino o femenino. Si es verdadero es porque es masculino.
eliminado	boolean	Dice si el primo está eliminado o no.

Tabla modg_familiograma.

Nombre de la clase: modg_familiograma		
Descripción: Tabla que almacena los familiogramas.		
Atributo	Tipo	Descripción
id	long	En este atributo se almacena el identificador del familiograma.
id_expediente	long	Identifica a que persona pertenece el familiograma. Este atributo es llave foránea que se absorbe de la tabla modg_expediente.
fecha_confeccion	Date	Almacena la fecha en que fue confeccionado el familiograma.
eliminado	boolean	Dice si el familiograma está eliminado o no.

Glosario de términos.

Árbol genealógico: Un árbol genealógico es un esquema gráfico (genograma) que representa las relaciones familiares entre personas.

El árbol genealógico permite situar a nuestros antepasados en relación a nosotros y nos permite entender con un vistazo quién es padre de quién, hermano, tío.

La representación gráfica de las relaciones familiares puede ser simple o muy compleja. En la infancia iniciamos la curiosidad por la estructura familiar, un árbol genealógico sencillo nos ayudará a explicar las relaciones más cercanas e inmediatas. Por otro lado, el árbol puede ser un esquema muy completo tanto por la cantidad de generaciones y relaciones representadas como por la información adicional que puede llegar a contener. Los árboles son una herramienta imprescindible para la investigación genealógica.

Brecha digital: Se define como la separación que existe entre las personas (Comunidades, estados, países) que utilizan las Tecnologías de la Información y Comunicación (TIC) como una parte rutinaria de su vida diaria y aquellas que no tienen acceso a las mismas y que aunque las tengan no saben cómo utilizarlas.

La brecha digital es para las TIC lo mismo que es el cáncer o el sida para la medicina. Una enfermedad que eliminando poco a poco los países pobres de los beneficios de las Tecnologías y del progreso económico.

Casos de uso: Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Epidemiología: La Epidemiología. Es una ciencia básica del campo de la Salud Pública que ha alcanzado importantes desarrollo. En Cuba se define como una rama de la ciencia médica cuya metodología de trabajo permite la investigación y el conocimiento de las causas objetivas y condiciones de cualquier problema de salud que afecte a grupos humanos, así como el planteamiento de soluciones adecuadas.

Esquemmatización: Acción y efecto de esquematizar. Es una técnica de análisis textual que refleja la relación entre los elementos constitutivos del texto, es decir, establece la relación entre conceptos. Determina cuáles son las ideas principales y qué relaciones existen entre ellas.

Framework: Un framework o marco de trabajo, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de sistemas informáticos concretos, que puede servir de base para la organización y desarrollo de sistemas. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

Genética: Genética. Proviene del término "Gen" y significa "descendencia". Es el campo de las ciencias biológicas que trata de comprender cómo la herencia biológica es transmitida de una generación a la siguiente, y cómo se efectúa el desarrollo de las características que controlan estos procesos.

Genotipo: En Biología, el genotipo resulta ser el conjunto de genes característicos de cada especie, vegetal o animal, es decir, el genotipo son los genes en formato de ADN que un animal, un vegetal o un ser humano recibe de herencia de parte de sus dos progenitores, madre y padre, y que por tanto se encuentra conformado por las dos dotaciones de cromosomas que contienen la información genética del ser en cuestión.

Neonatal: Un recién nacido o neonato se considera todo niño que tenga hasta un mes de vida fuera del útero y todo lo relacionado con este período de tiempo se entiende como neonatal, dígase "cuidados neonatales" se refiere al cuidado del neonato.

Neurodesarrollo: Neurodesarrollo son los mecanismos a través de los cuales se organiza el Sistema Nervioso como un sistema de relación. El propósito del Neurodesarrollo en la práctica clínica es reconocer, anticipar y/o identificar los posibles riesgos y/o alteraciones del Sistema Nervioso y subsistemas involucrados que puedan ejercer un impacto negativo en el desarrollo integral del niño, permitiendo tomar decisiones oportunas e intervenir.

Llave foránea: Es llamada clave externa, es uno o más campos de una tabla que hacen referencia al campo o campos de clave principal de otra tabla, una clave externa indica cómo están

Glosario de Términos

relacionadas las tablas. Los datos en los campos de clave externa y clave principal deben coincidir, aunque los nombres de los campos no sean los mismos.

SENDN: El proyecto Sistema de Evaluación del Neurodesarrollo en Niños (SENDN) es un sistema basado en el Proyecto Renacer Contigo del Hospital Pediátrico Universitario William Soler, desarrollado como proyecto de investigación y aprobado por la Dirección Nacional Materno Infantil. Este proyecto fue desarrollado por el Grupo Interdisciplinario de Atención Temprana perteneciente a dicho hospital. Estos fueron motivados por la influencia que sobre la evolución del niño en riesgo ejercen los programas de atención temprana utilizados hace ya varios años a nivel mundial; además de la pobre presencia de estos programas en los servicios de atención al enfermo en el país y la necesidad cada vez mayor de mejorar la calidad de vida de la población infantil en el país, ya con tasas de mortalidad comparables con los países desarrollados.