



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7

Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas

Componente web Análisis de Familiograma para el Sistema Integral
para la Atención Primaria de Salud

Autor:

Rolando Michel Fernández Pérez

Tutores:

Ing. Yudenia Hernández Gallego

Ing. Reynier Alejandro Rodríguez Ronquillo.

La Habana, junio de 2011

“Año 53 de la Revolución”

Datos de Contacto

Ing. Yudenia Hernández Gallego, Graduada de Ingeniera en Ciencias Informáticas en el año 2008. Labora en la Universidad de Ciencias Informáticas (UCI), desempeñándose como profesor de la Facultad No. 7 vinculado a la producción en el Centro de Informática Médica (CESIM). Actualmente es parte del proyecto Sistema Integral para la Atención Primaria de Salud (SIAPS) donde se desarrolla como Administradora de la Configuración y Planificadora.

e-mail: yhgallego@uci.cu

Ing. Reynier Alejandro Rodríguez Ronquillo, Graduado de Ingeniero en Ciencias Informáticas en el año 2010. Labora en la Universidad de Ciencias Informáticas (UCI), desempeñándose como profesor de la Facultad No. 7 vinculado a la producción en el Centro de Informática Médica (CESIM). Actualmente es parte del proyecto Sistema Integral para la Atención Primaria de Salud (SIAPS) donde se desarrolla como jefe de Equipo de Desarrollo.

e-mail: rarodriguez@uci.cu

Agradecimientos

Dedicatoria

Resumen

Debido a la incorporación de las Tecnologías de la Información y la Comunicación (TIC) en el Sistema Nacional de Salud (SNS), se han comenzado a desarrollar herramientas que faciliten el trabajo del personal de salud en las diferentes instituciones. Una de estas aplicaciones 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). Este sistema no cuenta con una herramienta que facilite la gestión del familiograma que es un componente de la historia de salud familiar de vital importancia para el estudio de la familia.

La presente investigación tiene como objetivo diseñar e implementar un componente web que facilite la gestión de la información del familiograma en la APS, basado en tecnologías libres, multiplataforma y de código abierto.

Para el desarrollo de la aplicación se utiliza la Plataforma Java Enterprise Edition 5.0. Como lenguaje de programación Java, 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 para la lógica del negocio.

Con el desarrollo del sistema se espera obtener una herramienta que permita al Equipo Básico de Salud (EBS) estudiar con mayor precisión la familia y posibilite identificar factores que puedan ocasionar problemas a los miembros del núcleo familiar, contribuyendo a elevar la calidad de los servicios prestados.

Palabras Claves: Familiograma, Atención Primaria de la Salud, Equipo Básico de Salud, Historia de Salud Familia, Componente Web.

Tabla de Contenidos

Introducción.....	1
Capítulo 1. Fundamentación Teórica	6
1.1. Sistema Nacional de Salud.	6
1.2. Medicina Familiar en Cuba	7
1.3. Conceptos asociados al dominio del problema	9
1.4. Informatización de la Atención Primaria de Salud	9
1.5. Estudio del arte.....	10
1.6. Tendencias y tecnologías actuales.....	20
1.7. Herramientas.	28
Capítulo 2. Características del Sistema	29
2.1. Modelo de dominio	29
2.2. Propuesta de Sistema.....	31
Capítulo 3. Diseño del Sistema	39
3.1. Modelo de diseño	39
3.2. Fundamentación del uso de patrones.....	39
3.3. Definición de elementos de diseño	42
3.4. Diagramas de Clases del Diseño.....	44
3.5. Descripciones textuales.	46
Capítulo 4. Implementación	59
4.1. Propuesta de integración	59
4.2. Implementación	59
Conclusiones.....	68
Recomendaciones	69
Referencias Bibliográficas	70
Bibliografía	74
Glosario de Términos.....	80

Introducción

“La sociedad actual se sustenta en una serie de avances técnicos a los que se denominan Tecnologías de la Información y las Comunicaciones (TIC)” [1], las cuales han tenido un avance vertiginoso en las últimas décadas. El desarrollo de la informática ha propiciado la aparición e implementación de nuevas formas para el registro de eventos y actividades, el tratamiento de los procesos de control, la gestión y la toma de decisiones. Las nuevas tecnologías aportadas por las comunicaciones permiten compartir recursos e introducir nuevas formas de aplicaciones en la utilización de las redes informáticas, lo que ha permitido estandarizar plataformas para la difusión e intercambio acelerado de la información y el conocimiento.

“El diseño de nuevas herramientas para el tratamiento de la información en el que se integran informática y comunicaciones, permite obtener un valor añadido en los procesos de planificación, gestión y evaluación que hacen que no se pueda pensar en el desarrollo de ninguna esfera de la sociedad, si no es con la presencia de las TIC.” [2]

A pesar de las dificultades que están asociadas a la introducción de los avances tecnológicos en los países en vía de desarrollo, Cuba consciente de que las TIC aportarán mayor eficacia para el auge socio – económico de su sociedad ha tomado la decisión de introducir estas tecnologías en todos los sectores económicos y sociales. La Salud Pública es uno de los primeros sectores en reconocer la necesidad de introducir estas tecnologías que aportarán mayor eficacia y eficiencia en la prestación de los servicios, permitirá una mejor capacitación, formación y perfeccionamiento de los técnicos y profesionales, un adecuado control de los recursos y como resultado de ello una mejor calidad en la atención a los pacientes y población en general.

En Cuba se han desarrollado soluciones informáticas aisladas como parte de este proceso de informatización, una de estas aplicaciones es el Sistema Integral para la Atención Primaria de Salud (SIAPS) el cual está siendo desarrollado por el Departamento de Atención Primaria de Salud del Centro de Informática Médica (CESIM). Este sistema tiene como objetivo darle solución a los problemas existentes en la gestión y procesamiento de la información generada en la Atención Primaria de Salud (APS).

El Sistema Nacional de Salud (SNS) cubano se organiza según la complejidad de las acciones preventivas, curativas, de rehabilitación y el grado de especialización de los servicios; se estructura en los niveles de atención médica: Primaria, Secundaria y Especializada. Las áreas y unidades de salud que prestan servicios en la APS se encargan de resolver los problemas más específicos de la población de su radio de acción.

“El programa del médico y la enfermera de la familia se ratifica como el eje del actual desarrollo estratégico, orientándose el resto de las estrategias en función del mismo. Este modelo de atención es la mayor fortaleza y potencialidad que tiene el SNS, donde se comienzan a experimentar cambios; servicios que eran exclusivos de hospitales son abiertos en instituciones y unidades de la atención primaria; surgiendo así el novedoso modelo de policlínico comunitario, que se convierte en la columna vertebral del sistema de salud de Cuba.” [3]

Este modelo, también denominado Programa del Médico de la Comunidad, se caracterizó en su concepción por una voluntad política basada en el trabajo en equipo de un médico y una enfermera(o), que brindarían atención médica integral, dispensarizada y proactiva a un sector de la población, y que consideran a la familia como la unidad principal de atención, aunque también realizaron acciones sobre el individuo y la comunidad donde radica. Para el trabajo del médico y la enfermera(o) de la familia es necesaria la participación activa de la comunidad y la intersectorialidad en la gestión de salud.

Este programa establece como su principal objetivo mejorar el estado de salud de la población mediante acciones integrales dirigidas al individuo, a la familia, la comunidad y el ambiente, siempre a través de una íntima vinculación con la comunidad. Además, promueve la salud a través de cambios positivos en los conocimientos, hábitos de vida y costumbres higiénico-sanitarias de la población, prevenir la aparición de enfermedades y daños en la salud de la población. También, debe garantizar el diagnóstico precoz y la atención médica integral, alcanzar cambios positivos en el saneamiento ambiental de su radio de acción, así como de las condiciones higiénicas en que viven las familias bajo su control y desarrollar investigaciones que respondan a las necesidades de salud de la comunidad.

“La Historia de Salud Familiar (HSF) es el instrumento que el EBS utiliza para explorar las características de la familia, identificar sus factores de riesgo y de protección, detectar oportunamente problemas de salud y definir estrategias, con la participación y compromiso de la familia, dirigida a modificar factores

nocivos, fortalecer aspectos que promuevan la salud de la familia y resolver los problemas existentes con la totalidad de sus miembros.” [4]

Para cumplir con sus objetivos, la HSF debe incluir algunas dimensiones como la estructura y funcionamiento familiar. La representación gráfica de estas dimensiones se conoce tradicionalmente como familiograma, que permite evaluar el funcionamiento sistémico de la familia y es útil para la identificación de familias cuya estructura la coloca en algún riesgo de carácter biológico, psicológico o social. Conjuntamente, el familiograma permite conocer en qué etapa del ciclo de vida familiar se encuentra la familia, el tipo de familia, los eventos de salud más importantes que le han ocurrido a cada uno de sus miembros y aspectos relevantes de las interrelaciones entre ellos, con lo que constituye un resumen bastante completo de los diversos elementos que el profesional requiere para el estudio de la dinámica familiar.

El familiograma es un instrumento de vital importancia para el EBS, permite estudiar los antecedentes familiares, clínicos, patológicos, biológicos y sociales que pudieran originar problemas en algún miembro de la familia, obteniendo información de sus miembros a lo largo de varias generaciones. Con una mirada rápida al familiograma, se puede estimar la composición de la familia y determinar si es una familia nuclear, extensa o extensa compuesta; el número y la edad de cada uno de los miembros, tanto de los integrantes como de los que han fallecido; sus interrelaciones y las situaciones de enfermedad. Constituye un medio para el intercambio de información, entre los profesionales que se encargan de ofrecer atención integral a la familia, y que requieren de un medio fácil y práctico para conocer acerca del historial familiar.

La creación del familiograma constituye sin dudas, un proceso que se dificulta actualmente debido al gran volumen de información que se maneja, además, esta información es muy variada y se encuentra en diferentes documentos, lo que entorpece aún más su recolección y procesamiento. El hecho de que en Cuba este proceso sea realizado manualmente provoca que el EBS tenga que utilizar una gran parte de su tiempo en su realización.

Por otra parte, el familiograma es un documento que se actualiza periódicamente por lo que se deteriora con el tiempo y por tanto se pierde la información contenida en él. Además, es imposible acceder a dicha información cuando la familia se encuentra fuera del área del Equipo Básico de Salud a la que pertenece.

Debido a las dificultades antes mencionadas para la realización del familiograma en muchos casos este no se realiza con la calidad requerida, dificultando el trabajo del EBS, la calidad de vida del paciente y de los servicios prestados.

Por lo antes expuesto se identifica como **problema a resolver**: ¿cómo viabilizar el proceso de gestión de la información del familiograma en la Atención Primaria de Salud?

El **objetivo general** es: desarrollar un componente web que permita viabilizar la gestión de la información que se genera en los procesos relacionados con el familiograma en la Atención Primaria de Salud.

Para darle cumplimiento a los objetivos trazados se plantea como **objeto de estudio** el proceso de gestión de la información de la Historia de Salud Familiar en la Atención Primaria de Salud, y el **campo de acción** se centra en el proceso de gestión de la información del Familiograma en la Atención Primaria de Salud.

La **idea a defender** es: con el desarrollo de un sistema informático que permita gestionar la información del familiograma en la Atención Primaria de la Salud, se contará con una herramienta que facilitará el intercambio de información entre profesionales que ofrecen atención integral a la familia, se tendrá un mayor dominio y seguridad de la información de la familia.

Tareas a cumplir por los estudiantes:

- ✓ Evaluar las tendencias actuales en el mundo relacionadas con la gestión del familiograma en la Atención Primaria de Salud.
- ✓ Caracterizar los procesos en las áreas de salud relacionado con la gestión del familiograma en la Atención Primaria de Salud.
- ✓ Aplicar la arquitectura J2EE 5.0, aprobada por el CESIM para el desarrollo de las aplicaciones web en los diferentes Departamentos de Producción.
- ✓ Obtener mediante la guía de desarrollo propuesta por el proceso de mejora los artefactos correspondientes a las fases: “Modelación del Negocio”, “Requisitos”, “Análisis y Diseño” e “Implementación” de los procesos que se realizan en el área de Análisis de familiograma.

- ✓ Implementar los procesos propuestos del componente web Análisis de Familiograma para el Sistema Integral para la Atención Primaria de Salud (SIAPS).
 - Gestionar familiograma.
 - Gestionar Relaciones.

El documento está estructurado en cuatro capítulos:

CAPÍTULO I. Fundamentación Teórica: se hace referencia a la estructura del SNS, comprende un análisis de los sistemas que existen en la actualidad que se vinculan con la investigación y el estado del arte de las tecnologías y herramientas a utilizar en el desarrollo de la aplicación.

CAPÍTULO II. Características del Sistema: se describe los conceptos fundamentales relacionados con el negocio y las relaciones existentes entre ellos a través del Modelo de Dominio, el cual facilita el análisis necesario para determinar las funcionalidades del sistema a desarrollar. Estas se describen detalladamente mediante la especificación y descripción de los requerimientos funcionales y no funcionales.

CAPÍTULO III. Diseño del Sistema: se realiza el diseño del sistema a desarrollar, con el propósito de adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales obtenidos con anterioridad para facilitar la preparación, modificación y mantenimiento de los mismos. Describe los aspectos relacionados al diseño de la solución propuesta, se modelan los diagramas de clases del diseño.

CAPÍTULO IV. Implementación: describe los componentes fundamentales de todo el proceso de implementación, argumentando la justificación de la integración con otros módulos, estándares de diseño, codificación y excepciones, así como la representación general del diagrama de despliegue.

Capítulo 1. Fundamentación Teórica

En este capítulo se realizará un análisis de los elementos fundamentales que constituyen el soporte teórico del sistema que se desea diseñar e implementar y se describir los conceptos que deben ser conocidos para comprender el proceso de gestión del familiograma. Se realizará una valoración de los sistemas informáticos existentes tanto en el ámbito internacional como nacional en cuanto a sus ventajas y desventajas en torno al problema planteado. Además se definirán las técnicas, tecnologías, metodologías y herramientas de software que serán usadas durante el desarrollo del sistema.

1.1. Sistema Nacional de Salud.

Se entiende por SNS al conjunto de recursos físicos, financieros, humanos, entre otros, que dispone un país para proveer de salud a sus ciudadanos, en todos los aspectos de promoción y protección a la salud, reducción de riesgos y prevención de enfermedades, cuidados y recuperación frente a los agravantes. Según la Organización Mundial de Salud (OMS) lo define como: “Un sistema que es la suma de todas las organizaciones, instituciones y recursos cuyo objetivo principal consiste en mejorar la salud.”[5]

Para lograr una mayor organización y calidad de los servicios prestados en Cuba el SNS se estructura en tres niveles:

Atención Primaria de Salud (APS): Es la asistencia sanitaria esencial accesible a todos los individuos y familias de la comunidad a través de medios aceptables para ellos, con la plena participación de la familia. Es el núcleo del sistema de salud del país y forma parte integral del desarrollo socioeconómico general de la comunidad. Se brinda fundamentalmente a nivel de los policlínicos y/o hospitales rurales a través del programa de medicina familiar y abarca a todos los EBS.

Atención Secundaria de Salud (ASS): Su quehacer está orientado a un segmento menor de la población. Recibe para su diagnóstico y tratamiento los pacientes que no pueden ser tratados en el nivel primario. Generalmente está estrechamente relacionado con los servicios clínicos de los hospitales. Se llevan a cabo acciones de salud más complejas y especialidades médicas a mayor profundidad que en el nivel primario. Se relaciona tanto con el nivel primario para devolver los casos resueltos, como con el terciario cuando la complejidad así lo requiere. En la medida que el nivel primario va siendo más eficiente

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

este nivel debe adecuarse y diferenciarse en las especialidades o subespecialidades que sean necesarias.

Atención Terciaria de Salud (ATS): La atención terciaria, conocida habitualmente como atención hospitalaria, otorga prestaciones de mayor complejidad, tanto en atención cerrada como ambulatoria. A este nivel pertenecen los institutos y hospitales especializados.

1.2. Medicina Familiar en Cuba

El Consejo Americano de Medicina Familiar (ABFP) adoptó la siguiente definición de medicina familiar: "...es la especialidad médica que se ocupa de la salud integral del individuo y la familia. Es la especialidad más amplia que integra las ciencias biológicas, clínicas y conductuales. El ámbito de la medicina familiar no está limitado por la edad, sexo, el sistema orgánico o la entidad patológica." [6]

A principios de la década de los 80 el SNS se encontraba en una etapa de madurez y fortalecimiento, pues se obtenían resultados favorables en los indicadores de salud, resultados que son el sueño de muchos países en la actualidad. No obstante, a pesar de los resultados alcanzados hasta ese momento y los esfuerzos realizados, se advertía la necesidad de introducir cambios en el sistema de salud debido a la convergencia de los siguientes factores:

- ✓ La necesidad de incrementar cualitativamente los niveles de salud de la población.
- ✓ Los cambios producidos en el cuadro de morbilidad y mortalidad.
- ✓ La tendencia a la insatisfacción de los pacientes con la calidad del servicio prestado.
- ✓ Los cambios demográficos, pues es evidente un incremento de la población adulta especialmente los mayores de 60 años.
- ✓ La elevación del nivel educacional y cultural de la población cubana la hace más exigente y menos tolerante a la incapacidad del sistema de salud para solucionar sus problemas.

Debido a lo antes expuesto, era ineludible el fomento de una práctica médica basada en un profundo enfoque clínico, epidemiológico y social, y es así como, a principios de la década de los 80, surgen las

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

primeras ideas y análisis que desembocaron, posteriormente, en la creación del modelo cubano de medicina familiar.

La introducción de este modelo en el SNS, generó importantes transformaciones en la atención médica en el nivel primario. El nuevo tipo de médico y enfermera de la familia sustenta su actuación profesional en el análisis de la situación de salud comunitaria y la dispensarización aplicando el enfoque familiar.

“El objetivo de la medicina familiar es mejorar el estado de salud de la población mediante acciones integrales dirigidas al individuo, a la familia, la comunidad y el ambiente, siempre sirviéndose de una íntima vinculación con la comunidad. Promover la salud a través de cambios positivos en los conocimientos, hábitos de vida y costumbres higiénico-sanitarias de la población, prevenir la aparición de enfermedades y daños en la salud de la población, garantizar el diagnóstico precoz y la atención médica integral, desarrollar la rehabilitación con base comunitaria dirigida a la población discapacitada física o psíquicamente, alcanzar cambios positivos en el saneamiento ambiental de su radio de acción, así como de las condiciones higiénicas en que viven las familias bajo su control, y desarrollar investigaciones que respondan a las necesidades de salud de la comunidad.”[7]

“La salud familiar hay que verla en una relación individuo-familia-sociedad. Esto quiere decir que la salud de la familia va a depender de la interacción entre factores personales (psicológicos, biológicos, sociales), factores propios del grupo familiar (funcionalidad, estructura, economía familiar, etapa del ciclo vital, afrontamiento a las crisis) y factores sociológicos (modo de vida de la comunidad, de la sociedad).” [8]

“La medicina familiar comprende tres características básicas:” [9]

- ✓ **Continuidad:** Se comprende del hecho de integrar los aspectos biológicos, psicológicos y sociales en la atención de un paciente permitiendo así el seguimiento del proceso salud-enfermedad y facilitando establecer el tratamiento más adecuado y coordinar los recursos médicos con los de la familia del paciente y los de su entorno.
- ✓ **Anticipación:** Se basa en un enfoque de riesgo el cual contribuye a la identificación de la probabilidad de que ciertas características del paciente o del medio que lo rodea produzcan un conjunto de enfermedades. Este enfoque brinda la posibilidad de planear las estrategias más

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

adecuadas para el manejo de las enfermedades en una perspectiva integradora de medidas preventivas junto con el tratamiento médico curativo.

- ✓ **Estudio de la familia y el entorno del paciente:** El estudio de estos dos factores permite un entendimiento biopsicosocial fundamental del entorno en el que se desarrolla la vida del paciente y donde se pueden encontrar una serie de factores de tipo estructural, funcional y cultural que pueden favorecer o limitar la salud, ya sea individual o familiar.

1.3. Conceptos asociados al dominio del problema

“El estudio de la dinámica familiar es importante para entender cómo funciona cada familia y así poder planificar mejor su Atención Integral. Este describe y analiza un conjunto amplio de características tales como: la propia definición de familia; los aspectos relacionados a la composición y grado de parentesco de los miembros del grupo familiar; las relaciones entre ellos, incluidos su nivel de integración o disociación, proximidad o rechazo; sus roles, límites e interacciones; los sub-grupos constituidos al interior del grupo familiar, así como las funciones intrínsecas a la familia.” [10]

“La OMS define como familia a los miembros del hogar emparentados entre sí, hasta un grado determinado por sangre, adopción y matrimonio. El grado de parentesco utilizado para determinar los límites de la familia dependerá de los usos a los que se destinen los datos y, por lo tanto, no puede definirse con precisión en escala mundial.” [11]

1.4. Informatización de la Atención Primaria de Salud

La informatización de la APS en muchas ocasiones se percibe como un problema simple que se resuelve solamente creando redes de computadoras personales y otras Tecnologías de la Información y Comunicación entre los consultorios y policlínicos. Sin embargo, en la realidad es un problema complejo que requiere aún de múltiples proyectos de investigación científica multidisciplinaria, de desarrollo tecnológico y de mucho intercambio académico científico-tecnológico.

Actualmente el Ministerio de Salud Pública (MINSAP) ha definido a la informatización del SNS como una de sus prioridades. El centro del proceso de informatización del sector lo constituye la atención primaria. Este proceso tiene como objetivo perfeccionar la calidad asistencial ofrecida a la sociedad, facilitar las

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

funciones del personal de la salud y colaborar con la gestión administrativa, asistencial, docente y de investigación.

“El programa de informatización de la APS tiene claramente definidos tres propósitos fundamentales:” [12]

- ✓ Fortalecer la conectividad de las instituciones de salud territorial que permita una interrelación más efectiva entre ellas, así como un flujo informativo, eficiente y eficaz para la toma de decisiones a nivel local y niveles superiores.
- ✓ Fortalecer el Subsistema del Médico y la Enfermera de la Familia, para lo cual hay que centrar los esfuerzos en los policlínicos, consultorios y comunidad.
- ✓ Fortalecer el resto de los subsistemas de salud territorial, interrelacionados con el primero que le brindan la interacción necesaria.

1.5. Estudio del arte

En la actualidad la medicina familiar están orientadas a satisfacer y resolver las exigencias y necesidades sociales de salud de la población, las cuales deben ser enfocadas de manera integral considerando sus aspectos biológicos, sociales y psicológicos en diferentes escenarios de acción, garantizando adecuados niveles de calidad, calidez y equidad en la atención de la salud individual, familiar y comunitaria que permita lograr un adecuado estado de salud y calidad de vida.

“Los estudios de la salud familiar se iniciaron en 1986 a partir de que la OMS afirmara que la salud es un hecho que determina y está determinado por el funcionamiento efectivo de la familia como unidad biosocial en el contexto de una sociedad dada. De esta forma se puede apreciar que cuando la familia pierde su capacidad de funcionar, ocurren alteraciones en el proceso salud-enfermedad de todos sus miembros, sus relaciones, y por tanto, en el todo del sistema.”[13]

El estudio de la salud familiar debería ser una estrategia característica de la práctica médica familiar. La trascendencia de esta actividad se observa en la utilidad de la información obtenida para detectar cuáles son las familias más expuestas a cierto tipo de riesgo. La aplicación de instrumentos para el estudio de la familia como el APGAR familiar, el ecomapa y el familiograma entre otros ayudan al médico a observar de forma concisa, la información de la familia y sus miembros, permitiendo analizar las necesidades de

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

atención que requiere la misma. La evaluación de su funcionalidad bajo un enfoque sistémico propiciando el entendimiento de la familia desde una perspectiva dinámica.

“El **APGAR** familiar es un cuestionario de cinco preguntas, que buscan evidenciar el estado funcional de la familia, funcionando como una escala en la cual el entrevistado coloca su opinión al respecto del funcionamiento de la familia para algunos temas clave considerados marcadores de las principales funciones de la familia” [14]. El acrónimo APGAR hace referencia a los cinco componentes de la función familiar: adaptabilidad (adaptability), cooperación (partnership), desarrollo (growth), afectividad (affection) y capacidad resolutoria (resolve).

Con este instrumento los integrantes del EBS pueden conseguir una primera aproximación para la identificación de aquellas familias con problemas de conflictos o disfunciones familiares, con el propósito de darles seguimiento y orientación para una remisión oportuna y adecuada en los casos que sea necesario. El APGAR familiar evalúa cinco funciones básicas de la familia, estas se muestran en la tabla 1.1.

Tabla 1.1 Funciones de la Familia evaluadas por el APGAR Familiar.

Funciones	Contenido
Adaptación	Evalúa la forma en que los miembros de la familia, utilizan los recursos intra y extrafamiliares en los momentos de grandes necesidades y periodos de crisis, para resolver sus problemas y adaptarse a las nuevas situaciones.
Participación	Mide la satisfacción de cada miembro de la familia, en relación con el grado de comunicación existente entre ellos en asuntos de interés común y en la búsqueda de estrategias y soluciones para sus problemas.
Ganancia	Hace referencia a la forma en que los miembros de la familia encuentran satisfacción en la medida en que su grupo familiar acepte y apoye las acciones que emprendan

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

	para impulsar y fortalecer su crecimiento personal.
Afecto	Evalúa la satisfacción de cada miembro de la familia en relación a la respuesta ante expresiones de amor, afecto, pena o rabia, dentro del grupo familiar.
Recursos	Evalúa la satisfacción de cada miembro de la familia, en relación a la forma en que cada uno de ellos se compromete a dedicar tiempo, espacio y dinero a los demás. Los recursos son elementos esenciales en la dinámica familiar.

El **ecomapa** dibuja al individuo y a la familia en su espacio de vida, representando en forma dinámica el sistema ecológico que rodea al familiar: las conexiones importantes, los límites intrasistémicos, los conflictos, los recursos, las carencias, además de la naturaleza e intensidad de las relaciones familiares con los sistemas externos, o sea en sus relaciones con el suprasistema, la sociedad, y/o con otros sistemas.

Para su graficación se parte con el famiolograma o genograma en su forma más simple en un círculo o rectángulo central grande. A su alrededor se dibujan tantos círculos o rectángulos como interacciones con sistemas externos tenga la familia en su totalidad o cada uno de sus componentes: salud, educación, deporte, recreación, cultura, religión, política, relaciones afectivas y /o sociales, laborales, familia extensa o de origen.

Este instrumento brinda las siguientes utilidades:

- ✓ Señala las interacciones de la familia y de cada uno de sus miembros con los sistemas que lo rodean en forma gráfica y resumida.
- ✓ Permite identificar posibles motivos de estrés o de conflicto.
- ✓ Sugiere recursos para movilizar en momentos de crisis.
- ✓ Indica si algunas personas tienen buenas relaciones con elementos externos a la familia.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- ✓ Indica si la familia en conjunto o alguno de sus miembros están aislados de aquellas fuentes de recreación, educación, salud, social, etc.

El **genograma** es un instrumento utilizado antiguamente para el conocimiento de su familia, quiénes eran los antepasados y sus historias. Más tarde este instrumento se generaliza por la necesidad de recoger información capaz de sintetizar los antecedentes familiares, el nivel sociocultural y la información laboral de los pacientes, como factores condicionantes de sus problemas de salud. Es un instrumento clínico práctico que facilita el abordaje sistémico en la atención del paciente, proporcionando al profesional y al paciente una muestra gráfica de la familia, incluyendo los patrones de enfermedad de la familia y los problemas biomédicos y psicosociales. Es por ello que también se le da el nombre de familiograma.

La creación del familiograma implica tres pasos:

1. Trazado de la estructura familiar.
2. Registro de la información de la familia.
3. Señalamiento de las relaciones Familiares

Trazado de la estructura familiar.

El trazado de la estructura se realiza expresando a través de símbolos las relaciones biológicas y legales entre los miembros de la familia. Círculos y cuadrados representan a las personas y líneas describen las relaciones.



Figura 1.1.

El Varón siempre se coloca a la izquierda. Las relaciones de parejas legalmente establecidas se trazan con líneas continuas (Figura 1.1) y las relaciones de parejas no formalizadas legalmente (Concubinato) se expresan en líneas discontinuas incluyendo el equivalente de pareja (Figura 1.2).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

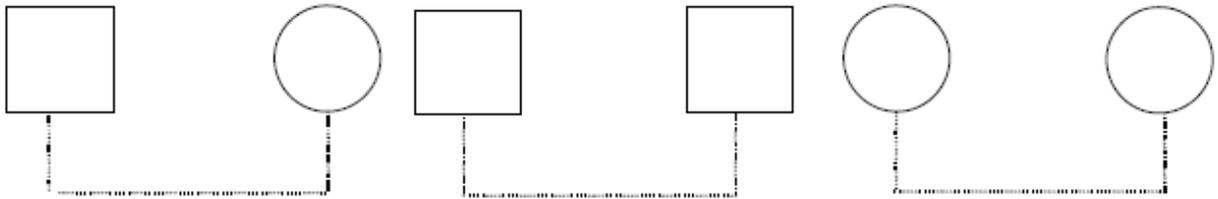


Figura 1.2.

Por encima de la barra se puede colocar la fecha de matrimonio (Figura 1.3).

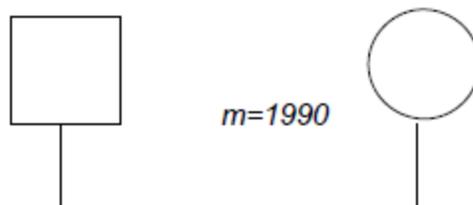


Figura 1.3.

En la línea horizontal se representa el divorcio con dos barras inclinada representa ruptura y una sola barra simboliza separación. La fecha de separación o divorcio se coloca encima de la barra inclinada (Figura 1.4).



Figura 1.4.

Si es necesario, por la repercusión que tenga en la familia actual, reflejar las uniones anteriores también pueden representarse, poniendo ausente la pareja anterior lo cual se realiza por el trazo de punto o discontinuo. En esta caso es importante significar que con vista de mantener en principio de que el gráfico

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

de la estructura familiar la figura masculina se refleja a la izquierda de la femenina y teniendo en cuenta la presencia de la familia de segunda nupcias y más en la población cubana se recomienda:

1. Cuando se estudia una familia en la que el hombre tenga segunda nupcias el familiograma quedara reflejado el matrimonio o relación disuelta a la izquierda de la figura masculina y la relación actual a la derecha (Figura 1.5).

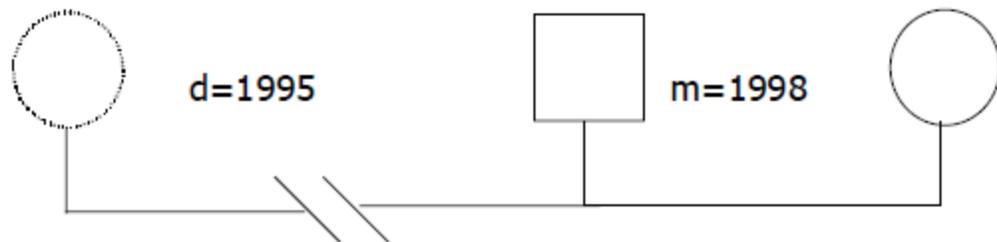


Figura 1.5.

2. Cuando se estudia una familia en la que la mujer tenga segunda nupcias el familiograma quedara reflejado la relación actual a la izquierda de la figura femenina y la relación disuelta a la derecha de la misma (Figura 1.6).

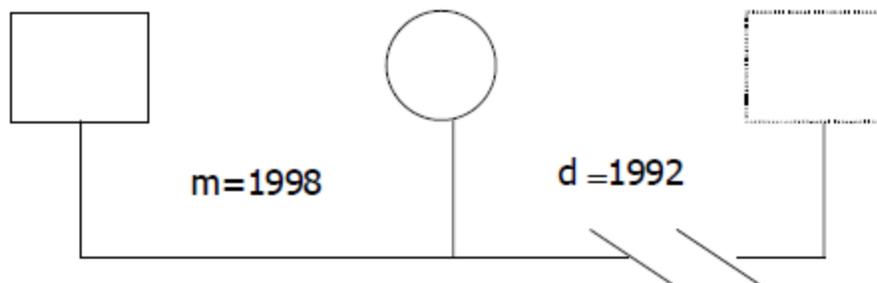


Figura 1.6.

Los hijos penden con líneas verticales de la línea de unión horizontal de sus padres. Se sitúan de izquierda a derecha por orden de nacimiento sin tener en cuenta el sexo (Figura 1.7).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

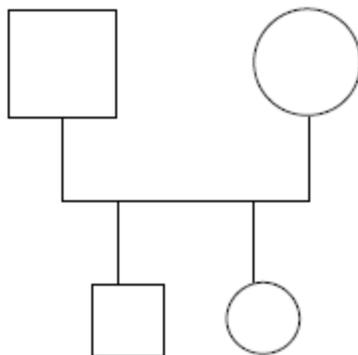


Figura 1.7

El miembro ausente se señala con líneas de puntos o discontinua, ya sea cuadrado o círculo según corresponda y según sea el caso y se coloca donde le corresponda su lugar en la representación de la estructura familiar (Figura 1.8).



Figura 1.8.

Para representar miembros fallecidos se coloca una **X** dentro del círculo o del cuadrado según corresponda (Figura 1.9).



Figura 1.9

Registro de la información de la familia.

Consiste en ponerle al gráfico la información demográfica y de salud de los miembros de la familia. La información demográfica se escribe dentro del cuadrado o del círculo de la figura, así puede reflejarse la

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

fecha de nacimiento y la de fallecimiento también, solo si fuera necesario para el análisis de la situación de salud de la familia (Figura 1.10).

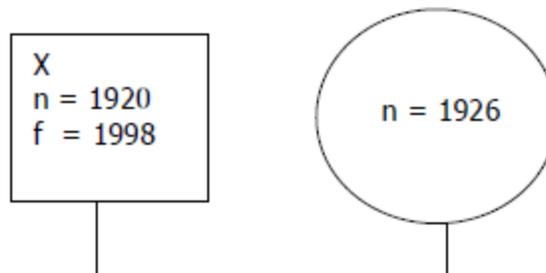


Figura 1.10

La información de salud de los miembros de la familia suele escribirse por fuera de los correspondientes símbolos siempre que sean necesarios (Figura 1.11).

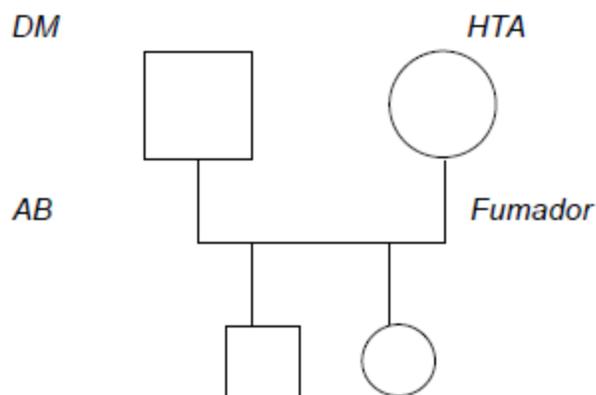


Figura 1.11.

Señalamiento de las relaciones familiares

Este tipo de información refleja la relación entre los miembros, puede ser cambiante o perdurar por largo período de tiempo, significa el tipo de relación que establece cada miembro con todos los integrantes de la familia. Si la relación se considera normal no se traza ninguna línea adicional.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

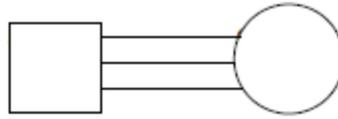


Figura 1.12 Relación Fusionada.



Figura 1.13 Relación de Conflicto.



Figura 1.14 Relación Distante.



Figura 1.15 Ruptura de la Relación.

La informática, las comunicaciones y las nuevas tecnologías de la información han tenido un crecimiento sin precedente en los últimos años. Es de vital importancia para todas las esferas de la sociedad adoptar el uso de estas herramientas para lograr un mayor desarrollo. El sector de la Salud no está ajeno a este proceso de incorporación de las TIC con el objetivo de lograr una mejor atención médica a la población. En el mundo existen múltiples empresas que desarrollan sistemas informáticos para gestionar y procesar la información procedente de los diferentes niveles de la atención médica. A continuación se realizará un análisis de algunos de los sistemas existentes tanto en el ámbito internacional como nacional.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

“El software **Retrato de Salud de mi Familia** fue elaborada en colaboración entre la Oficina del Cirujano General de Estados Unidos y el Instituto Nacional de Investigación del Genoma Humano. La primera versión de Retrato de Salud de mi Familia salió al mercado, como un software descargable para computadora. Este sistema solamente podía usarse en computadoras que poseían sistema operativo Windows, siendo esto su principal desventaja.”[15]

En la actualidad existe una nueva versión de este software basada en tecnología web. Este sistema recopila información del individuo y su familiares más cercanos con el objetivo de crear un familiograma, también posee otra presentación de los datos en forma de tabla. Brinda la posibilidad de exportar el trabajo en formato HTML y PDF. No brinda la posibilidad de guarda la información que se procesa en el mismo.

Uno de sus principales inconvenientes es que permite darle seguimiento solamente a un pequeño grupo de las enfermedades que puede padecer el paciente además esta herramienta no permite recoger información de aquellas personas que son parte del núcleo familiar pero no posee lazos consanguíneos con los miembros de la familia. Otra desventaja que presenta dicho software es que no permite representar las relaciones interpersonales entre los miembros del núcleo familiar, datos que son de vital importancia para evaluar el funcionamiento sistémico de la familia.

“**GenoPro** es un programa genealógico que ofrece una solución práctica a la creación de árboles familiares y genogramas o familiogramas. Este sistema puede mostrar una representación gráfica completa del árbol genealógico de una persona. GenoPro permite al usuario la modificación manual de la disposición de los diferentes componentes gráficos, resaltar con colores los comportamientos presentes en la familia tales como etnia, cultura, ciudadanía, nivel de la educación, religión, afiliaciones políticas, y enfermedades.”[16]

Este sistema brinda la posibilidad de dividir un árbol cuando es demasiado grande en subestructuras más pequeñas facilitando el análisis del mismo. Incluye hipervínculos para navegar entre los elementos que conformar el gráfico. Presenta un grupo de símbolos para representación de las relaciones interpersonales entre los miembros de la familia. Posee capacidad para exportar el resultado en formato HTML. Una de sus principales desventajas es que ha sido diseñado únicamente para funcionar en plataforma Windows además de que es privativo. Este software solamente permite darle seguimiento a un pequeño grupo de

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

las enfermedades que puede presentar un paciente, desechando información valiosa para un posterior análisis de la familia.

Como parte del proceso de informatización del SNS en Cuba se han desarrollado algunos productos informáticos aislados, unos de estos productos es “**El Sistema Informático para la Gestión Médica en la APS** que es uno de los software más importantes desarrollados en el ámbito nacional. La principal función de este sistema es informatizar la información de un centro de salud en la Atención Primaria, ofreciendo al Equipo de Salud y a los Directivos del Centro, la información necesaria, oportuna y útil para la toma de decisiones que ayude a elevar el nivel de salud de la población bajo su atención. Además brinda información para la elaboración y actualización periódica del análisis de la Situación de Salud, que permite evaluar la eficiencia de los principales servicios y actividades médicas realizadas.” [17]

Este sistema fue desarrollado por el Centro de Desarrollo Informático de la Salud (CEDISAP). El lenguaje de programación utilizado para su desarrollo fue Borland Delphi y para la gestión de las bases de datos SQL Server. Uno de sus principales inconvenientes es que no cuenta con una herramienta capaz de brindar una vista general de la estructura de la familia que permita al médico evaluar el funcionamiento de la familia, identificar riesgo causados por la estructura familiar y otros elementos que el profesional requiere para el estudio de la dinámica familiar.

Valoración de las soluciones existentes

Las soluciones existentes a nivel mundial para gestión del familiograma no cumplen con las normas establecidas para la informatización de la Atención Primaria en Cuba. Las soluciones existentes en el ámbito nacional no cuentan con un componente que le facilite al Equipo Básico de Trabajo (EBT) el análisis de la estructura familiar así como otros elementos de vital importancia para la toma de decisiones y el intercambio de información entre los profesionales de la salud. Por todo lo antes expuesto surge la necesidad de agregarle un componente web al SIAPS para facilitar al personal médico la gestión del familiograma.

1.6. Tendencias y tecnologías actuales.

Con la rápida evolución que tiene en la actualidad las tecnologías de la informática, es de vital importancia a la hora de desarrollar un software realizar un análisis exhaustivo de las tendencias tecnológicas

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

actuales. Teniendo en cuenta las características del software que se desarrollará y el entorno donde será desplegado, se definieron las tecnologías que serán usadas para lograr un software con calidad.

1.6.1 Sistemas Distribuidos. Modelo Cliente-Servidor

“Un sistema distribuido es una colección de ordenadores autónomos enlazados por una red de ordenadores y soportados por un software que hace que la colección actúe como un servicio integrado.” [18]

Características claves de los sistemas distribuidos:

- ✓ **Concurrencia:** Es la capacidad de acceder de forma paralela a los recursos por diferentes componentes.
- ✓ **Comportamiento ante fallos del sistema:** Cuando un componente o parte de un componente falle el resto del sistema debe continuar funcionando como si no pasara nada y debe ser capaz de recuperar el estado de los datos permanentes ante que se produjera el fallo.
- ✓ **Escalabilidad:** Se dice que un sistema es escalable si se mantiene efectivo cuando hay un incremento significativo en el número de recursos y el número de usuarios.
- ✓ **Seguridad:** Los sistemas distribuidos contienen información valiosa para sus usuarios. Por lo tanto, su seguridad es muy importante. Se deben usar técnicas de cifrado para proporcionar una protección adecuada de los recursos compartidos y mantener secreta, mientras se transmite por la red, la información que se crea oportuna.

La arquitectura Cliente-Servidor es la más conocida y más ampliamente adoptada en la actualidad en los entornos distribuidos. Esta arquitectura permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma.

“Entre las principales ventajas de la arquitectura Cliente-Servidor están:” [19]

- ✓ Uno de los aspectos que más ha promovido el uso de la arquitectura Cliente-Servidor, es la existencia de plataformas de hardware cada vez más baratas.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- ✓ La arquitectura Cliente-Servidor facilita la integración entre sistemas diferentes.
- ✓ Una ventaja adicional del uso de la arquitectura Cliente-Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes.
- ✓ Facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- ✓ La arquitectura Cliente-Servidor proporciona a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global.

1.6.2 Frameworks, librerías y componentes.

Un framework es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le añade las últimas piezas para construir una aplicación concreta. Estos permiten acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo de software.

Java Server Faces (JSF)

“La tecnología Java Server Faces 1.2 (JSF) constituye un marco de trabajo de interfaces de usuario del lado de servidor para aplicaciones web basadas en tecnología Java y en el patrón MVC. Una de las características que lo hace más atractivo es que, brinda un modelo basado en componentes y dirigidos por eventos para el desarrollo de aplicaciones web. Otra característica muy importante de JSF es que, a pesar de que HTML es un lenguaje de enmarcado por default, no está limitado a este ni a ningún otro, pues tiene la capacidad de utilizar diferentes renderers para los componentes de la Interfaz Gráfica de Usuario (GUI) y obtener así diferentes salidas para mandar al cliente. Así mismo, JSF es suficientemente flexible para soportar diferentes tecnologías de presentación.” [20]

RichFaces

“RichFaces 3.3.1 es un framework de código abierto que añade capacidad Ajax dentro de aplicaciones JSF existentes sin recurrir a JavaScript. RichFaces incluye ciclo de vida, validaciones, conversores y la

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

gestión de recursos estáticos y dinámicos. Los componentes de RichFaces están contruidos con soporte Ajax y un alto grado de personalización que puede ser fácilmente incorporado dentro de las aplicaciones JSF.” [21]

RichFaces 3.3.1 permite obtener mayores beneficios JSF al trabajar con Ajax, agregar funcionalidades Ajax en las páginas sin necesidad de escribir nada de código JavaScript, crear rápidamente vistas complejas basándose en la caja de componentes, escribir componentes propios con funciones soportadas por Ajax. Conjuntamente, proporciona un avanzado soporte a la gestión de diferentes recursos: imágenes, código JavaScript y hojas de estilo CSS. Los componentes de la interfaz de usuario de RichFaces vienen preparados para su uso fuera del paquete, así los desarrolladores ahorrarán tiempo y podrán disponer de las ventajas mencionadas para la creación de aplicaciones web.

Facelets

Facelets 1.1 es un framework para plantillas centrado en la tecnología JSF que facilita la gestión de la estructura y el estilo de las páginas, lo cual tiene beneficios en cuanto a reutilización de código en la capa de presentación y por consecuencia facilita el mantenimiento de esta capa.

“Este framework incluye muchas características siendo las más importantes:” [22]

- ✓ Tiempo de desarrollo cero de los tags para UIComponents.
- ✓ Facilidad en la creación del templating para los componentes y páginas.
- ✓ Habilidad de separar los UIComponents en diferentes archivos.
- ✓ Un buen sistema de reporte de errores.
- ✓ Soporte completo a EL (Expression Language).
- ✓ Validación de EL en tiempo de construcción.
- ✓ No es necesaria configuración XML.
- ✓ Trabaja con cualquier RenderKit.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Al usar Facelets la plantillas contendrán la estructura de la páginas y las definiciones de estilo, en esta plantilla se definen zonas o áreas (como el header, menú, espacios comunes) y el motor de facelets se encargará de llenar esas áreas con el contenido apropiado, así por ejemplo no tendrá el header en todas las páginas y si después de un tiempo algo tiene que cambiar se hará solo en la plantilla en lugar de hacerlo en cada página.

Ajax4Jsf

“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 este framework 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 automáticas al servidor, control de cualquier evento de usuario, etc. En definitiva Ajax4jsf permite dotar a las aplicaciones JSF de contenido mucho más profesional con muy poco esfuerzo.” [23]

JBoss Seam

JBoss Seam 2.2.1 es un framework para la creación de aplicaciones web que integra y unifica varias tecnologías como AJAX, Enterprise Java Beans (EJB3), Java Server Faces (JSF), Java Portlets and Business Process Management (BPM), Hibernate, y mucho más de la plataforma Java EE 5.0, pudiendo trabajar con todos ellos siguiendo el mismo modelo de programación.

“Ha sido diseñado intentando simplificar al máximo el desarrollo de aplicaciones, basando el diseño en Plain Old Java Objects (POJOs) con anotaciones. Estos componentes se usan desde la capa de persistencia hasta la de presentación, poniendo todas las capas en comunicación directa.” [24]

Hibernate.

“Hibernate 3.3 es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada se pueden generar BBDD en cualquiera de los entornos soportados: Oracle, DB2, MySql entre otros. Y lo más importante de todo, es de código abierto, lo que supone, entre otras cosas, que no se tiene que pagar nada por adquirirlo.” [25]

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

“Hibernate 3.3 en su funcionamiento genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todas las bases de datos con un ligero incremento en el tiempo de ejecución. A la hora de realizar una optimización en las consultas, el hecho de generar el código en caliente hace que sea más fácil modificarlo y ajustarlo. Este está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.” [26]

1.6.3 Lenguaje de programación

El lenguaje de programación seleccionado para el desarrollo de este proyecto es Java 1.6. Es un lenguaje orientado a objetos de propósito general, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores. Aunque Java comenzará a ser conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto. En el diseño de Java se prestó especial atención a la seguridad. Existen varios niveles de seguridad en Java, desde el ámbito del programador, hasta el ámbito de la ejecución en la máquina virtual. Con respecto al programador, Java realiza comprobación estricta de tipos durante la compilación, evitando con ello problemas tales como el desbordamiento de la pila.

“Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que los mismos programas escritos para Linux pueden ser ejecutados en Windows sin ningún problema.” [27]

La programación en Java, permite el desarrollo de aplicaciones bajo el esquema de Cliente-Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar.

1.6.4 Servidor de Aplicaciones

JBoss Application Server 4.2.2 es un Servidor de Aplicaciones Java EE de Software Libre implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

soporte. Combinando una arquitectura orientada a servicios con una licencia de código abierto, JBoss Application Server puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia. Por este motivo es la plataforma más popular de middleware para desarrolladores, vendedores independientes de software y, también, para grandes empresas. Incluye servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en tecnologías web. También soporta Enterprise Java Beans, arquitectura componente del lado del servidor para la plataforma Java, que permite realizar la administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portables y persistencia de datos.

1.6.5 Sistema Gestor de Base de Datos (SGBD)

“Un Sistema Gestor de Base de Datos (SGBD) o DBMS (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.” [28]

“PostgreSQL 8.4 es un SGBD objeto-relacional, distribuido bajo licencia Berkeley Software Distribution (BSD) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.” [29]

“PostgreSQL 8.4 utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Algunas de las características más representativas son: estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.” [30]

1.6.6 Guía para el proceso de desarrollo de software.

CMMI (Capability Maturity Model Integration) es un modelo que permiten obtener un diagnóstico preciso de la madurez de los procesos de desarrollo, mantenimiento y operación de sistemas de software. Describe las tareas que se tienen que llevar a cabo para mejorar esos procesos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Entre sus características esenciales este modelo está compuesto por cinco niveles de madurez, dentro de los cuales cada uno tiene un conjunto de áreas de procesos asociadas a él, siendo todos estos elementos parte fundamental de su estructura básica. Este modelo tiene dos formas de representarse: continua y escalonada, donde cada empresa u organización puede optar por cualquiera de ellas en correspondencia con sus características.

La Universidad de las Ciencias Informáticas se encuentra desarrollando un Proceso de Mejora (PM) con el objetivo de obtener la certificación internacional del nivel 2 de CMMI. Este programa ofrece una guía para el proceso de desarrollo de software. Las áreas de procesos que lo forman son:

- ✓ Administración de Requisitos (REQM).
- ✓ Planeación del Proyecto (PP).
- ✓ Monitoreo y Control del Proyecto (PMC).
- ✓ Medición y Análisis (MA).
- ✓ Aseguramiento de la Calidad de Procesos y productos (PPQA).
- ✓ Administración de la Configuración (CM).
- ✓ Administración de Acuerdos con Proveedores (SAM).

1.6.7 Lenguaje Unificado de Modelado (UML).

“El Lenguaje de Modelamiento Unificado 2.1 (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.” [31]

1.6.8 Notación Utilizada para Modelar los Procesos del Negocio.

La Notación para Modelar los Procesos del Negocio (BPMN) es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Su objetivo fundamental es coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.7. Herramientas.

En el desarrollo de sistemas de software es de vital importancia el uso de Herramientas CASE para el sustento de la metodología usada, pues estas permiten mejorar la productividad en el desarrollo y mantenimiento del software reduciendo el costo de la misma en términos de tiempo y dinero. La herramienta seleccionada para el sustento de la metodología fue Visual Paradigm suite 6.4, esta Herramienta CASE da soporte al modelado visual de UML 2.1 y BPMN.

Para el desarrollo de la aplicación web, se seleccionó el Entorno de desarrollo integrado (IDE) Eclipse SDK en su versión 3.1, con la colección de plug-ins de JBoos Tools, necesarios para utilizar los frameworks de desarrollo que se proponen. Además, se utiliza el pgAdmin III en su versión 1.10.5 para la administración de la base de datos.

En este capítulo se profundizó en los conceptos relacionados con SNS, APS y Medicina Familiar. Se realizó un análisis del proceso de informatización de la APS que se lleva a cabo en Cuba para mejorar la atención médica a la población. Se analizaron las herramientas existentes relacionadas al problema y las tecnologías que serán usadas para darle solución al mismo. Se fundamentaron el lenguaje a utilizar, las herramientas usadas para el desarrollo, la arquitectura, el SGBD, las metodologías, exponiendo sus características y ventajas que aportan al desarrollo de la aplicación.

Capítulo 2. Características del Sistema

La descripción de las características del sistema es una de las tareas más difíciles e importantes en el proceso de desarrollo de un software, 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. En este capítulo se explican los conceptos fundamentales abordados en el desarrollo del Modelo de Dominio 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.

2.1. Modelo de dominio

En el proceso de desarrollo de un software es de vital importancia realizar un profundo análisis del problema que se desea automatizar con el objetivo de lograr un alto nivel de entendimiento de este. Para realizar este análisis con la mayor calidad posible los especialistas en informática, tienen que interactuar con personal que, aunque perfectamente capacitado en su campo de acción, raramente dominan terminologías relacionadas con la informática, por lo que los informáticos deben ser capaces de entrar en su entorno e identificar los problemas para luego ofrecer una solución.

El modelamiento del proceso de gestión del familiograma se torna un poco complejo debido a que no está bien definido y sus fronteras no están bien delimitadas por los que se decidió en este caso realizar el modelado del dominio, que es una representación de las clases conceptuales del mundo real, no de componentes de software, no se trata de un conjunto de diagramas que describen clases u objetos de software con responsabilidades, sino más bien representa las clases conceptuales u objetos del mundo real en un dominio de interés. Este se debe concebir como un diccionario visual de abstracciones con vocabulario e información estrictamente del dominio.

2.1.1 Conceptos fundamentales

Paciente: Es la persona que recibe los servicios de un médico u otro profesional de la salud, sometiéndose a un examen, a un tratamiento o a una intervención.

Relaciones: Especifica los vínculos afectivos y legales existente entre los miembros de la familia.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Relaciones legales: Describe los vínculos de carácter legal que existen entre dos personas cualesquiera.

Relaciones interpersonales: Define los vínculos afectivos que establece cada miembro del núcleo familiar con los restantes miembros de la familia.

Símbolo: “Representación sensorialmente perceptible de una realidad, en virtud de rasgos que se asocian con esta por una convención socialmente aceptada.” [32]

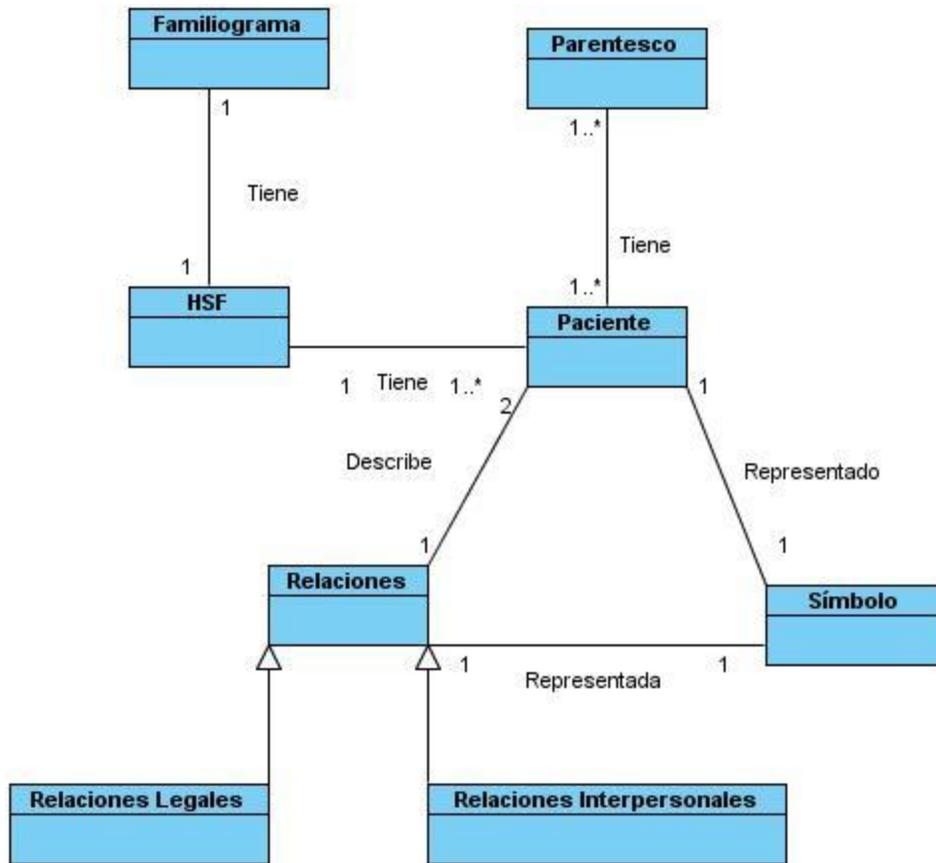
Parentesco: “Es el vínculo por consanguinidad, adopción, matrimonio, afinidad u otra relación estable de afectividad. Se trata, por tanto, de vínculos que pueden ser biológicos o no y que se organizan de acuerdo a líneas que permiten medir diversos grados de parentesco.” [33]

Familiograma: Es la representación esquemática de la familia, que provee información de los miembros de esta en cuanto a su estructura y relaciones.

HSF: Es un documento oficial que reúne como un todo los datos referentes a la familia y proporciona al médico la información necesaria para la planificación de las acciones a realizar con el objetivo de mantener un control del estado de salud de su población.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.1.2 Diagrama del modelo de dominio



2.2. Propuesta de Sistema

2.2.1 Especificación de Requerimientos de Software

Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción de un software. Durante muchos años muchas aplicaciones han fallado (no se culminaron o no se usaron) porque existieron incongruencias entre lo que el usuario quería, lo que realmente necesitaba, lo que interpretaba cada miembro del equipo de proyecto y lo que realmente se obtiene. Aquí radica la importancia que en los últimos años se le ha dado a la identificación de los requerimientos como parte del proceso de desarrollo del software.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

“IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.” [34] Los requerimientos se clasifican en funcionales y no funcionales.

2.2.2 Requerimientos Funcionales

“Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera que este debe reaccionar a entrada particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer.” [35] El estudio realizado en el negocio del problema arrojó los siguientes requerimientos funcionales:

RF 1. Gestionar familiograma.

RF 1.1 Representar familiograma: Muestra una representación gráfica de la estructura de la familia, expresando a través de símbolos los miembros de esta así como las relaciones biológicas, legales, interpersonales e información relacionada con estos.

RF 1.2 Generar familiograma: Modificar la estructura de la familia o las relaciones biológicas, legales e interpersonales existentes entre los miembros del núcleo familiar.

RF 2. Gestionar relaciones.

RF 2.1. Agregar relaciones: Registrar las relaciones existentes entre los miembros de la familia.

RF 2.2. Buscar relaciones: Obtener las relaciones existentes entre los miembros de la familia según la opción de búsqueda seleccionada (criterio de búsqueda o listar todos).

RF 2.3. Eliminar relaciones: Permite eliminar las relaciones existentes entre los miembros de la familia (cambia de estado pues no se elimina físicamente de la Base de Datos).

RF 2.4. Modificar relaciones: Modificar el tipo de relación existente entre los miembros de la familia.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RF 3. Gestionar datos de relaciones legales.

RF 3.1. Agregar relaciones legales: Registrar los datos de un nuevo tipo de relación legal.

RF 3.2. Buscar relaciones legales: Obtener los datos de una relación legal en dependencia de la opción de búsqueda seleccionada (criterios de búsqueda o listar todos).

RF 3.3. Eliminar relaciones legales: Permite eliminar una relación legal (cambia de estado pues no se elimina físicamente de la base de datos).

RF 3.4. Modificar relaciones legales: Modificar los datos de una relación legal.

RF 4. Gestionar datos de relaciones interpersonales.

RF 4.1. Agregar relación interpersonal: Registrar los datos de una nueva relación interpersonal.

RF 4.2. Buscar relación interpersonal: Obtener los datos de una relación interpersonal en dependencia de la opción de búsqueda seleccionada (criterios de búsqueda o listar todos).

RF 4.3. Eliminar relación interpersonal: Permite eliminar una relación interpersonal (cambia de estado pues no se elimina físicamente de la base de datos).

RF 4.4. Modificar relación interpersonal: Modificar los datos de una relación interpersonal.

2.2.3 Requerimientos no Funcionales

Los requisitos no funcionales 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 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, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Requerimientos de Usabilidad

RNF 1: El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

- ✓ Usuarios normales: 20 días
- ✓ Usuarios avanzados: 30 días

Requerimientos de Fiabilidad

RNF 2: En los servidores de los policlínicos y en el Centro de Datos Nacional se garantizará una arquitectura de máxima disponibilidad, tanto de servidores de aplicación como de base de datos.

RNF 3: Se garantizarán políticas de respaldo a toda la información, evitando pérdidas en caso de desastres ajenos al sistema.

RNF 4: Las informaciones médicas relacionadas con los pacientes y que vayan a ser intercambiadas con otros policlínicos por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas.

RNF 5: Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan.

RNF 6 Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.

RNF 7 Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.

RNF 8: Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.

RNF 9: Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude.

RNF 10: Los mecanismos serán capaces de informar al personal autorizado sobre posibles irregularidades que den indicios sobre la introducción de información falseada.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RNF 11: 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.

RNF 12: Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista.

RNF 13: El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

RNF 14: Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo.

RNF 15: Se permitirá realizar réplica de la base de datos entre las diferentes entidades y estas a su vez con el Centro de Datos, esta réplica se podrá hacer de forma manual y automatizada a través de la red.

RNF 16: Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema, para esto debe existir un registro de trazas que almacene todas las transacciones realizadas en el sistema, indicando para cada caso como mínimo: usuario que realizó la transacción, tipo de operación que se realizó, fecha y hora en que se realizó la operación e información contenida en el registro modificado.

Requerimientos de Eficiencia.

RNF 17: Se deberá usar siempre que sea posible el patrón Singleton, destruir referencias que ya no estén siendo usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

RNF 18: El sistema minimizará el volumen de datos en las peticiones y optimizará el uso de recursos críticos como la memoria.

RNF 19: Se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.

RNF 20: El sistema adoptará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Requerimientos de Soporte

RNF 21: Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación.

RNF 22: Las notificaciones de las deficiencias detectadas en la aplicación desplegada deberán realizarse por escrito.

RNF 23: Una vez notificada por la entidad médica, la deficiencia detectada en la aplicación desplegada, el equipo de desarrollo deberá solucionarla en un período de 7 días.

RNF 24: La capacitación y entrenamiento del profesional de salud para el uso del sistema, se realizará en un período de 6 meses.

Restricciones de diseño

RNF 25: La capa de presentación contendrá todas las vistas y la lógica de la presentación.

RNF 26: El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio.

RNF 27: La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario.

RNF 28: La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas.

RNF 29: El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

RNF 30: La nomenclatura de los bases de datos, tablas, campos de las tablas, esquemas, procedimientos almacenados o funciones, vistas, triggers y tipos de datos se elaborarán siguiendo los estándares definidos en el proyecto.

RNF 31: La codificación se realizará siguiendo el estándar definido en el proyecto.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RNF 32: Las interfaces se realizarán siguiendo el estándar definido en el proyecto.

Requisitos para la documentación de usuarios en línea y ayuda del sistema.

RNF 33: Se posibilitará el uso de ayudas y manuales dinámicos en línea sobre el funcionamiento del sistema.

Requerimientos de Interfaz

RNF 34: Las ventanas del sistema contendrán los datos claros y bien estructurados para facilitar la interpretación correcta de la información.

RNF 35: La interfaz contará con teclas de función y menús desplegable que faciliten y aceleren su utilización.

RNF 36: La entrada de datos incorrecta será detectada claramente e informada al usuario.

RNF 37: Todos los textos y mensajes en pantalla aparecerán en el idioma inherente al país o región donde se despliegue el sistema.

RNF 38: Se interactuará con el sistema alas RIS para realizar solicitudes y obtener resultados de estudios radiológicos e imagenológicos a través de una interfaz definida en el proyecto.

RNF 39: Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven).

RNF 40: El sistema usará el formato estándar WSDL (Web Services Description Language) para la descripción de los servicios web.

RNF 41: El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos.

RNF 42: El sistema utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Requerimientos de hardware

RNF 43: Las estaciones de trabajo debe tener como mínimo 256 Mb de memoria RAM y un microprocesador de 2.0 Hz.

RNF 44: La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad, residencia de la información y aplicaciones bajo esquemas seguros y confiables.

RNF 45: Los servidores de base de datos deberán tener: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.

RNF 46: Los servidores de aplicaciones deberán tener: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.

RNF 47: Los servidores de Intercambio deberán tener: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

Requerimientos de software

RNF 48: El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL).

RNF 49: El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2.

En este capítulo se realizó el modelo de dominio y se explicaron los conceptos fundamentales que se abordan en la elaboración de este. Se describieron las capacidades y cualidades que tendrá el sistema a través de los requisitos funcionales y no funcionales obteniendo una perspectiva del sistema que se desea desarrollar. Se establecieron las bases para el desarrollo exitoso de las posteriores disciplinas.

Capítulo 3. Diseño del Sistema

Para desarrollar un sistema de software con calidad se debe prestar cuidadosa atención al diseño, que tiene como objetivo modelar el sistema y encontrar su forma para que soporte todos los requisitos, tanto funcionales como no funcionales. Además permite al equipo de desarrollo de software evaluar la calidad del software antes de implementarlo.

En este capítulo se describirá el diseño del sistema propuesto, se construyen los diagramas de clases del diseño y para una mayor comprensión de este se realiza la descripción de sus clases. Se explica la arquitectura utilizada y los principales patrones de diseño usados.

3.1. Modelo de diseño

“El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con los lenguajes de programación, tienen impacto en el sistema a considerar. Sirve de abstracción para actividades de implementación y es utilizada como una entrada fundamental de la actividad de implementación.” [36]

3.2. Fundamentación del uso de patrones

En términos generales se define un patrón como un modelo que se sigue para la realización de una actividad determinada. Un patrón de diseño es una solución a un problema de diseño, aceptada como correcta, a la que se le ha dado un nombre y que puede ser aplicada en otros contextos.

3.2.1 Patrones de arquitectura

Para el desarrollo del sistema propuesto se hará uso del patrón Modelo Vista Controlador (MVC) que es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se usa con frecuencia en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el SGBD y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. Se decidió usar dicho patrón debido a que facilita agregar nuevos tipos de datos según sea requerido por la aplicación, pues son independientes del funcionamiento de las otras capas y el

CAPÍTULO 3. DISEÑO DEL SISTEMA

mantenimiento en caso de errores. Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema y posibilita el escalamiento de la aplicación en caso de ser requerido.

Con el uso del framework JSF se pone de manifiesto el uso del patrón antes mencionado. Este framework posee un grupo de componente que facilita al usuario interactuar con la aplicación implementando de este modo la vista; permite definir clases controladoras que recojan la información introducida y que dispongan de métodos que responden a las acciones del usuario y manipulen los datos contenidos en el modelo.

En el sistema, las vistas son todas las clases interfaz con las que interactúa directamente el usuario y están construidas mayormente con componentes JSF. Usa la librería de componentes Richfaces que se integra fácilmente con el framework Seam y permite generar vistas no necesariamente basadas en HTML; incluye conversión y validación de campos, establecimiento de reglas de navegación declarativas, la internacionalización y accesibilidad de la interfaz de usuario, un modelo orientado a eventos y combinado con Facelets, se elimina la necesidad de dos motores de renderización mejorando el rendimiento en general además de que provee la capacidad añadida de la tecnología de plantillas de Facelets.

El controlador que engloba toda la lógica del negocio, está representado por las clases controladoras que son las encargadas de responder a las solicitudes del usuario. Como framework de integración en esta capa se usa Seam. El mismo integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB) facilitando la comunicación entre estas capas.

El modelo de datos se encarga del almacenamiento y acceso a los datos para lo cual utiliza la implementación de Java Persistence API (JPA) de Hibernate 3.3, lo cual posibilita minimizar por un lado las configuraciones en XML sin chequeo de tipos y por otro lado se elimina gran parte del código infraestructural en cuanto a transacciones, la transmisión del contexto de persistencia, etc. Hibernate implementa una serie de patrones para la administración del modelo de datos, entre ellos están los siguientes:

- ✓ **Active record:** Permite la asociación de filas únicas de la base de datos con objetos del lenguaje de programación usado.

- ✓ **Identity map:** Mantiene referencias a los objetos cargados y devuelve las mismas cada vez que se solicita un objeto ya cargado. De esta forma, actúa como un caché dentro de la transacción, lo cual también contribuye a reducir el número de accesos a la base de datos.
- ✓ **Identity field:** Permite mapear las relaciones entre referencias entre objetos y claves entre tablas.
- ✓ **Foreign key mapping:** Posibilita mapear las relaciones de asociación de clave foránea.
- ✓ **Association Table mapping:** Permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, lo cual crea una base de datos orientada a objetos virtuales, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).
- ✓ **Lazy load:** Carga los distintos componentes de una clase a medida que son solicitados.
- ✓ **Query object:** Facilita manipular las consultas en tiempo de ejecución. Dado que las consultas son los objetos, se pueden cambiar para agregar o quitar condiciones. Esto proporciona la flexibilidad necesaria para crearlas o modificarlas dinámicamente.

3.2.2 Patrones de diseño

En los últimos años, ha crecido de manera espectacular el interés por los patrones de diseño, constituyéndose como una importante técnica para construir software orientado a objetos. En el desarrollo del sistema se utilizan múltiples patrones de diseño, entre ellos se encuentran los Patrones de Software para la Asignación General de Responsabilidades (GRASP). A continuación se mencionan los más significativos:

- ✓ **Experto:** Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se puede obtener un diseño con mayor cohesión y así la información se mantiene encapsulada.
- ✓ **Creador:** El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, almacena o maneja varias instancias de la clase, contiene o agrega la clase.

- ✓ **Controlador:** Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.
- ✓ **Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases
- ✓ **Alta cohesión:** La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento, cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

Estos patrones se aplicaron en la definición de cada clase con el objetivo de lograr que las clases realicen las funciones relacionadas con la información que maneja para que no se sobrecarguen. Además las clases se diseñaron de forma tal que estén lo menos relacionadas entre sí para que los cambios que puedan ocurrir en una clase repercutan lo menos posibles en las otras. Se definieron las clases controladoras, las cuales son las responsables de gestionar las operaciones del sistema. Se le dio la responsabilidad de instanciar a una clase a aquellas clases que contengan a la misma.

3.3. Definición de elementos de diseño

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.

<<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

CAPÍTULO 3. DISEÑO DEL SISTEMA

<<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.

Para modelar los diagramas de paquetes y los de clases del diseño se utilizará la siguiente nomenclatura: Diagrama de paquetes: Diagrama de paquetes_<Nombre del módulo>, Diagramas de clases del diseño: DCD_<Nombre de la funcionalidad>.

Para representar las clases previamente mencionadas se procedió usando la siguiente nomenclatura: frmsiaps, <Nombre de la vista>.siaps, CR<Nombre de la Opción>.java (Exportar), CC<Nombre de la Opción>.java (Registrar), CL<Nombre de la Opción>.java (Listar), CM<Nombre de la Opción>.java (Modificar), CE<Nombre de la Opción>.java (Eliminar), CD<Nombre de la Opción>.java (Detalles) y <Tr o Tn o Tb><Nombre de la Opción>.java, para los formularios, páginas clientes, clases servidoras y entidades respectivamente ubicadas en los diferentes niveles de la aplicación.

En los diagramas se encuentran otros elementos como son los frameworks JSF, EJB 3.0, JasperReport, JPA, Hibernate y Seam. Además, existen un conjunto de clases entre las que se identifican librerías, JavaScript, JSF (Servlets), Seam (Servlets), Bitácora, entre otras.

3.5. Descripciones textuales.

3.5.1 Clases comunes. Descripción.

Capa de Presentación	
Nombre:	Propósito:
 <p>Figura 3.3. Clase JavaScript</p>	<p>Proveer el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.</p>
Descripción:	
<p>Permite realizar las validaciones del lado del cliente y construir páginas más dinámicas integradas a un navegador web</p>	
Nombre:	Propósito:
 <p>Figura 3.4. Clase Librerías</p>	<p>Permite la creación de interfaces.</p>
Descripción:	
<p>Contiene un conjunto de librerías que permiten construir una interfaz de usuario con componentes reutilizables y extensibles. Reducen significativamente la carga de construir y mantener aplicaciones web con componentes de interfaz del lado del servidor.</p>	
Nombre:	Propósito:
 <p>Figura 3.5. Clase JSF Servlet</p>	<p>Interceptar las peticiones a las páginas JSF.</p>
Descripción:	
<p>Es el controlador de JSF que intercepta las peticiones de las páginas clientes, asociándoles a estas páginas, clases java que recogen la información introducida y que disponen de métodos que responden a</p>	

CAPÍTULO 3. DISEÑO DEL SISTEMA

las acciones del usuario. Además, prepara el contexto JSF antes de enrutar a las páginas correspondientes e interviene en la construcción de la respuesta para generar la vista, luego de ser invocada una petición.

Nombre:	Propósito:
---------	------------



Figura 3.6. Paquete Framework JSF

Simplificar el desarrollo de las interfaces de usuario en aplicaciones java basadas en el patrón MVC.

Descripción:

La tecnología Java Server Faces es un framework 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.

Capa de Negocio

Nombre:	Propósito:
---------	------------



Figura 3.7. Clase CRExportar

Proveer las funcionalidades para exportar a diferentes formatos.

Descripción:

Permite realizar las funciones para exportar a diferentes tipos de archivos mediante el uso de las librerías del Framework JasperReports.

Nombre:	Propósito:
---------	------------

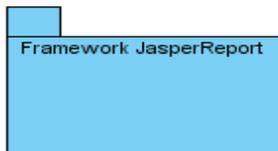


Figura 3.8. Paquete Framework JasperReports

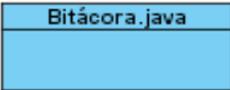
Añadir características de generación de reportes a aplicaciones java.

Descripción:

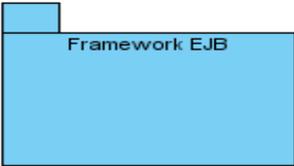
Es una librería de clases de Java de código abierto desarrollada para facilitar el agregar capacidades de reporte a las aplicaciones Java. Permite realizar reportes de código abierto que tiene como función el

CAPÍTULO 3. DISEÑO DEL SISTEMA

llevar documentos ricos en contenido a la pantalla, a la impresora, o a archivos PDF, HTML, XLS, CSV y XML.

Nombre:	Propósito:
 <p>Figura 3.9. Clase Bitácora.java</p>	Proveer las funcionalidades para realizar las auditorias del sistema.

Descripción:
 Permite realizar las funciones para almacenar datos como la fecha, hora, usuario, contraseña, entre otros, del usuario que inicia y finaliza la sesión. Además, datos como el módulo y funcionalidad accedidos así como las que acciones ejecutadas.

Nombre:	Propósito:
 <p>Figura 3.10. Paquete Framework EJB</p>	Encapsular la lógica de negocio que cumplimenta el propósito de la aplicación.
	Observaciones:
	El Framework EJB está incluido en las capas de Negocio y Persistencia.

Descripción:
 Es una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables utilizando el lenguaje de programación Java. El objetivo de Enterprise JavaBeans (EJB) 3.0 es simplificar el desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java. Forma parte de la especificación JEE 5.

Capa de Datos

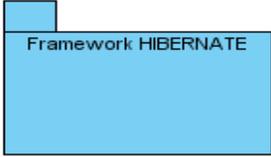
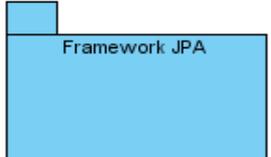
Nombre:	Propósito:
 <p>Figura 3.11. Clase EntityManager</p>	Gestionar las entidades proveyendo servicios de persistencia.

Descripción:
 Permite realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) que impliquen entidades.

CAPÍTULO 3. DISEÑO DEL SISTEMA

Nombre:	Propósito:
 <p>Figura 3.12. Clase EntityManager</p>	Agregar consultas que pueden aplicarse a las entidades del modelo.
Descripción:	
Permite encontrar objetos persistentes manejando cierto criterio de búsqueda. 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.	
Nombre:	Propósito:
 <p>Figura 3.13. Clase EntityManager</p>	Agrupar las operaciones sobre datos persistentes en una unidad transaccional.
Descripción:	
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.	
Nombre:	Propósito:
 <p>Figura 3.14. Clase Entidades Bitácora</p>	Agrupar las entidades que contienen la información de las auditorías del sistema.
Descripción:	
Contiene el conjunto de entidades que poseen la información de los usuarios y sus trazas en cuanto a sesión utilizada, módulos accedidos, funcionalidades permitidas, acciones realizadas y atributos modificados. Son utilizadas por la clase Bitácora.java para realizar las auditorías del sistema.	
Nombre:	Propósito:

CAPÍTULO 3. DISEÑO DEL SISTEMA

 <p style="text-align: center;">Figura 3.15. Paquete Framework HIBERNATE</p>	<p>Proveer el mapeo objeto/relacional con la base de datos.</p>
Descripción:	
<p>Conjunto de clases agrupadas en componentes que constituyen una herramienta de Mapeo objeto/relacional ó ORM de código abierto (Object Relational Mapping) y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada permite generar Base de Datos en cualquiera de los entornos soportados: Oracle, PostgreSQL, DB2, MySQL, entre otras.</p>	
Nombre:	Propósito:
 <p style="text-align: center;">Figura 3.16. Paquete Framework JPA</p>	<p>Unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.</p>
Descripción:	
<p>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, aunque su uso no se limita a los componentes software EJB. Permite 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 orientación a objetos al interactuar con una base de datos.</p>	
Nombre:	Propósito:
 <p style="text-align: center;">Figura 3.17. Clase SEAM Servlet</p>	<p>Proveer la interacción de la capa de presentación con la de negocio.</p>
Observaciones:	
<p>No se encuentra en ninguna capa en específico, sino que se encuentra ubicado entre la capa de Presentación y la de</p>	

	Negocio.
Descripción:	
Es el controlador de SEAM que capta las peticiones derivadas de la interacción del usuario después de interactuar con el Servlet de JSF. 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.	

3.5.2 Gestionar relaciones. Descripción.

Capa de Presentación	
Nombre:	Propósito:
 <p>Figura 3.18. Clase CCRelaciones.java</p>	Proveer la interacción con el usuario.
Descripción:	
La clase CCRelaciones.java es una página web que se ejecuta del lado del cliente sobre un navegador. Posibilita registrar los datos necesarios referentes a la relación interpersonal existente entre los miembros de la familia. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.	
Nombre:	Propósito:
 <p>Figura 3.19. frmsiaps</p>	Enviar los datos a las páginas servidoras.
Descripción:	
La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para insertar todos los datos necesarios para registrar una nueva relación entre dos miembros de la familia. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.	
Capa de Negocio	
Nombre:	Propósito:

 CCRelaciones.java	Proveer una respuesta a las peticiones realizadas en la vista.
---	--

Figura 3.20. Clase CCRelaciones.java

Descripción:

La clase CCRelaciones.java 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. Se encarga de gestionar la acción de registrar los datos sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Capa de Datos

Nombre:	Propósito:
<div style="border: 1px solid black; background-color: #e1eef6; padding: 5px; margin-bottom: 10px;"> TrRelacionPaciente.java </div> <pre style="font-family: monospace; font-size: 0.9em; margin: 0;">-id : int -version : Integer -cid : Integer -idFamiliar : int -estado : boolean</pre> <p style="text-align: center;">Figura 3.21. Clase TrRelacionPaciente.java</p>	Proveer el mapeo con la base de datos.

Descripción:

La clase TrRelacionPaciente.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de las relaciones interpersonales entre los miembros de la familia. Representa 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 del Framework Hibernate y JPA.

Nombre:	Propósito:
---------	------------

<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;">TnTipoRelacion.java</p> <pre style="margin: 0;">-id : int -version : Integer -cid : Integer -valor : string -codigo : string</pre> </div> <p>Figura 3.22. Clase TnTipoRelacion.java</p>	<p>Proveer el mapeo con la base de datos.</p>
--	---

Descripción:

La clase TnTipoRelacion.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de los tipos de relaciones interpersonales. Representa 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 del Framework Hibernate y JPA.

Nombre:

Propósito:

<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;">TbPaciente.java</p> <pre style="margin: 0;">-idPaciente : int -idHsf : int -idNivelEducacional : int -idZonaAps : int -idOcupación : int -idFactorRh : int -idGrupoDispensarial : int -idColorPiel : int -idGrupo : int -idSexo : int -idUnidad : int -fechaNacimiento : date -idProfesión : int -nroHci : int -fechaCaptación : date -ci : varchar -nombrePaciente : varchar -primerApellido : varchar -segundoApellido : varchar -jefeNúcleo : int</pre> </div> <p>Figura 3.23. Clase TbPaciente.java</p>	<p>Proveer el mapeo con la base de datos.</p>
--	---

Descripción:

La clase TbPaciente.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de los pacientes. Representa 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 del Framework Hibernate y JPA.

3.5.3 Gestionar familiograma. Descripción.

Capa de Presentación	
Nombre:	Propósito:
<div style="text-align: center;">  <p>Figura 3.24. Clase mostrar.siaps</p> </div>	<p>Proveer la interacción con el usuario.</p>
Descripción:	
<p>La clase mostrar.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite mostrar gráficamente la estructura familiar. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>	
Nombre:	Propósito:
<div style="text-align: center;">  <p>Figura 3.25. frmsiaps</p> </div>	<p>Enviar los datos a las páginas servidoras.</p>
Descripción:	
<p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para mostrar la estructura de la familia. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>	
Nombre:	Propósito:
<div style="text-align: center;">  <p>Figura 3.26. Clase generar.siaps</p> </div>	<p>Proveer la interacción con el usuario.</p>
Descripción:	
<p>La clase <generar.siaps> es una página web que se ejecuta del lado del cliente sobre un navegador. Permite crear gráficamente la estructura familiar. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>	
Nombre:	Propósito:

CAPÍTULO 3. DISEÑO DEL SISTEMA

 Figura 3.27. frmsiaps	Enviar los datos a las páginas servidoras.
Descripción:	
La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para mostrar la estructura de la familia. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.	
Capa de Negocio	
Nombre:	Propósito:
 CMostrarFamiliograma.java Figura 3.28. Clase CMostrarFamiliograma.java	Proveer una respuesta a las peticiones realizadas en la vista.
Descripción:	
La clase CMostrarFamiliograma.java 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. Se encarga de gestionar la acción de mostrar la estructura de la familia sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.	
Nombre:	Propósito:
 CGenerarFamiliograma.java Figura 3.29. Clase CGenerarFamiliograma.java	Proveer una respuesta a las peticiones realizadas en la vista.
Descripción:	
La clase CGenerarFamiliograma.java es una clase que se ejecuta del lado del servidor. Permite darle	

CAPÍTULO 3. DISEÑO DEL SISTEMA

respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de generar la estructura de la familia sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Nombre:	Propósito:
 <p data-bbox="365 688 495 714">CDatos.java</p> <p data-bbox="227 745 633 777">Figura 3.30. Clase CDatos.java</p>	Proveer una respuesta a las peticiones realizadas en la vista.

Descripción:

La clase CDatos.java es una clase auxiliar 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. Se encarga de encapsular un conjunto de datos con el objetivo de facilitar la gestión de estos. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Nombre:	Propósito:
 <p data-bbox="365 1207 495 1232">CPersona.java</p> <p data-bbox="211 1260 649 1291">Figura 3.31. Clase CPersona.java</p>	Proveer una respuesta a las peticiones realizadas en la vista.

Descripción:

La clase CPersona.java es una clase auxiliar 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. Se encarga de encapsular un conjunto de datos de una persona con el objetivo de facilitar la gestión de estos. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Capa de Datos

Nombre:	Propósito:
---------	------------

<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> TrRelacionPaciente.java -id : int -version : Integer -cid : Integer -idFamiliar : int -estado : boolean </div> <p style="text-align: center;">Figura 3.32. Clase TrRelacionPaciente.java</p>	Proveer el mapeo con la base de datos.
Descripción:	
La clase TrRelacionPaciente.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de las relaciones interpersonales entre los miembros de la familia. Representa 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 del Framework Hibernate y JPA.	
Nombre:	Propósito:
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> TnTipoRelacion.java -id : int -version : Integer -cid : Integer -valor : string -codigo : string </div> <p style="text-align: center;">Figura 3.33. Clase TnTipoRelacion.java</p>	Proveer el mapeo con la base de datos.
Descripción:	
La clase TnTipoRelacion.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de los tipos de relaciones interpersonales. Representa 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 del Framework Hibernate y JPA.	
Nombre:	Propósito:

<pre>TbPaciente.java -idPaciente : int -idHsf : int -idNivelEducativo : int -idZonaAps : int -idOcupación : int -idFactorRh : int -idGrupoDispensarial : int -idColorPiel : int -idGrupo : int -idSexo : int -idUnidad : int -fechaNacimiento : date -idProfesión : int -nroHci : int -fechaCaptación : date -ci : varchar -nombrePaciente : varchar -primerApellido : varchar -segundoApellido : varchar -jefeNúcleo : int</pre>	<p>Proveer el mapeo con la base de datos.</p>
Descripción:	
<p>La clase TbPaciente.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de los pacientes. Representa 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 del Framework Hibernate y JPA.</p>	

Figura 3.34. Clase TbPaciente.java

En este capítulo se realizó una propuesta de solución al problema planteado sentándose las bases para comenzar la implementación del sistema. Se elaboraron los diagramas de clases del diseño, se describieron sus clases para un mejor entendimiento, se especificaron y justificaron los patrones de arquitectura y diseño utilizados.

Capítulo 4. Implementación

La implementación comienza con el resultado del diseño y tiene como propósito fundamental desarrollar la arquitectura y el sistema como un todo. En este capítulo se obtienen los principales artefactos correspondientes al flujo de trabajo implementación. Se describe cómo los elementos del modelo de diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Se realiza un estudio de los estándares de codificación, estilos y tratamientos de errores a utilizar en la implementación del sistema.

4.1. Propuesta de integración

Con el objetivo de reutilizar software existente y por tanto disminuir los costos, en la actualidad no se concibe ningún sistema de forma aislada por lo que se hace necesario pensar en su integración con otros sistemas en funcionamiento o en vías de desarrollo. El módulo medicina familiar al cual pertenece el componente web análisis del familiograma no es la excepción. El mismo es el núcleo del SIAPS y mantiene una estrecha relación con los restantes módulos del sistema antes mencionado. El módulo de configuración brinda información útil y necesaria al módulo de medicina familiar y éste a su vez nutre al resto de los módulos del sistema con su información para que los mismos puedan realizar sus funcionalidades.

Se contempla la integración al SIAPS como plataforma única para la gestión, procesamiento y transmisión de la información clínica en el SNS. La comunicación se realizará a nivel de base de datos.

4.2. Implementación

En este flujo de trabajo se comienza con el resultado del diseño y se implementan el sistema en términos de componentes, es decir, ficheros de código fuente, ficheros de código binario, ejecutables entre otros. El objetivo fundamental de la implementación es desarrollar la arquitectura y el sistema como un todo.

4.2.1 Diagrama de despliegue

El Diagrama de Despliegue es un diagrama que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de

despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP, entre otros. El objetivo de estos es mostrar cómo y dónde se desplegará el sistema.

El diagrama de despliegue de la solución propuesta quedaría del siguiente modo:

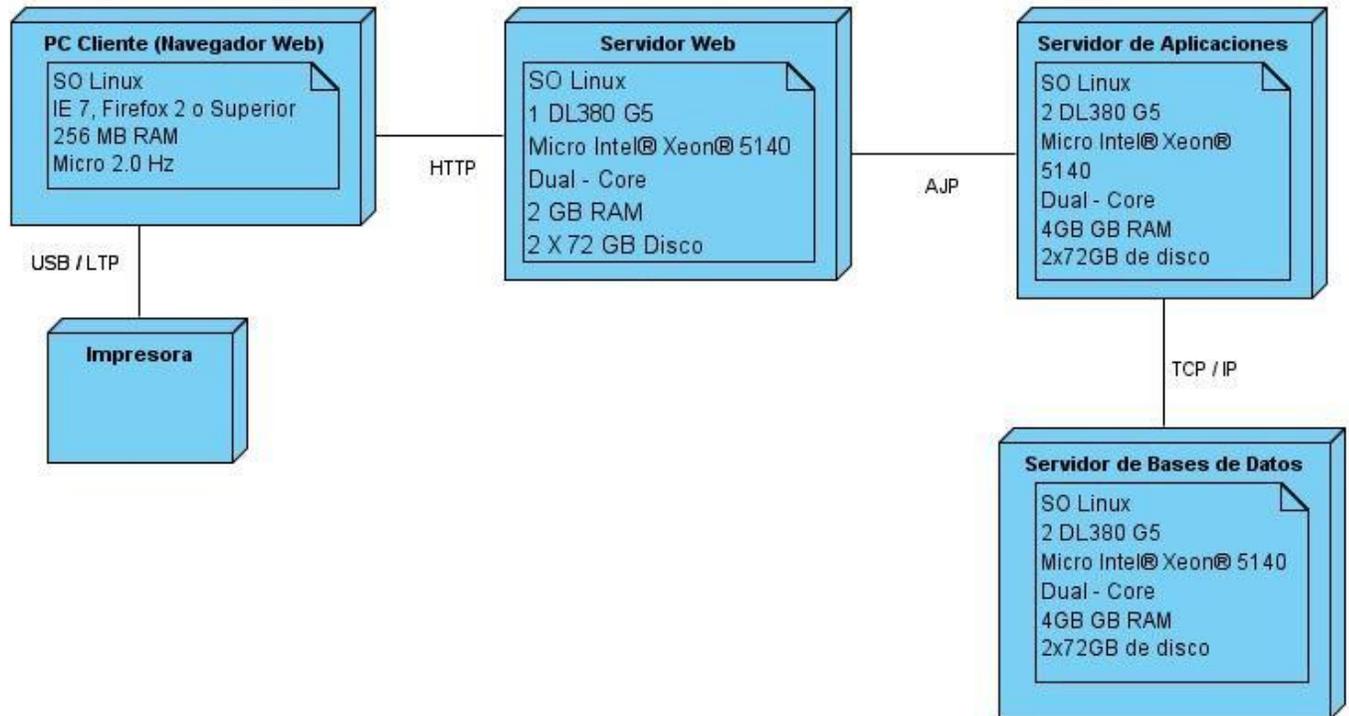


Figura 4.1. Diagrama de Despliegue.

4.2.1.1 Descripción de los elementos

Servidor Web: Servidor Web Apache que provee el servicio de interfaz al usuario final mediante un portal convencional y otro WAP para dispositivos móviles, pues es el ordenador que estará como fachada a Internet, al mismo tiempo será el puente o proxy para entrar al clúster de servidor(es) de aplicaciones que proporciona Jboss. En escenarios híbridos brindará la ejecución y actualización de la solución local mediante la tecnología Java Web Start.

Servidor(es) de Aplicaciones: Servidor de Aplicaciones Jboss certificado por SUN para el estándar JEE5, hospedará la solución integrada, proveerá de un clúster para balancear la carga de peticiones hechas por los usuarios garantizando de esta manera disponibilidad de la información mostrada.

Servidor de Bases de Datos: Servidor de Datos PostgreSQL, en el cual residirá toda la información operacional, que será alimento del repositorio central de Historias Clínicas Electrónicas (HCE).

Estaciones Clientes: Serán las estaciones los usuarios, las cuales servirán para acceder al sistema web o integrado mediante el navegador o al sistema local o de escritorio mediante un ícono en el escritorio del ordenador.

Procesadores:

Servidor(es) de Aplicaciones:

Sistema Operativo Linux

2 DL380 G5

Procesador Intel® Xeon® 5140 Dual - Core

4GB de memoria

2x72GB de disco

Servidor de Base de Datos:

Sistema Operativo Linux

1 DL380 G5

Procesador Intel® Xeon® 5140 Dual – Core

4GB de memoria

Servidor Web:

Sistema Operativo Linux

1 DL380 G5

Procesador Intel® Xeon® 5140 Dual - Core

2 GB de memoria

2x72GB de disco

Estaciones Clientes:

Sistema Operativo Linux.

IE 7, Firefox 2 o versiones superiores.

256 Mb de memoria RAM

Microprocesador de 2.0 Hz

4.2.2 Tratamiento de errores

En un sistema informático, una excepción es un evento que ocurre durante la ejecución de un programa que interrumpe el flujo normal de la secuencia de instrucciones de ese programa, estos eventos requieren de un tratamiento especial para evitar que interrumpan el correcto funcionamiento del programa en ejecución. Si se logra un correcto manejo de las excepciones aumentará la calidad de la aplicación que se desarrolle y se obtendrán sistemas más robustos y fiables.

Para el tratamiento de las excepciones en el sistema propuesto se hizo uso de todas las facilidades que brinda la plataforma utilizada. Se definieron un conjunto de excepciones para cada porción de código donde se espere una situación anómala que pueda interrumpir el correcto funcionamiento del sistema. Además se hace uso de un grupo de excepciones predefinidas por el marco de trabajo que se utiliza en el sistema.

El uso de diferentes tecnologías y la integración que existe entre ellas, permiten capturar y controlar posibles situaciones desde diferentes puntos de la aplicación. En las páginas clientes se cuenta con un conjunto de componentes denominados validadores, que permiten establecer tipos de datos y formatos controlando que el envío de los activos al servidor sean los esperados.

Además, el marco de trabajo Seam brinda un potente conjunto de excepciones predefinidas, que conjuntamente con la clase `FacesMessages`, permite tratar estas situaciones desde las clases controladoras correspondientes y mostrar mediante la clase antes mencionada, los resultados del tratamiento. Seam permite además mediante el fichero de configuración `page.xml`, todo un flujo de navegación basado en excepciones.

4.2.3 Estándares de codificación.

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. A continuación se enuncian las reglas de codificación definidas para el desarrollo del sistema:

Idioma:

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

Comentarios

Todos los ficheros fuente comienzan con un comentario en el que se lista el nombre de la clase, información de la versión, fecha, y copyright. Los comentarios deben añadir claridad al código, ser concisos y contar el por qué y no el cómo.

Existen dos tipos de comentarios: comentarios de implementación y comentarios de documentación. Los comentarios de implementación describen el código, los de documentación una visión de más alto nivel para gente que no tiene el código a mano y que sólo quiere usarlo. La información de los comentarios no debe ser evidente, deben ser cosas que no están contenidas en el código. Los comentarios no deben estar en cajas dibujadas con asteriscos o similar ni deben incluir caracteres extraños.

Formatos de los comentarios de implementación

Los programas pueden tener cuatro estilos de comentarios de implementación: de bloque, de una línea, de remolque, y de fin de línea.

Comentarios de Bloque

Los comentarios de bloque se usan para dar descripciones de ficheros, métodos, estructuras de datos y algoritmos. Los comentarios de bloque se podrán usar al comienzo de cada fichero o antes de cada método. También se pueden usar en otros lugares, tales como el interior de los métodos. Los comentarios de bloque en el interior de una función o método deben ser indentados al mismo nivel que el código que describen.

Un comentario de bloque debe ir precedido por una línea en blanco que lo separe del resto del código.

```
/*
 * Aquí hay un comentario de bloque.
 */
```

Los comentarios de bloque pueden comenzar con `/*-`, que es reconocido por `indent(1)` como el comienzo de un comentario de bloque que no debe ser reformateado. Ejemplo:

```
/*-
 * Aquí tenemos un comentario de bloque con cierto
 * formato especial que quiero que ignore indent(1).
 *
 * uno
 *     dos
 *         tres
 */
```

Comentarios de una Línea

Pueden aparecer comentarios cortos de una única línea al nivel del código que siguen. Si un comentario no se puede escribir en una línea, debe seguir el formato de los comentarios de bloque. Un comentario de una sola línea debe ir precedido de una línea en blanco.

```
if (condicion) {
    /* Código de la condicion. */
    ...
}
```

Comentarios de Fin de Línea

El delimitador de comentario // puede convertir en comentario una línea completa o una parte de una línea. No debe ser usado para hacer comentarios de varias líneas consecutivas; sin embargo, puede usarse en líneas consecutivas para comentar secciones de código.

Aquí tienes ejemplos de los tres estilos:

```
if (foo > 1) {
  // Hacer algo.
  ...
}
else {
  return false; // Explicar aquí por qué.
}
//if (bar > 1) {
//
// // Hacer algo.
// ...
//}
//else {
// return false;
//}
//}
```

Comentarios de Remolque

Pueden aparecer comentarios muy pequeños en la misma línea que describen, pero deben ser movidos lo suficientemente lejos para separarlos de las sentencias. Si más de un comentario corto aparece en el mismo trozo de código, deben ser indentados con la misma profundidad.

Aquí se muestra un ejemplo de comentario de remolque:

```
if (a == 2) {
  return TRUE;           /* caso especial */
} else {
  return isPrime(a);     /* caso general */
}
```

Líneas y espacios en blanco.

Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas.

Se deben usar siempre dos líneas en blanco en las siguientes circunstancias:

- ✓ Entre las secciones de un fichero fuente
- ✓ Entre las definiciones de clases e interfaces.

Se debe usar siempre una línea en blanco en las siguientes circunstancias:

- ✓ Entre métodos
- ✓ Entre las variables locales de un método y su primera sentencia
- ✓ Antes de un comentario de bloque o de un comentario de una línea.
- ✓ Entre las distintas secciones lógicas de un método para facilitar la lectura.

Se deben usar espacios en blanco en las siguientes circunstancias:

Una palabra clave del lenguaje seguida por un paréntesis debe separarse por un espacio.

Indentación y longitud de la línea.

Se deben emplear cuatro espacios como unidad de indentación. La construcción exacta de la indentación (espacios en blanco contra tabuladores) no se especifica. Los tabuladores deben ser exactamente cada 8 espacios.

Evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

Variables, constantes, clases y métodos.

Todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres subguión "_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje.

CAPÍTULO 4. IMPLEMENTACIÓN

Los nombres de las variables deben ser cortos pero con significado. La elección del nombre de una variable debe ser un mnemónico, designado para indicar a un observador casual su función. Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.

Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión ("_"). (Las constantes ANSI se deben evitar, para facilitar su depuración.)

Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas.

Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula.

En este capítulo se realizó un resumen de los principales elementos de la implementación del sistema. Se definieron los módulos con que debe integrarse el sistema propuesto para su correcto funcionamiento. Se estableció la estrategia para el despliegue de la aplicación. Además se describieron los estándares de codificación usados durante la implementación así como el modo en que se llevaría a cabo el manejo de las excepciones en el sistema para lograr un software con mayor calidad y fiabilidad.

Conclusiones

La realización del presente trabajo se cumplieron los objetivos propuestos, por lo que se pueden plantear las siguientes conclusiones:

- El estudio realizado del proceso de gestión de la información del familiograma en la APS demostró la necesidad de realizar un sistema informático que solucione los problemas encontrados en el proceso de gestión del familiograma.
- Los sistemas informáticos analizados para la gestión de la información en la APS, no responden totalmente a las necesidades identificadas.
- El componente web desarrollado permitirá a los profesionales de la salud estudiar a la familia con una mayor profundidad y precisión. Lo que contribuirá a mejorar los servicios prestados y por tanto la calidad de vida de la población.

Recomendaciones

Las recomendaciones de la investigación están dirigidas a sugerir acciones para complementar el producto obtenido. Por lo que para el buen desempeño y puesta en marcha de la aplicación se hacen las siguientes recomendaciones:

- ✓ Continuar el perfeccionamiento de las funcionalidades del sistema para mejorar el rendimiento y rapidez del procesamiento de la información.
- ✓ Implementar la funcionalidad Generar Familiograma que permita al usuario su creación personalizada.
- ✓ Perfeccionar la visualización de los problemas de salud de los miembros de la familia en la funcionalidad Representar Familiograma.

Referencias Bibliográficas

1. Navarrete Bermello, Rosa Lic. y Suárez Rodríguez, Ada Téc. Revista de Ciencias Médicas. EL A, B, C DE LA GESTIÓN DE INFORMACIÓN: PARTE II LA BÚSQUEDA AVANZADA. [En línea] 2010. http://www.cpicmha.sld.cu/hab/Vol16_1_10/hab19110.html.
2. Dirección de Informática del MINSAP. PROGRAMA DE INFORMATIZACION DEL SECTOR DE LA. [En línea] 2003. [Citado el: 22 de ENERO de 2011.] <http://www.di.sld.cu/estrategias/Programa%20de%20Informatizacion.pdf>.
3. Ídem 2.
4. Alegre Palomino, Yuri Dr. y Suárez Bustamante, Miguel Dr. Fascículos CADEC. Instrumentos de Atención a la Familia. [En línea] 2006. 6. http://www.saluddealtura.com/fileadmin/PDF/CAPACITACION/Documento3_familiograma.pdf.
5. Organización Mundial de la Salud. ¿Qué es un sistema de salud? [En línea] 2005. [Citado el: 13 de Enero de 2011.] <http://www.who.int/features/qa/28/es/index.html>.
6. Taylor, Robert B. Medicina de Familia Principios y prácticas. [En línea] 2006. http://books.google.com.cu/books?id=d3A0gNxXoVQC&pg=PA5&dq=medicina+familiar&hl=es&ei=9VL4TNcOw4qcB7TlxOwD&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCYQ6AEwAA#v=onepage&q=medicina%20familiar&f=false.
7. Presno Labrador, Clarivel Dra. Revista Cubana de Medicina General Integral. El médico de familia en Cuba. [En línea] 2006. http://bvs.sld.cu/revistas/mgi/vol20_5-6_04/mgi135_604.htm.
8. Montoya Guerrero, Liliana y García Vargas, Alexis. Módulo Medicina Familiar para Escritorio del Sistema Integral para la Atención Primaria. Ciudad de La Habana: s.n., 2010.
9. Campos Cosme, Anny y Rodríguez Ronquillo, Reynier Alejandro. Informatización de la Historia de Salud Familiar de la Atención Primaria de Salud. Universidad de las Ciencias Informáticas. La Habana, 2010. 80 Páginas.

REFERENCIAS BIBLIOGRÁFICAS

10. Ídem 5.
11. Ídem 5.
12. Guerrero Montoya, Liliana y García Vargas, Alexis. Módulo Medicina Familiar para Escritorio del Sistema Integral para la Atención Primaria. Universidad de las Ciencias Informáticas. La Habana, 2010. 82 Páginas.
13. Dorado, M. P. Gaspar; Loria Castellanos, J.; Gerrero Martínez, M. B. SISTEMA DE INFORMACIÓN CIENTÍFICA REDALYC. Aptitud clínica hacia el estudio integral de la familia en residentes de medicina familiar. [En línea] 2006. <http://redalyc.uaemex.mx/pdf/507/50780305.pdf>.
14. Ídem 5.
15. My Family Health Portrait. Sobre Mi Retrato de Salud Familiar. [En línea] 2009. <https://familyhistory.hhs.gov/fhh-web/popup/getHelp/footerDetailsAboutThisSite.action>.
16. GenoPro. GenoPro - Programa de árboles familiares. [En línea] 2007. <http://www.genopro.com/es/>.
17. Colectivo de Autores. Habana 2001. Sistema Informático para la Gestión Médica en la APS (APUS). [En línea] [Citado el: 17 de Enero de 2011.] <http://www.hab2001.sld.cu/arrepdf/00196.pdf>.
18. Marquès Puig, Joan Manuel, Vilajosana Guillén, Xavier y López, Pedro A. García. Arquitecturas, Paradigmas y aplicaciones de los sistemas distribuidos. [En línea] http://docencia.izt.uam.mx/sgm8/nos/sistemas_distribuidos_01.pdf.
19. Avendaño, Márquez, y otros. UDLAP. Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español. [En línea] 2004. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf.
20. Viveros Serrano, Carlos Andrés. UDLAP. Sistema generador de aplicaciones Web configurables para el registro en línea de congresos utilizando JavaServer Faces . [En línea] 2006. [Citado el: 10 de Enero de 2011.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/viveros_s_ca/capitulo3.pdf.

REFERENCIAS BIBLIOGRÁFICAS

21. Junta de Andalucía. RichFaces. [En línea] [Citado el: 10 de Enero de 2011.]
<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces>.
22. Theo, Otto. Desarrollo en Web. Facelets y JSF – Uso de Templates. [En línea] 2008. [Citado el: 11 de Enero de 2011.] <http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>.
23. Alonso Ramos, Juan. Adictos al Trabajo. Introducción a Ajax4jsf. [En línea] 2007. [Citado el: 11 de Enero de 2011.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
24. Seam City. Qué es JBoss Seam? [En línea] [Citado el: 11 de Enero de 2011.]
<http://seamcity.madeinxpain.com/archives/que-es-jboss-seam>.
25. Suárez González, Héctor. Java Hispano. Manual Hibernate. [En línea] 2003. [Citado el: 11 de Enero de 2011.] <http://www.javahispano.org/contenidos/archivo/77/ManualHibernate.pdf>.
26. Junta de Andalucía. Hibernate. [En línea] [Citado el: 11 de Enero de 2011.]
<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Hibernate>.
27. Ciberaula. ¿Qué es Java? [En línea] [Citado el: 11 de Enero de 2011.]
http://java.ciberaula.com/articulo/que_es_java/.
28. CAVSI. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? . [En línea] [Citado el: 12 de Enero de 2011.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
29. PostgreSQL-es. Sobre PostgreSQL. [En línea] 2010. [Citado el: 12 de Enero de 2011.]
http://www.postgresql-es.org/sobre_postgresql.
30. Ídem 25.
31. Salinas Caro, Patricio. Universidad de Chile. Unified Modeling Language. [En línea] [Citado el: 13 de Enero de 2011.] <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>.

REFERENCIAS BIBLIOGRÁFICAS

32. Real Academia Española. Diccionario de la Lengua Española. [En línea] 2008. [Citado el: 2011 de Enero de 27.] <http://drae2.es/símbolo>.
33. Definición.de. Definición de Parentesco. [En línea] [Citado el: 22 de Enero de 2011.] <http://definicion.de/parentesco/>.
34. Entorno Virtual de Aprendizaje. Introducción a la Disciplina de Requisitos de RUP. [En línea] 2011. [Citado el: 10 de Marzo de 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_06/Conf_7/Materiales_complementarios/Introduccion_a_la_Disciplina_de_Requisitos.pdf.
35. Sommerville, Ian. Ingeniería de Software. Madrid: Pearson Educación, 2005. pág. 712.
36. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de software. Madrid: Pearson Educación.S.A, 2000.

Bibliografía

1. Alegre Palomino, Yuri Dr. y Suárez Bustamante, Miguel Dr. Fascículos CADEC. Instrumentos de Atención a la Familia. [En línea] 2006. 6.
http://www.saluddealtura.com/fileadmin/PDF/CAPACITACION/Documento3_familiograma.pdf.
2. Alonso Ramos, Juan. Adictos al Trabajo. Introducción a Ajax4jsf. [En línea] 2007. [Citado el: 11 de Enero de 2011.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
3. Álvarez, Sara. desarrolloweb.com. Sistemas gestores de bases de datos. [En línea] 2007.
<http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
4. Avendaño, Márquez, y otros. UDLAP. Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español . [En línea] 2004.
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf.
5. Belmonte Fernández, Oscar. Universitat Jaume. Introducción al lenguaje de programación Java. [En línea] 2005. <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>.
6. Bermello Navarrete, Rosa Lic. y Rodríguez Suárez, Ada Téc. Revista de Ciencias Médicas. EL A, B, C DE LA GESTIÓN DE INFORMACIÓN: PARTE II LA BÚSQUEDA AVANZADA. [En línea] 2010. http://www.cpicmha.sld.cu/hab/Vol16_1_10/hab19110.html.
7. Campos Cosme, Anny y Rodríguez Ronquillo, Reynier Alejandro. Informatización de la Historia de Salud Familiar de la Atención Primaria de Salud. Universidad de las Ciencias Informáticas. La Habana, 2010. 80 Páginas.
8. casidiablo.net|Java, Linux y Programación. JBoss Seam Framework + Laboratorio de iniciación. [En línea] 2009. <http://casidiablo.net/jboss-seam-framework-introduccion/>.
9. CAVSI. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? . [En línea] [Citado el: 12 de Enero de 2011.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

10. Ciberaula. ¿Qué es Java? [En línea] [Citado el: 11 de Enero de 2011.]
http://java.ciberaula.com/articulo/que_es_java/.
11. Colectivo de Autores. Habana 2001. Sistema Informático para la Gestión Médica en la APS (APUS). [En línea] [Citado el: 17 de Enero de 2011.] <http://www.hab2001.sld.cu/arrepdf/00196.pdf>.
12. Cubasolidaridad. Cuba avanza en informatización de sistema de salud . [En línea]
http://www.cubasolidaridad.org/index.php?option=com_content&view=article&id=174:cuba-avanza-en-informatizacie-sistema-de-salud&catid=13:noticias.
13. Definición.de. Definición de Parentesco. [En línea] [Citado el: 22 de Enero de 2011.]
<http://definicion.de/parentesco/>.
14. Desarrollo Fácil. ¿Qué es el paradigma MVC (Modelo, Vista, Controlador)? [En línea] 2010. [Citado el: 10 de Enero de 2011.] <http://www.desarrollofacil.com/que-es-el-paradigma-mvc-modelo-vista-controlador/>.
15. Dirección de Informática del MINSAP. PROGRAMA DE INFORMATIZACION DEL SECTOR DE LA. [En línea] 2003. [Citado el: 22 de ENERO de 2011.]
<http://www.di.sld.cu/estrategias/Programa%20de%20Informatizacion.pdf>.
16. Dorado, M. P. Gaspar; Loria Castellanos, J.; Gerrero Martínez, M. B. SISTEMA DE INFORMACIÓN CIENTÍFICA REDALYC. Aptitud clínica hacia el estudio integral de la familia en residentes de medicina familiar. [En línea] 2006. <http://redalyc.uaemex.mx/pdf/507/50780305.pdf>.
17. Entorno Virtual de Aprendizaje. Introducción a la Disciplina de Requisitos de RUP. [En línea] 2011. [Citado el: 10 de Marzo de 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_06/Conf_7/Materiales_complementarios/Introduccion_a_la_Disciplina_de_Requisitos.pdf.
18. Garbaje Colletor. Sistema Gestor de base de datos SGBD. [En línea] 2004.
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.

19. González, Héctor. javaHispano Tu lenguaje, tu comunidad. Manual Hibernate. [En línea] 2003.
http://www.javahispano.org/contenidos/es/manual_hibernate/.
20. Gutiérrez, Javier J. Universidad de Sevilla. ¿Qué es un framework web? [En línea] [Cited: Enero 10, 2011.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
21. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de software. Madrid: Pearson Educación.S.A, 2000.
22. Junta de Andalucía. Hibernate. [En línea] [Citado el: 11 de Enero de 2011.]
<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Hibernate>.
23. Junta de Andalucía. RichFaces. [En línea] [Citado el: 10 de Enero de 2011.]
<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces>.
24. King, Gavin, y otros. Hibernate. Tutorial basico de Hibernate. [En línea] 2004.
<http://www.davidmarco.es/tutoriales/hibernate-reference/>.
25. La red para los profesionales. ¿Qué es Jboss Seam? [En línea] 2008.
<http://es.debugmodeon.com/articulo/que-es-jboss-seam>.
26. Le Sánchez, Lien y Rosado Roselló, Reinaldo. alasARBOLGEN: aplicación informática para la representación de árboles genealógicos. Universidad de las Ciencias Informáticas. La Habana, 2008. 104 Páginas
27. Lenguajes de Programación. Programación Java. [En línea] 2009. <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
28. Luna Morales, Estela Cristina Dra. TALENTO HUMANO PARA SISTEMAS DE SALUD BASADOS EN LA APS: Perfiles, Formación y Competencias. [En línea] 2009.
<http://guajiros.udea.edu.co/fnsp/congresosp6/memorias6/Viernes%2012/>.
29. Manuales de Ayuda. Características de PostgreSQL. [En línea]
<http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/caracteristicas-de-postgresql-01844.html>.

30. Marquès Puig, Joan Manuel, Vilajosana Guillén, Xavier y García López, Pedro A. Arquitecturas, Paradigmas y aplicaciones de los sistemas distribuidos. [En línea]
[http://docencia.izt.uam.mx/sgm8/nos/sistemas distribuidos 01.pdf](http://docencia.izt.uam.mx/sgm8/nos/sistemas_distribuidos_01.pdf).
31. Montoya, Liliana Guerrero y García Vargas, Alexis. Módulo Medicina Familiar para Escritorio del Sistema Integral para la Atención Primaria. Universidad de las Ciencias Informáticas. La Habana, 2010. 82 Páginas.
32. Mouriz Coca, Yandira y Gonzáles Cruz, Maité. RevistaCiencia.com. Cuba, las TICs y el Bloqueo. [En línea] 2006. <http://www.revistaciencias.com/publicaciones/EEZVpElyyplXUCmqSb.php>.
33. My Family Health Portrait. Sobre Mi Retrato de Salud Familiar. [En línea] 2009.
<https://familyhistory.hhs.gov/fhh-web/popup/getHelp/footerDetailsAboutThisSite.action>.
34. Nájera, Gilberto. mailxmail.com. Desarrollo de aplicaciones para dispositivos inalámbricos (J2ME). [En línea] 2055. <http://www.mailxmail.com/curso-desarrollo-aplicaciones-dispositivos-inalambricos-j2me/lenguaje-programacion-java>.
35. Organización Mundial de la Salud. ¿Qué es un sistema de salud? [En línea] 2005. [Citado el: 13 de Enero de 2011.] <http://www.who.int/features/qa/28/es/index.html>.
36. Paz Noa, Eliane. Diseño y aplicación de pruebas al producto “Árbol Genealógico”. Universidad de las Ciencias Informáticas. La Habana, 2007. 113 Páginas.
37. Pontificia Universidad Javeriana. PATRONES GRASP. [En línea]
http://sophia.javeriana.edu.co/~lcdiaz/ADOO2006-3/grasp_cpaternostro-lvargas-jviafara.pdf.
38. PostgreSQL-es. Sobre PostgreSQL. [En línea] 2010. [Citado el: 12 de Enero de 2011.]
http://www.postgresql-es.org/sobre_postgresql.
39. Presno Labrador, Clarivel Dra. Revista Cubana de Medicina General Integral. El médico de familia en Cuba. [En línea] 2006. http://bvs.sld.cu/revistas/mgi/vol20_5-6_04/mgi135_604.htm.

40. Programa de Atención Primaria de Salud. [En línea] 2009.
<http://revolucioncubana.cip.cu/logros/modelo-social-socialista/salud/programa-de-atencion-primaria-de-salud>.
41. Real Academia Española. Diccionario de la Lengua Española. [En línea] 2008. [Citado el: 2011 de Enero de 27.] <http://drae2.es/símbolo>.
42. Rivera López, Alejandro. UDLAP. Sistema asistente para la generación de horarios de cursos. [En línea] 2008. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf.
43. Sáez Martínez, José, Molina, García Jesús y Jiménez García, Pedro J. Universidad de Murcia. Una Arquitectura para una Herramienta. [En línea] <http://www.dis.um.es/~jmolina/arquipatronesjis99.pdf>.
44. Salinas Caro, Patricio. Universidad de Chile. Unified Modeling Language. [En línea] [Citado el: 13 de Enero de 2011.] <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
45. Seam City. Qué es JBoss Seam? [En línea] [Citado el: 11 de Enero de 2011.] <http://seamcity.madeinxpain.com/archives/que-es-jboss-seam>.
46. Sommerville, Ian. Ingeniería de Software. Madrid: Pearson Educación, 2005. pág. 712.
47. Suárez González, Héctor. Java Hispano. Manual Hibernate. [En línea] 2003. [Citado el: 11 de Enero de 2011.] <http://www.javahispano.org/contenidos/archivo/77/ManualHibernate.pdf>.
48. Taylor, Robert B. Medicina de Familia Principios y prácticas. [En línea] 2006.
http://books.google.com.cu/books?id=d3A0gNxXoVQC&pg=PA5&dq=medicina+familiar&hl=es&ei=9VL4TNcOw4qcB7TlxOwD&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCYQ6AEwAA#v=onepage&q=medicina%20familiar&f=false.
49. Theo, Otto. Desarrollo en Web. Facelets y JSF – Uso de Templates. [En línea] 2008. [Citado el: 11 de Enero de 2011.] <http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>.
50. TLDP-ES/LuCAS. Modelado de Sistemas com UML. [En línea]
<http://lucas.hispalinux.es/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.

51. Universidad de Jaén. GESTORES DE BASES. [En línea]
<http://wwwdi.ujaen.es/~barranco/publico/ofimatica/tema7.pdf>.

52. Viveros Serrano, Carlos Andrés. UDLAP. Sistema generador de aplicaciones Web configurables para el registro en línea de congresos utilizando JavaServer Faces. [En línea] 2006. [Citado el: 10 de Enero de 2011.]
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/viveros_s_ca/capitulo3.pdf.

Glosario de Términos

Aplicación o Sistema Informático: Programas con los cuales el usuario final interactúa a través de una interfaz y que realizan tareas útiles para éste.

Cliente Servidor: Modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información.

Código Alfanumérico: Código que contienen datos alfabéticos (letras) y caracteres especiales (tales como símbolos de puntuación y matemáticos) en adición a números.

Componente: Parte física y reemplazable de un sistema que se ajusta a, y proporciona la realización de, un conjunto de interfaces.

Concurrencia: Ejecución simultánea de dos o más actividades durante el mismo intervalo de tiempo.

Deficiencia: Es toda pérdida o anomalía de una estructura o función psicológica, fisiológica o anatómica.

Dependencia: Relación semántica entre dos elementos, en la cual un cambio en uno puede afectar al otro.

Dominio: Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminología comprendidos por los practicantes de ese dominio.

Equipos Básicos de Salud: Binomio conformado por el médico y enfermera de la familia, que atiende una población geográficamente determinada, que puede estar ubicado en la comunidad, centros laborales o educacionales.

Informática: Disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

GLOSARIO DE TÉRMINOS

Informatizar: Proceso de aplicar sistemas o equipos informáticos al tratamiento de la información.

Internet: Método de interconexión de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red (lógica) única.

Interoperabilidad: Condición necesaria para que los usuarios (humanos o mecánicos) tengan un acceso completo a la información disponible. Entre las iniciativas recientes más destacadas para dotar a la Web de interoperabilidad se encuentran los servicios Web y la Web semántica.

Paquete: Mecanismo de propósito general para organizar elementos en grupos.

Policlínico: Unidad de salud donde se brindan servicios médicos a una población geográficamente determinada perteneciente al nivel asistencial de Atención Primaria de Salud.

Servicio Web: es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Servicio: Unidad de software que encapsula alguna funcionalidad de negocio y proporciona estas a otros servicios a través de interfaces públicas bien definidas.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Software: Conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

Subsistema: Agrupación de elementos, de los que algunos constituyen una especificación del comportamiento ofrecido por los elementos contenidos.

Unidad de Salud: Centro de trabajo que pertenece al Ministerio de Salud Pública (MINSAP).

Anexos

Anexo 1. Interfaz registrar relaciones.

Registrar Relaciones
Q Buscar...

Seleccione el tipo de relación

Relaciones Interpersonales
 Relaciones Legales

Miembros de la Familia

Nombre(s)	Apellidos	
Johan	Leon Garces	↓
Humbertico	Ruiz Salcedo	↓
SandraS	Adulto Mayor	↓
Martha	Estrada Jimenez	↓
johander	leon garces	↓

⏪ 1 ⏩

1 - 5/6

Martha Estrada Jimenez

Nombre(s)	Apellidos	
Johan	Leon Garces	↓
Humbertico	Ruiz Salcedo	↓
SandraS	Adulto Mayor	↓
johander	leon garces	↓
YoilerK	Frometa Joaquin	↓

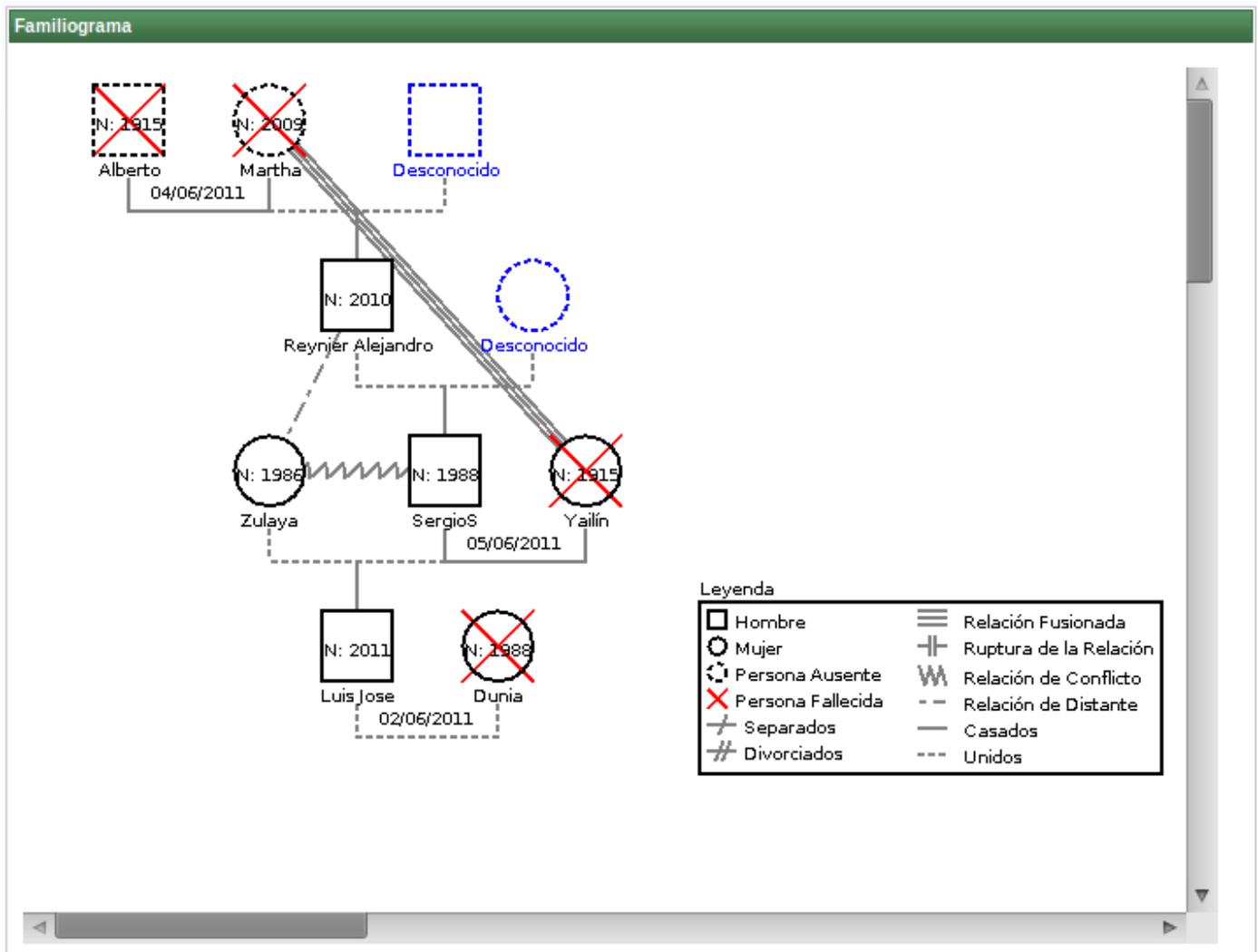
+ Agregar

Listado de relaciones interpersonales entre Martha Estrada Jimenez y johander leon garces

Tipo de relación interpersonal	Observación		
Ruptura Relación	ghgg	✍	🗑
Distante	gffff	✍	🗑

Salir

Anexo 2. Interfaz mostrar famiogramas.



Salir