



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7

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

Componente web para el módulo Salud Materno-Infantil del
Sistema Integral para la Atención Primaria de Salud

Autores: Yenier Vega Rodríguez
Yoel Pelegrín Nicot

Tutores: Ing. Mairenys Mendoza Santana
Ing. Yaney Gómez Domínguez

La Habana, junio de 2011

“Año 53 de la Revolución”

«Queremos

que tengan el máximo de conciencia de su papel, de lo que pueden hacer por su país, de lo que pueden hacer por la Revolución, de lo que pueden hacer por su futuro.»

Fidel.

Datos de Contacto

Ing. Mairensys Mendoza Santana (mmendoza@uci.cu): Graduada de Ingeniería en Ciencias Informáticas en el año 2008 en la Universidad de las Ciencias Informáticas (UCI). Presenta la categoría docente de Instructor Recién Graduado. Ha impartido la asignatura Segundo Perfil (Software Libre para 1er año) y pertenece al Departamento de Atención Primaria de Salud del Centro de Informática Médica (CESIM). Se ha desempeñado como responsable de la Práctica Profesional de 3ro a 5to año en el departamento, como jefa del equipo de desarrollo del módulo Configuración y como Jefa del equipo de desarrollo del módulo Salud Materno-Infantil.

Ing. Yaney Gómez Domínguez (ygomezd@uci.cu): Graduada de Ingeniería en Ciencias Informáticas en el año 2008 con título de oro, en la Universidad de las Ciencias Informáticas (UCI). Presenta la categoría docente de Instructor Recién Graduado. Ha impartido la asignatura de Ingeniería de Software para 3er año y Preparación para Prueba de Nivel de Ingeniería de Software para 5to año. Pertenece al Departamento de Atención Primaria de Salud del Centro de Informática Médica (CESIM). Se desempeña como Analista Principal del departamento y jefa del equipo de Procesos. Ha participado en diferentes eventos científicos del centro obteniendo resultados relevantes. Ha cursado varios posgrados y posee publicaciones como parte de su superación profesional.

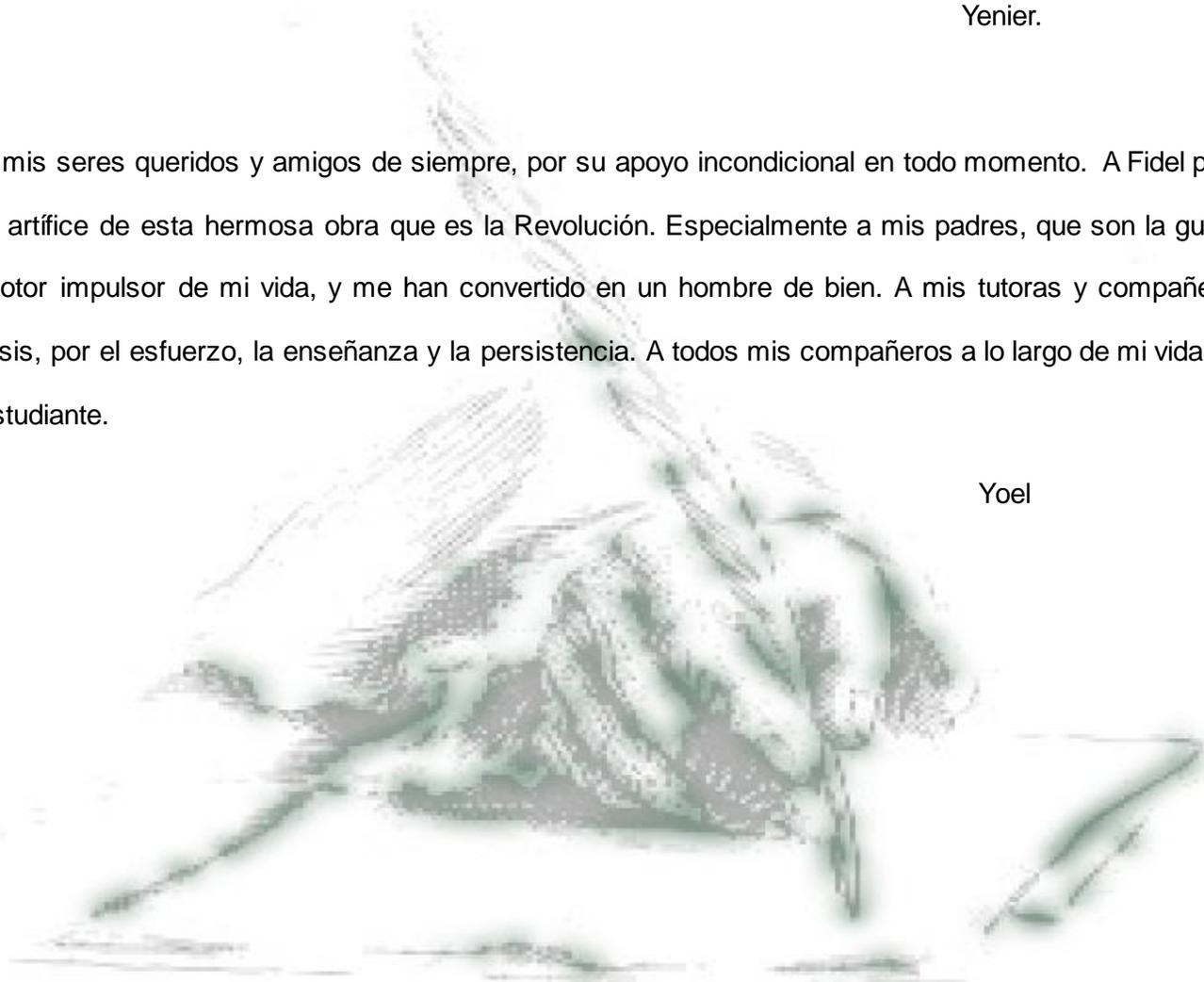
Agradecimientos

A mi familia por el apoyo de siempre, a Fidel por su Revolución y la oportunidad de hacerla nuestra, a todos los educadores que han hecho de mí un hombre de bien, en especial a mis padres y las tutoras de este trabajo de diploma. A mi hermano por el ejemplo, por ser la proa de mi barco en este constante navegar hacia el futuro. A los amigos que he tenido y tengo, entre ellos, a mi compañero de tesis.

Yenier.

A mis seres queridos y amigos de siempre, por su apoyo incondicional en todo momento. A Fidel por ser el artífice de esta hermosa obra que es la Revolución. Especialmente a mis padres, que son la guía y el motor impulsor de mi vida, y me han convertido en un hombre de bien. A mis tutoras y compañero de tesis, por el esfuerzo, la enseñanza y la persistencia. A todos mis compañeros a lo largo de mi vida como estudiante.

Yoel



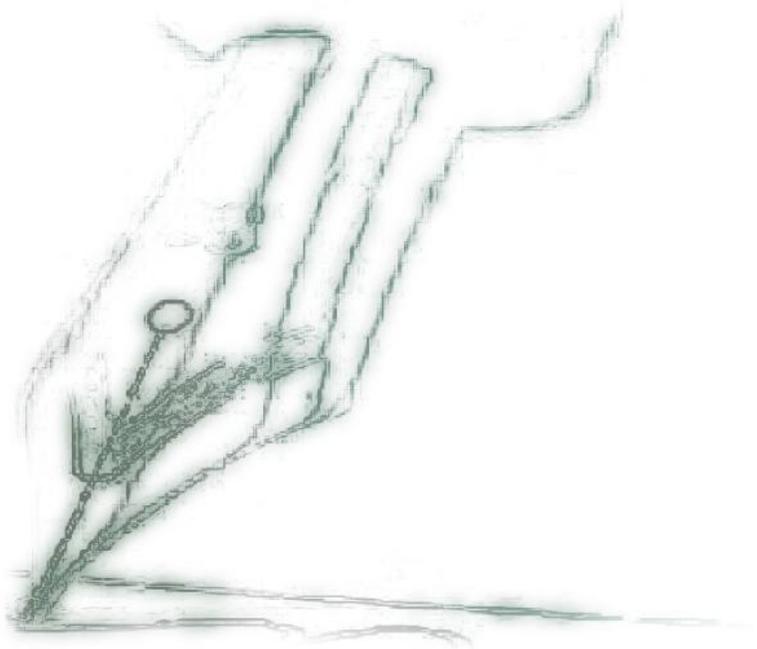
Dedicatoria

A mis padres por el amor y la dedicación con que me han educado, a mi hermano por ser amigo y guía, a Fidel por la oportunidad de poder flotar sobre mi tiempo sin vanagloria ni egoísmo,

Yenier.

A mis padres y hermanos, por la formación, la confianza y el amor de siempre. A mis abuelos, por el cariño infinito. A mi paradigma, Fidel, por regalarnos el futuro.

Yoel.



Resumen

La presente investigación forma parte de los esfuerzos por aplicar la Informática al Sistema Nacional de Salud (SNS) en Cuba. Su objetivo es desarrollar un componente web perteneciente al módulo Salud Materno-Infantil del Sistema Integral para la Atención Primaria de Salud (SIAPS), que facilite la gestión de la información asociada a los procesos captación de la embarazada y el recién nacido, ingreso y atención prenatal de la gestante en el hogar materno, dentro del Programa Nacional de Atención Materno-Infantil (PAMI).

Para ello se utilizó PostgreSQL 8.4 como Sistema Gestor de Base de Datos, Eclipse Ganymede 3.4.2 como Entorno Integrado de Desarrollo, ER/Studio 8.0 para el diseño de la base de datos, Hibernate 3.3 como herramienta para el mapeo objeto-relacional, Java como lenguaje orientado a objetos y multiplataforma y JBoss 4.2.2 como servidor de aplicaciones, entre otras herramientas y tecnologías.

Durante la investigación se hace referencia a los principales conceptos asociados al dominio del problema, se describen los procesos del negocio y se tratan los aspectos relacionados con el diseño y la implementación del componente web. Se empleó la observación y la experimentación como métodos de la investigación científica así como la entrevista a especialistas de la medicina, para conocer las especificidades de los procedimientos.

La identificación del flujo de negocio asociado a cada una de las áreas relacionadas con el campo de acción de la investigación, permitió caracterizar detalladamente el objeto de automatización. El conjunto de tecnologías, arquitectura y patrones de diseño empleados, facilitaron la obtención de la solución.

Palabras Claves: atención prenatal en hogar materno, captación de embarazada y recién nacido, componente web para gestión de información en salud.

Tabla de Contenidos.

Introducción.....	1
Capítulo 1. Fundamentación teórica	7
1.1 Sistema Nacional de Salud	7
1.2 Sistema Integral para la Atención Primaria de Salud	9
1.3 Descripción de la situación problemática.....	10
1.4 Análisis de las soluciones existentes	11
1.5 Tendencias en la industria del software.	15
1.6 Arquitectura de software.	17
1.7 Herramientas.....	23
1.7.2 Sistema Gestor de Base de Datos.....	24
1.7.3 Visual Paradigm	25
1.7.4 Eclipse Ganymede 3.4.2	26
1.7.5 PgAdmin III 1.10.5.....	26
1.7.6 ER/Studio 8.0.....	26
1.8 Frameworks, librerías y componentes.....	27
1.9 Tecnologías.....	32
1.10 Guía para el desarrollo del software.	35
1.11 Proceso de negocio	38
Capítulo 2. Características del Sistema	40
2.1 Flujo actual de los procesos involucrados en el campo de acción.....	40
2.2 Objeto de automatización.....	42
2.3 Modelado del negocio.....	43
2.4 Propuesta del sistema.....	47
Requerimientos de hardware	51
Capítulo 3. Diseño del Sistema.....	53
3.1 Patrones de diseño	53
3.2 Modelo de Datos.....	55
3.3 Modelo de Diseño.....	56
Capítulo 4. Implementación.....	68
4.1 Integración con otros sistemas.....	68
4.2 Implementación.....	69
4.3 Diagrama de Despliegue.....	69
4.4 Estándares de codificación y tratamiento de excepciones.....	70
Conclusiones.....	74
Recomendaciones	75
Referencias Bibliográficas	76
Bibliografía.....	79
Glosario de Términos	84
Anexos	85

Introducción

La atención al niño y la embarazada en Cuba no ha tenido siempre el mismo comportamiento. Antes del año 1959 no existía un programa encargado de asegurar niveles adecuados de salud para la mujer en etapa de gestación ni para el recién nacido. El parto era atendido en muchos casos por comadronas, fundamentalmente en las zonas rurales; los indicadores de mortalidad infantil y materna se comportaban como los propios de un país subdesarrollado. La nación no contaba con instituciones ni con el personal especializado para cubrir las necesidades de salud de la población de manera gratuita.

Con el triunfo de la Revolución Cubana se produjo un cambio radical en la Atención Primaria de Salud (APS). Entre las primeras acciones desarrolladas se encontraron las dirigidas a la atención materno-infantil por su sensible impacto en la población. La implementación y consolidación del Programa Nacional de Atención Materno-Infantil (PAMI) en la década de 1980-1989, le impregnó a las acciones desarrolladas en la atención al niño y la embarazada, la más alta prioridad y se ha convertido en uno de los principales baluartes de las conquistas de la sociedad cubana. Existen indicadores que así lo demuestran. La mortalidad infantil ha disminuido en una progresión casi lineal desde cifras de alrededor de 60 por cada mil nacidos vivos a finales de la década de 1950-1959 hasta 4,8 al cierre del año 2009. Los índices de bajo peso al nacer (BPN) y de mortalidad materna y fetal han mostrado también, una tendencia francamente descendente. [1]

Diversos estudios demuestran que el comportamiento de los indicadores de salud está muy relacionado con las condiciones de vida de la población. Desde el propio triunfo de la Revolución, tales condiciones fueron mejoradas y disminuida la brecha que en calidad de vida, existía entre las zonas rurales y urbanas. Se produjo entonces, un salto de calidad en lo que a la salud del niño y la gestante se refiere, sin todavía existir un conjunto de acciones coordinadas y dirigidas específicamente a estos dos sectores tan sensibles de la sociedad.

Con la creación del Sistema Nacional de Salud (SNS) y su principio de que la salud es derecho del pueblo y responsabilidad del Estado, la atención médica de carácter gratuito cubrió a toda la población, lo que condujo a un fortalecimiento del sistema de salud, de su nivel primario de atención y de su programa materno-infantil.

El PAMI comprende la captación temprana y el seguimiento de la embarazada durante la etapa prenatal y posnatal así como la captación del recién nacido. Los controles prenatales consisten en un conjunto de

actividades que se realizan con la embarazada a fin de asegurar el mejor estado de salud para ésta y su hijo. El objetivo del control prenatal es vigilar la evolución del embarazo y obtener una adecuada preparación para el parto y la maternidad. Para que el control sea eficiente debe ser precoz o temprano, periódico o continuo, completo o integral, extenso o de amplia cobertura [2].

La información generada debe ser rigurosamente recogida en documentos como la Historia Clínica de Atención Pre y Post-natal y el Carné de la Embarazada; para su constante consulta y estudio. Con la captación temprana del recién nacido se persigue la detección precoz de patologías congénitas, fomentar la lactancia materna y la adquisición de un estado nutricional adecuado, así como educar para la salud y colaborar en el logro de un buen entorno familiar y social.

En Cuba las gestantes que presenten determinados problemas de salud son ingresadas en hogares maternos con el objetivo de brindarles una atención diferenciada y un seguimiento más detallado de su enfermedad. Los hogares maternos se han desarrollado en forma continua por más de treinta años y han contribuido a lograr que todos los nacimientos se realicen en los hospitales. Este ha sido, al menos, su resultado más inmediato y constituyó desde sus inicios, un importante paso de avance que se fue produciendo simultáneamente con la ampliación de los hospitales maternos y servicios ginecobstétricos en los hospitales generales. [3]

Los hogares maternos que se crearon en el siglo pasado en Europa y los Estados Unidos tuvieron una existencia transitoria y de hecho, desaparecieron cuando terminó la centuria. Los actuales no han mostrado esa característica, por el contrario, tienen propensión a su continuidad, al menos es lo que ha sucedido con los de África y América Latina después de más de treinta años de experiencia, de la misma forma que tampoco estos países han podido salir del subdesarrollo, base de la situación socioeconómica que ha originado su creación y sustentado la continuidad de esta institución en ambas regiones. [3]

Durante el ingreso y seguimiento de la embarazada en el hogar materno, se generan varios documentos que recogen con estricta disciplina la información de cada acción de salud que se realiza a la embarazada; la cual resulta de vital importancia para el estudio de cada caso y para la prevención y control de patologías que puedan poner en riesgo la vida de la gestante y el feto.

En la nación cubana y a nivel internacional existen dificultades con la gestión de la información de las actividades de salud materno-infantil en la Atención Primaria de Salud que facilite el seguimiento de la embarazada y el niño tanto en la comunidad como en el entorno familiar. Los cuidados de la mujer durante

su embarazo así como en el posembarazo constituyen un elemento esencial para la prevención de afecciones en la familia. De igual forma la captación temprana del recién nacido garantiza la detección precoz de enfermedades y la prevención de futuros problemas en el crecimiento y desarrollo madurativo del niño en todas sus vertientes. La atención diferenciada con un enfoque integral posibilita diagnosticar, rehabilitar e indicarle tratamientos más certeros al paciente, teniendo en cuenta los antecedentes y el entorno donde convive.

En el proceso de captación de la embarazada que tiene lugar en el consultorio médico de la familia, el volumen de datos que se consulta y registra es considerablemente grande, teniendo en cuenta que se trabaja con documentos tales como la Historia de Salud Familiar, la Historia Clínica de Atención Pre y Post-Natal, la Tarjeta de Vacunación, Tarjeta de Citología, Carné de la Embarazada entre otros. Los procesos de ingreso y seguimiento que ocurren una vez que acude la paciente al hogar materno, tienen características similares al anterior, incorporando además la intervención de diversos roles del personal de salud en el proceso de atención prenatal, con la actualización de disímiles documentos. Durante la captación del recién nacido se genera también una gran cantidad de datos, tanto del niño como de la madre, los cuales se registran en la Historia Clínica del Niño (HCN) y en el Carné de Salud Infantil (CSI).

Se requiere por tanto, que el equipo de salud pueda contar con mecanismos rápidos y eficientes que le permita emplear un mínimo de tiempo y lograr un máximo de efectividad y viabilidad en el proceso de gestión de la información cada vez más amplia, diversa y cambiante.

Lo antes descrito y la necesidad de buscar una solución a los problemas que hacen ineficiente el desarrollo de las actividades en el consultorio médico de la familia así como en el hogar materno, conducen a que se identifique la siguiente situación problemática:

La gestión de la información generada en los procesos de captación del recién nacido y la embarazada, ingreso y atención prenatal de la gestante en el hogar materno, es un procedimiento amplio y complejo que se realiza manualmente, archivándose los datos en papel, lo que propicia el deterioro de la información y la pérdida de la misma. El personal vinculado a estos procesos se ve obligado a obrar con lentitud al no contar con herramientas que apoyen su trabajo en la gestión de la información. Las principales deficiencias encontradas en las actividades que se realizan son:

- Demora en la elaboración y actualización de documentos.
- Posible consulta de información no actualizada o en mal estado.

- Registro de datos de formas muy diversas teniendo en cuenta la participación de varios roles en la actualización de un mismo documento y la carencia de modelos oficiales que estandaricen la estructura de la información.
- Resulta difícil acceder, desde una institución de salud y con la inmediatez que se requiere, a la información que sobre el paciente; posee otra institución de salud.
- La veracidad de la información que se registra en los documentos generados en los procesos, se ve afectada al depender de las respuestas del paciente en el momento del interrogatorio.
- No se cuenta con un acceso rápido y viable a toda la información necesaria en cada momento, para lograr una atención integral al niño y la embarazada.

Por tanto el problema a resolver es: ¿Cómo facilitar la gestión de la información generada en los procesos de captación del recién nacido y la embarazada, ingreso y atención prenatal de la gestante en el hogar materno?

Se determina como objeto de estudio el proceso de gestión de la información del Programa Nacional de Atención Materno Infantil en la Atención Primaria de Salud.

Como objetivo general se plantea: Desarrollar un componente web perteneciente al módulo Salud Materno-Infantil del Sistema Integral para la Atención Primaria de Salud, que facilite la gestión de la información asociada a los procesos de captación de la embarazada, ingreso y atención prenatal en el hogar materno, y captación del recién nacido.

El campo de acción se centra en la gestión de la información en la captación de la embarazada y el recién nacido en el consultorio médico de la familia e ingreso y atención prenatal de la gestante en el hogar materno en la Atención Primaria de Salud.

Para el desarrollo de la investigación se ha empleado el método científico de la observación. En diferentes momentos y en diversas instituciones de salud, con la participación de varias personas, se estudiaron los procesos involucrados; permitiendo la validación de la información a partir de los puntos coincidentes. Mediante la entrevista a especialistas de la medicina en la Atención Primaria de Salud, se conocieron las especificidades de los procedimientos. Se utilizaron guías para garantizar que los aspectos fundamentales fueran descritos correctamente, sin embargo la creatividad requerida por parte del entrevistador para que se conociera en profundidad cada proceso, fue un elemento decisivo en el aseguramiento del rigor de la investigación. La aplicación que se pretende lograr es sometida a diversas pruebas durante todo su

desarrollo para comprobar su correcto funcionamiento. En ello se ve reflejada la experimentación como otro método de la investigación científica.

Las tareas de investigación que se proponen son:

- Caracterizar los procesos de cada una de las áreas de salud relacionadas con la captación de la embarazada y el recién nacido, ingreso y atención prenatal de la gestante en el hogar materno.
- Analizar la bibliografía asociada a las tendencias actuales en Cuba y a nivel internacional, relacionadas con la salud materno-infantil en la Atención Primaria de Salud.
- Obtener mediante la guía de desarrollo propuesta por el proceso de mejora de la universidad, los artefactos correspondientes a las fases: “Modelado del Negocio”, “Requisitos”, “Análisis y Diseño” e “Implementación”.
- Implementar las funcionalidades correspondientes a los procesos propuestos del módulo Salud Materno-Infantil del Sistema Integral para la Atención Primaria de Salud:
 - Captación de la embarazada.
 - Captación del recién nacido.
 - Ingreso de la embarazada en el hogar materno.
 - Atención prenatal de la embarazada en el hogar materno.

El documento está estructurado en cuatro capítulos:

CAPÍTULO 1. Fundamentación Teórica: Se hace referencia a los trabajos realizados y que se vinculan con la temática de la investigación, los principales conceptos asociados al dominio del problema y las características de las tecnologías y herramientas a utilizar en el desarrollo de la propuesta.

CAPÍTULO 2. Características del Sistema: Se refleja la investigación realizada con la descripción de los procesos del negocio y se describe la propuesta del sistema utilizando los requerimientos funcionales y no funcionales que se han identificado.

CAPÍTULO 3. Diseño del Sistema: Se describen los aspectos relacionados con el diseño de la solución propuesta. Para ello se modelan los diagramas de clases y se especifican los patrones para el diseño gráfico de la aplicación. Se tratan los aspectos relacionados con el Modelo de Datos.

CAPÍTULO 4. Implementación: Se argumentan los temas referentes a la construcción de la solución propuesta, se modelan los diagramas de componentes y despliegue, se describen los estándares de diseño y codificación aplicados en la solución del sistema.

Capítulo 1. Fundamentación teórica

Como temáticas a tratar en este capítulo se encuentran los principales conceptos asociados al dominio del problema así como los elementos que fundamentan en teoría, el componente web que se quiere obtener. Se hace referencia a diferentes soluciones informáticas, tanto nacionales como internacionales y que en la actualidad, justifican su existencia en la mejora de procesos relacionados con los que se mencionan en la situación problémica por la cual ha surgido esta propuesta de solución. De igual forma recoge en su contenido las características de las tecnologías y herramientas seleccionadas así como la descripción de la guía utilizada para cumplir de manera óptima el objetivo general que se plantea.

1.1 Sistema Nacional de Salud

La Organización Mundial de la Salud (OMS) define el Sistema Nacional de Salud (SNS) como el conjunto de recursos físicos, financieros y humanos de 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 las agravantes. [4]

En Cuba, el Sistema Nacional de Salud es guiado por el Ministerio de Salud Pública (MINSAP) que es el organismo encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado referente a la salud del pueblo. Este ministerio cubre las funciones de atención médica, asistencia a ancianos y minusválidos, control higiénico y epidemiológico, formación de profesionales y la producción y distribución de medicamentos.

El Sistema Nacional de Salud de un país se encuentra organizado por niveles de atención con el objetivo de cumplir, con un máximo de cobertura, los servicios que se prestan en una forma eficiente, con calidad y con los recursos con que se cuenta. Estos niveles son:

- **Atención Primaria de Salud (APS):** La atención primaria de la salud es una estrategia que concibe integralmente los problemas de salud–enfermedad y de atención de las personas y el conjunto social. No solo provee servicios de salud, sino que afronta las causas últimas (sociales, económicas y políticas) de la ausencia sanitaria. En su visión integra la asistencia médica, la prevención de enfermedades, la promoción de la salud y la rehabilitación. Su misión consiste en extender el sistema sanitario hasta la intimidad de las viviendas para conocer la realidad social y sanitaria de la población y mejorar la comunicación entre el individuo y su familia –con sus saberes y cultura– y la medicina científica. [5]

Se ha comprobado que con una buena atención primaria se puede satisfacer entre un 80% a un 90% de los problemas de salud que se presentan. En Cuba, la red de policlínicos y consultorios médicos de la familia prestan servicios de salud que pertenecen a la APS.

- **Atención Secundaria de Salud (ASS):** Este nivel da cobertura a cerca del 15 % de los problemas de salud. Su función fundamental es tratar al hombre ya enfermo pero también desempeña funciones de rehabilitación, promoción y prevención de la salud. Comprende la atención de naturaleza especial, generalmente compleja y que no puede ser realizada por el médico general. Ocurre fundamentalmente en los hospitales.
- **Atención Terciaria de Salud (ATS):** Otorga prestaciones de mayor complejidad, tanto en atención cerrada como ambulatoria. El nivel terciario debe abarcar alrededor del 5 % de los problemas de salud relacionados con secuelas o aumento de las complicaciones de determinadas dolencias. Se brindan servicios de muy alta complejidad con la óptima utilización de los recursos y medios existentes en los mismos y el desarrollo de la investigación. A este nivel pertenecen los institutos y hospitales especializados.

Desde los comienzos de la Revolución, en Cuba se presta especial atención a los niños y las embarazadas a través de diversas políticas, estrategias y acciones a favor de su salud. Dentro del SNS se encuentra el Programa Nacional de Atención Materno Infantil concebido para garantizar niveles adecuados de salud en estos dos sectores tan sensibles de la población. En su concepción actual, el PAMI está conformado por un conjunto de programas, entre los que se pueden mencionar:

- Programa de Promoción de la Lactancia Materna.
- Programa para la Reducción del Bajo Peso al nacer.
- Programa para el Desarrollo de la Perinatología.
- Programa de Lucha contra las Enfermedades Diarreicas Agudas y las Infecciones Respiratorias Agudas.
- Programa Nacional de Inmunizaciones.
- Programa de Prevención y Diagnóstico Precoz de Enfermedades Genéticas.
- Programa Nacional para la Prevención de Accidentes en menores de 20 años.
- Programa de Maternidad y Paternidad Conscientes.

- Programa de Prevención y Control de los Síndromes Neurológicos Infecciosos.
- Plan de Control del Seguimiento del Crecimiento del Menor de 5 años.

El Consultorio Médico de Familia (CMF) es una institución de salud directamente involucrada en los procesos de captación de la embarazada y el recién nacido mientras que el Hogar Materno es escenario para el ingreso y atención prenatal de la gestante siempre que se considere necesario. Los consultorios médicos de familia se crearon con el objetivo de brindarles una atención primaria a los pacientes en cada barrio o comunidad. Fueron fundados en el año 1990 como parte de la emancipación del sistema de salud a lo largo de todo el país, lo cual constituye uno de los logros más importantes de la Revolución Cubana. Los hogares maternos se crearon con el objetivo fundamental de elevar el número de partos en las instituciones de salud y ya han cumplido en Cuba, más de tres décadas de existencia.

1.1.1 Informatización del Sistema Nacional de Salud

La informatización del SNS está dada por el conjunto de métodos, técnicas, procedimientos y actividades gerenciales dirigidas al manejo de la información en salud, la cual comprende la información sobre el estado de salud de la población, la información sobre el conocimiento de las ciencias de la salud y la información en general para la toma de decisiones clínico-epidemiológicas, operativas y estratégicas. [6]

Como parte del proceso de informatización de la sociedad cubana, el SNS ocupa un puesto crítico, por lo que se han destinado cuantiosos recursos materiales y humanos al desarrollo de herramientas que cumplan con los objetivos trazados. En el año 1997 se concibió una primera estrategia de informatización, la cual ha tenido siempre al paciente como centro del proceso, por ser este, el principal beneficiado con la garantía de calidad, oportunidad y consistencia de la información que logren las aplicaciones. El incremento de la efectividad y eficiencia de los procesos relacionados con la salud persigue el crecimiento continuo y sostenido de la calidad en la atención médica.

1.2 Sistema Integral para la Atención Primaria de Salud

El Sistema Integral para la Atención Primaria de Salud (SIAPS) se concibe para la integración de los componentes, servicios o sistemas que se desarrollen para la informatización de la APS así como para atender las necesidades de los clientes del nivel primario de atención de salud como plataforma única para la gestión, procesamiento y transmisión de la información clínica en ese nivel. Cuenta con un conjunto de módulos identificados con las diferentes áreas de procesos que conforman el negocio de la atención primaria. Algunos de estos módulos son:

- **Módulo de Configuración del SIAPS:** Gestiona todas las configuraciones de temas, roles, usuarios, funcionalidades, módulos, unidades de salud y áreas de salud, personal de salud, ubicación, gestión de codificadores o estándares internacionales, nomencladores médicos, las poblaciones o zonas de APS, así como la seguridad del sistema.
- **Módulo de Medicina Familiar del SIAPS:** Gestiona los procesos básicos y fundamentales de la APS. Es el núcleo del sistema y contiene los datos generales del paciente, Historia Clínica Individual, Historia Clínica Familiar, la planificación, seguimiento y control del paciente, los ingresos en el hogar y seguimiento diario, las remisiones, la dispensarización del paciente o clasificación en grupos homogéneos para la atención diferenciada del paciente en la Atención Primaria de Salud.
- **Módulo de Medios Diagnósticos del SIAPS:** Gestiona la información de los laboratorios clínicos y Parasitología, Microbiología, Imagenología, Endoscopía, Drenaje Biliar, Electrocardiografía, Laboratorio de Alergia, Optometría y Audiometría en la Atención Primaria de Salud.
- **Módulo Clínico Quirúrgico del SIAPS:** Gestiona la información relacionada con la planificación, control y seguimiento de las especialidades médicas: Nefrología, Medicina Interna, Cardiología, Angiología, Urología, Dermatología, Endocrinología, Hematología, Reumatología, Otorrinolaringología, Ortopedia, Neurología, Genética, Cirugía Menor, Cirugía Menor Ambulatoria, Cirugía Mayor, Quimioterapia, Atención Integral al Paciente Diabético y Oftalmología en la atención primaria.
- **Módulo de Enfermería del SIAPS:** Gestiona la información asociada al proceso de vacunación, esterilización, las especialidades de Enfermería y centros comunitarios de Enfermería en la Atención Primaria de Salud.

Otros módulos como Asistencia Social y Adulto Mayor, Higiene y Epidemiología, Salud Mental y Salud Materno-Infantil se corresponden con nuevos desarrollos en el actual período. Este último se refiere al área de procesos que cubre la captación de la embarazada, su atención prenatal en el hogar materno en caso de ingreso, y su seguimiento posnatal. De igual forma trata la captación del recién nacido.

1.3 Descripción de la situación problemática.

Al analizar el procedimiento actual para la gestión de la información que se genera en los procesos de captación de la embarazada y el recién nacido, ingreso y atención prenatal de la gestante en el hogar

materno dentro de la Atención Primaria de Salud, se encontraron como principales deficiencias las siguientes:

- La información obtenida es registrada manualmente y en copia impresa.
 - El proceso de elaboración y actualización de documentos se ve afectado por constantes demoras.
 - La información se expone al deterioro y a la consecuente pérdida.
 - Posible consulta de documentos no actualizados o en mal estado.
- Registro de la información de formas muy diversas teniendo en cuenta la participación de varios roles en la actualización de un mismo documento y la carencia de modelos oficiales que estandaricen la estructura de la información.
 - No se garantiza que siempre se recoja la información necesaria en los documentos.
 - Pueden existir malinterpretaciones o no entendimiento de la información consultada.
- Resulta difícil acceder, desde una institución de salud y con la inmediatez que se requiere, a la información que sobre el paciente; posee otra institución de salud.
 - No se puede consultar toda la información que se necesita, con la inmediatez que cada situación demanda.
- Se requiere de información cada vez más amplia y confiable para alcanzar niveles superiores de calidad en la atención prenatal y posnatal de la gestante y en el seguimiento del niño hasta la adolescencia. Esta necesidad no se satisface con las características actuales del proceso de gestión de la información en el PAMI.

1.4 Análisis de las soluciones existentes

Los sistemas informáticos para la gestión de la información en los centros de la atención primaria buscan ofrecer al equipo de salud y directivos, la información necesaria, oportuna y útil para la toma de decisiones que ayuden a elevar el nivel de salud de la población bajo su atención. Igualmente ofrecen 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. Por ello resulta muy importante que dentro de estos sistemas exista un componente dedicado a la gestión de la información relacionada con el

niño en sus etapas perinatal y posnatal y la embarazada en los períodos prenatal y posnatal. Estos sectores de la población serían altamente beneficiados con las ventajas que ofrecen los sistemas informáticos.

1.4.1 Sistema Informático para la Atención Primaria de Salud (APUS)

País de origen: Cuba

Este sistema fue desarrollado por el Centro de Desarrollo Informático para la Salud Pública (CEDISAP) para la automatización de la información de un centro de salud de la atención primaria y se encuentra ajustado a las normas nacionales e internacionales para su utilización en servicios de salud. Brinda información necesaria, oportuna y útil para:

- Tomar decisiones en todos los niveles de dirección.
- Analizar la situación de salud.
- Registrar, controlar y evaluar los servicios brindados y la productividad médica.
- Monitorear y evaluar los programas.
- Apoyar la vigilancia en salud.
- Elaborar los indicadores e información estadística necesaria para evaluar los resultados en término de salud.

El producto fue elaborado bajo el modelo Cliente/Servidor con el empleo de Delphi. Fue utilizado el sistema gestor de base de datos SQL (*Structured Query Language* por sus siglas en Inglés) *Server*. Se programó además en ambiente web utilizando la tecnología ASP (*Active Server Page* por sus siglas en inglés) basada en el servidor y la herramienta Macromedia Dreamweaver UltraDev para el diseño de las páginas web con acceso a base de datos. Para el diseño de la base de datos se utilizó la herramienta CASE (*Computer Aided Software Engineering* por sus siglas en Inglés) Erwin/ERX. [7]

Características de APUS:

- Fácil instalación y operación.
- Sistema de seguridad que garantiza la protección y confiabilidad de la información.
- Uso de codificadores de interés local y general.

- Rapidez en la captación y procesamiento de los datos.
- Eficiente búsqueda de información.
- Análisis visual de la información mediante gráficos.

Esta herramienta fue rechazada como posible cumplimiento del objetivo propuesto, por las siguientes razones:

- No permite la integración con el SIAPS.
- Aunque gestiona información perteneciente a la APS, no contempla las funcionalidades necesarias para la gestión integral de los procesos pertenecientes al campo de acción de la presente investigación.

1.4.2 Sistema de Administración y Gestión de Clínicas y Hospitales (Lolcli 2000)

País de origen: Perú

El sistema se ha desarrollado para el registro estandarizado, completo y adecuado de todas las actividades que se realizan en relación con el paciente, desde que ingresa hasta que abandona el establecimiento de salud. Este modelo de gestión está basado en los registros que se hacen en la historia clínica; es decir, en los registros de los actos médicos.

Está desarrollado en dos sistemas gestores de base de datos: SQLServer y Oracle y diseñado en modo gráfico con entorno Windows (Power Builder) y con múltiples usuarios de acceso simultáneamente.

Esta herramienta fue rechazada como posible cumplimiento del objetivo propuesto, por las siguientes razones:

- Aunque el sistema gestiona la información relacionada con los procesos de ingreso y seguimiento de los pacientes en las instituciones de salud, no lo hace específicamente ni de forma completa, para las embarazadas en el hogar materno. No gestiona la información relacionada con el proceso de captación de la embarazada ni del recién nacido.
- No permite la integración con el sistema SIAPS.
- Al realizarse sobre plataforma Windows no se puede acceder al código para su modificación.

1.4.3 ALERT® PRIMARY CARE.

País de origen: Brasil

Es un *software* para la informatización de los Centros de Salud, que permite la documentación y revisión del historial individual de cada paciente, incluyendo información de otras instituciones. [8]

Principales procesos que gestiona y servicios que brinda:

- Documentación para la consulta
- Abordaje SOAP (*Simple Object Access Protocol* por sus siglas en inglés)
- Listado de problemas
- Gestión de la vacunación
- Prueba de la tuberculina
- Evaluación pediátrica
- Observación periódica pediátrica y de adultos
- Acompañamiento de planes de salud
- Consultas de enfermería
- Factores y evaluaciones de riesgo
- Lista de tareas
- Información organizada por individuo y por familia

Esta herramienta fue rechazada como posible cumplimiento del objetivo propuesto, por las siguientes razones:

- No gestiona de forma íntegra la información de los procesos de captación de la embarazada, ingreso y atención prenatal en el hogar materno.
- Es un software privativo que impide el acceso al código para su modificación.

1.4.4 PROGRAMA ADUANA-SETIEMBRE.

País de origen: Argentina

Inicialmente el programa se llamó Aduana y fue creado en Argentina en un área de atención materno-infantil con el objetivo de asegurar el control de recién nacidos y lactantes luego del alta hospitalaria. El programa comenzó a funcionar desde el año 1974 y ha pasado por varias etapas hasta que en el 2002 se unió al programa Setiembre cuyo trabajo estaba enfocado en el seguimiento de las puérperas adolescentes. Es en ese año que toma el actual nombre de Aduana-Setiembre.

El programa se desarrolló bajo los principios de coordinación intrainstitucional e interinstitucional entre los diversos efectores del subsector público. Dicho programa tiene como objetivo:

- Asegurar la captación, control y seguimiento del recién nacido.
- Asegurar el control y seguimiento del lactante.
- Asegurar el control de la puérpera adolescente.
- Mejorar las condiciones de salud de las adolescentes madres, promoviendo el auto cuidado y facilitando el vínculo con los equipos de salud.
- Prevenir un nuevo embarazo no deseado en las adolescentes madres.

Esta herramienta fue rechazada como posible cumplimiento del objetivo propuesto, por las siguientes razones:

- No gestiona toda la información referente a los procesos captación de la embarazada y el recién nacido, ingreso y atención prenatal de la gestante en el hogar materno.
- Solo funciona sobre Windows el cual es un sistema operativo privativo que impide el acceso al código fuente para su modificación.

1.5 Tendencias en la industria del software.

1.5.1 Internet.

Internet es una red de redes que permite la interconexión descentralizada de computadoras a través de un conjunto de protocolos denominado TCP/IP (*Transmission Control Protocol / Internet Protocol* respectivamente por sus siglas en inglés). Tuvo sus orígenes en 1969 cuando una agencia del Departamento de Defensa de los Estados Unidos comenzó a buscar alternativas ante una eventual guerra atómica que pudiera incomunicar a las personas. Tres años más tarde se realizó la primera demostración pública del sistema ideado, gracias a que tres universidades de California y una de Utah lograron

establecer una conexión conocida como ARPANET (*Advanced Research Projects Agency Network* por sus siglas en inglés). [9]

Con el surgimiento de la WWW (*World Wide Web* por sus siglas en inglés) como medio de comunicación en forma de gráficos, textos y otros objetos multimedia que utiliza Internet para el intercambio de información; se hizo imprescindible su uso en las sociedades modernas. La llamada “autopista de la información” constituye una fuente inagotable de conocimientos de la cual es muy difícil prescindir. El Sistema Integral para la Atención Primaria de Salud, como expresión de las tecnologías web, necesita de aquellos grandes descubrimientos para la prestación y expansión de sus servicios.

1.5.2 Software Libre

Se entiende por software libre, la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Esencialmente significa que los usuarios de programas tienen las cuatro libertades esenciales: [10]

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa y adaptarlo a las necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa se considera software libre si los usuarios tienen todas estas libertades. Así pues, se debe tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y en cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no se tiene que pedir o pagar permisos. [10]

Como política del MINSAP y del Centro de Informática Médica (CESIM) se encuentra el enfoque de todas las soluciones informáticas que se desarrollen, hacia el uso de las potencialidades del software libre, aspecto que no limita la comerciabilidad de sus productos y favorece el desarrollo.

1.5.3 Sistema Operativo GNU/ Linux.

GNU/Linux es un sistema operativo surgido de la unión del *kernel* o núcleo, desarrollado por el finlandés Linus Torvalds y un sistema básico casi completo desarrollado por el proyecto GNU (acrónimo recursivo para “Gnu No es Unix”). Tiene su origen en Unix, un sistema operativo potente, flexible, multitarea y multiusuario del cual heredó estas importantes características. GNU/Linux en sus diferentes distribuciones es uno de los sistemas operativos más usados en el mundo y uno de los más prominentes ejemplos de software libre. Todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (*General Public License* por sus siglas en inglés).

1.6 Arquitectura de software.

Es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación y tiene la responsabilidad de:

- Definir los módulos principales
- Definir las responsabilidades que tendrá cada uno de estos módulos
- Definir la interacción que existirá entre dichos módulos:
 - Control y flujo de datos
 - Secuenciación de la información
 - Protocolos de interacción y comunicación
 - Ubicación en el hardware

La arquitectura de software aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los módulos definidos a pasos posteriores del diseño.

La definición oficial de Arquitectura de Software es la IEEE (*Institute of Electrical and Electronics Engineering* por sus siglas en inglés) Std 1471-2000 que reza así: “La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. [11]

1.6.1 Arquitectura Cliente/Servidor.

La arquitectura Cliente/Servidor es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales. En el modelo usual, un servidor se activa y espera las solicitudes de los clientes. Habitualmente, programas clientes múltiples comparten los servicios de un programa servidor común.

Desde el punto de vista funcional, se puede definir el modelo Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma. Se trata pues, de la arquitectura más extendida en la realización de sistemas distribuidos. [12].

Algunas de sus características son:

- Protocolos asimétricos. Los clientes inician “conversaciones”. Los servidores esperan su establecimiento pasivamente.
- Transparencia de localización física de los servidores y clientes. El cliente no tiene por qué saber dónde se encuentra situado el recurso que desea utilizar.
- Independencia de la plataforma hardware y software que se emplee.
- Sistemas débilmente acoplados. Interacción basada en envío de mensajes.
- Encapsulamiento de servicios. Los detalles de la implementación de un servicio son transparentes al cliente.
- Escalabilidad horizontal (añadir clientes) y vertical (ampliar potencia de los servidores).
- Integridad. Datos y programas centralizados en servidores facilitan su integridad y mantenimiento.

Ventajas de la arquitectura Cliente/Servidor:

- El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. El cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.
- El sistema es fácil de escalar.

Desventajas de la arquitectura Cliente/Servidor:

- Gasto. por lo general, el equipo del servidor central debe ser lo suficientemente potente como para mantener y compartir recursos con los demás equipos de la red. Esto implica un costo considerable.
- Dependencia. El modelo Cliente/Servidor se basa en un funcionamiento y un servidor centralizado disponibles. Si el servidor centralizado es eliminado del sistema o se cae por determinado problema, toda la red deja de funcionar.
- Congestión. Servidores centralizados deben manejar la mayor parte del tráfico de red al dirigirse toda la consulta de recursos hacia los mismos. Esto puede causar congestión en la red y disminuir los tiempos de respuesta para cada equipo disponible.
- Mantenimiento. Se requiere al menos de un administrador de red para mantener el correcto intercambio de la información.

1.6.2 Patrón de arquitectura Modelo Vista Controlador.

Modelo Vista Controlador (MVC) 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 ve frecuentemente en aplicaciones *web*, donde la vista es la página HTML (*HyperText Markup Language* por sus siglas en inglés) y el código que provee de datos dinámicos a la página; el modelo contiene los datos y funcionalidades que mantienen la lógica de negocio con sus reglas, y el controlador es el responsable de recibir los eventos de entrada desde la vista y de realizar acciones que representan la respuesta lógica del sistema ante estos eventos. [13]

La finalidad del modelo es mejorar la reusabilidad por medio del desacople entre la vista y el modelo.

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Definir las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: “Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor”.
- Es el encargado de notificar a la vista los cambios que en los datos pueda producir un agente externo.

El controlador es el responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, entre otros).
- Contiene reglas de gestión de eventos, del tipo “Si evento Z, entonces acción W”. Estas acciones pueden suponer peticiones al modelo o a la vista.

La vista es responsable de:

- Recibir datos del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado (además lo instancia).
- Puede prestar el servicio de “Actualización ()” para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Ventajas del patrón de arquitectura Modelo Vista Controlador:

- Es posible tener diferentes vistas para un mismo modelo (Ejemplo: representación de un conjunto de datos como una tabla o un diagrama de barras).
- Es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.
- Proporciona un mecanismo de configuración a componentes complejos mucho más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción).
- Facilita el mantenimiento en caso de errores.
- Permite escalar la aplicación en caso de ser requerido.

Desventajas del patrón de arquitectura Modelo Vista Controlador:

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.

1.6.3 Desarrollo de Software Basado en Componentes (DSBC).

La complejidad de los sistemas computacionales actuales ha llevado a buscar la reutilización del *software* existente. El desarrollo de aplicaciones informáticas basado en componentes permite reutilizar piezas de código preconcebidas que pueden ser reutilizadas en el desarrollo de otras aplicaciones. Según el SEI

(*Software Engineering Institute* por sus siglas en inglés) de la Universidad Carnegie-Mellon, un componente es “una implementación opaca de funcionalidad, sujeta a composición por terceros y que cumple con un modelo de componentes”. Tal como lo refleja la definición anterior, un componente de software puede ser visto desde dos perspectivas distintas:

- Implementación. Los componentes se pueden ensamblar y desplegar para crear sistemas y aplicaciones que se ejecutan en un computador.
- Abstracción de arquitectura. Los componentes expresan las reglas de diseño que impone el modelo de componentes.

El paradigma de ensamblar componentes y escribir código para hacer que estos componentes funcionen se conoce como Desarrollo de Software Basado en Componentes. El uso de este paradigma posee algunas ventajas: [14]

- Permite alcanzar un mayor nivel de reutilización de *software*.
- Simplifica las pruebas. Permite que las pruebas sean ejecutadas al probar cada uno de los componentes y luego el conjunto completo de componentes ensamblados.
- Simplifica el mantenimiento del sistema. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
- Mayor calidad. Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.
- Ciclos de desarrollo más cortos. La adición de una pieza dada de funcionalidad tomará días en lugar de meses o años.
- Mejor ROI (*Return On Investment* por sus siglas en inglés). Usando correctamente esta estrategia, el retorno sobre la inversión puede ser más favorable que desarrollando los componentes.
- Funcionalidad mejorada. Para usar un componente que contenga una pieza de funcionalidad, solo se necesita entender su naturaleza, más no sus detalles internos. Así, una funcionalidad que sería impráctica de implementar en la empresa, se vuelve ahora completamente asequible.

Desventajas del Desarrollo de Software Basado en Componentes: [15]

- Clarividencia. Se refiere a la dificultad con la que se encuentra el diseñador de un componente al realizar su diseño pues no conoce ni quién lo utilizará, ni cómo, ni en qué entorno, ni para qué aplicaciones.
- Evolución de los componentes. La gestión de la evolución es un problema serio, pues en los sistemas grandes han de poder coexistir varias versiones de un mismo componente.
- Percepción del entorno. Es la habilidad de un objeto o componente de descubrir tanto el tipo de entorno donde se está ejecutando como los recursos y servicios disponibles en él. Falta mucho por avanzar aún en este aspecto.
- Particularización. Existe la interrogante de cómo particularizar los servicios que ofrece un componente para adaptarlo a las necesidades y requisitos concretos de una aplicación.

Los modelos y plataformas de componentes proporcionan los mecanismos adecuados para poder tratar la complejidad de los problemas que aparecen en la construcción de aplicaciones para sistemas abiertos y distribuidos, como pueden ser la heterogeneidad, la sincronización, la dispersión, fallos en la seguridad, retrasos y problemas en las comunicaciones, la composición tardía y otros.

1.6.4 Arquitectura Orientada a Servicios.

La Arquitectura Orientada a Servicios SOA (*Service Oriented Architecture*, por sus siglas en inglés) es una filosofía de diseño que permite un mejor alineamiento de las Tecnologías de la Información IT (*Information Technology* por sus siglas en inglés) con las necesidades de negocio, permitiendo a empleados, clientes y socio-comerciales responder de forma más rápida y adaptarse adecuadamente a las presiones del mercado. [16]

Entre los beneficios de SOA se destacan: [17]

- La reducción de costos y tiempo en el desarrollo de aplicaciones ya que SOA permite reutilizar los módulos existentes y el código nuevo para generar nuevas aplicaciones. Como consecuencia también se reducen los costos de mantenimiento.
- Dado que todas las aplicaciones pretenden satisfacer un servicio, se incrementa así la calidad del mismo y la productividad de la empresa.

- Las metodologías que aterrizan el concepto de SOA facilitan la integración entre aplicaciones nuevas así como con los sistemas existentes.
- Desarrollo de aplicaciones más productivas, flexibles, más seguras y manejables para gestionar procesos de negocio críticos a medida que evolucionan o cambian las necesidades del negocio.
- Fortalecimiento y consolidación de los procesos de negocio a través de aplicaciones que comparten servicios comunes. En algunas empresas, hay procesos de negocio que necesitan de la misma funcionalidad de una aplicación; pero esto se ha resuelto con aplicativos diferentes. Con SOA la misma aplicación apoya procesos diversos.

Desventajas de SOA:

- La velocidad de intercambio de información entre sistemas es más lenta que mediante una conexión directa. Intercambiar grandes volúmenes de información puede afectar el rendimiento del bus.
- El que SOA sea una arquitectura muy estudiada para aportar grandes beneficios, no implica que sea recomendable su uso para todos los escenarios. Recomendar una arquitectura SOA, su alcance, dónde y cómo aplicarla, suele ser un proceso lento debido al gran impacto que tiene en los sistemas que se encuentran en producción.

1.7 Herramientas.

1.7.1 Servidor de aplicaciones

Un servidor de aplicaciones es un *software* que proporciona aplicaciones a los equipos o dispositivos clientes, por lo general a través de Internet y utilizando el protocolo HTTP (*HyperText Transfer Protocol* por sus siglas en inglés). Los servidores de aplicación se distinguen de los servidores web por el uso extensivo del contenido dinámico y por su frecuente integración con bases de datos. [18]

JBoss es un servidor de aplicaciones Java EE (*Java Enterprise Edition* por sus siglas en inglés) de *software* libre implementado en Java puro. Al estar basado en Java, puede ser utilizado en cualquier sistema operativo que lo soporte. JBoss Application Server 4.2.2 ha sido seleccionado para el desarrollo del componente web que se pretende desarrollar con la presente investigación. El mismo proporciona una gama completa de prestaciones para Java EE 5, así como ampliación de los servicios empresariales, incluyendo *clustering*, *caching* y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta EJB 3.0 (*Enterprise Java Beans* por sus siglas en inglés) y esto hace que el desarrollo de las aplicaciones sea mucho más simple.

Una de las facilidades que este servidor presenta es que puede ser instalado sobre varios sistemas operativos, tales como Windows o GNU/Linux.

1.7.2 Sistema Gestor de Base de Datos.

Un Sistema Gestor de Base de Datos (SGBD) 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. [19]

PostgreSQL es un sistema de gestión de base de datos objeto-relacional, que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Se distribuye bajo licencia BSD (*Berkeley Software Distribution* por sus siglas en inglés) y con su código fuente disponible libremente. Utiliza el modelo Cliente/Servidor así como procesos en vez de múltiples hilos de ejecución. Un fallo en uno de los procesos no afectará al resto y el sistema continuará funcionando. Se desempeña muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo simultáneamente al sistema. [20]

En sus más de 10 años de existencia, PostgreSQL ha logrado reunir las siguientes características: [21]

- Aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son las consultas SQL declarativas, el control de concurrencia multiversión, el soporte multiusuario, las transacciones, la optimización de consultas así como de herencia y arreglos.
- Altamente Extensible. Permite operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario.
- Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (más conocidas como *joins* por el uso de la terminología en inglés) SQL92.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

- La flexibilidad de la API (*Application Programming Interface* por sus siglas en inglés) de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo del propio sistema gestor.
- Posee soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL (*Procedural Language/PostgreSQL Structured Query Language* por sus siglas en inglés).
- Cuenta con habilidad para usar Perl, Python o TCL (*Tool Command Language* por sus siglas en inglés) como lenguaje procedural embebido.

PostgreSQL 8.4 fue seleccionado como el Sistema Gestor de Base de Datos a utilizar en la presente investigación. Entre las principales áreas que fueron mejoradas con esta versión se encuentran:

- Funciones de ventanas
- Expresiones de tablas comunes y consultas recursivas
- Permisos de columnas y configuración regional
- Índices *hash*
- Rendimiento de unión para consultas EXISTS Y NOT EXISTS
- Mapa de visibilidad
- Soporte de certificados SSL (*Secure Sockets Layer* por sus siglas en inglés) para la autenticación de usuarios

1.7.3 Visual Paradigm

Visual Paradigm para UML (*Unified Modeling Language* por sus siglas en inglés) es una herramienta CASE (*Computer Aided Software Engineering* por sus siglas en inglés) aplicable en todo el ciclo de vida del desarrollo de software. Soporta UML, SysML (*Systems Modeling Language* por sus siglas en inglés), BPMN (*Business Process Modeling Notation* por sus siglas en inglés), entre otras tecnologías. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones.

Visual Paradigm 6.4 ha sido la herramienta seleccionada para soportar el ciclo de desarrollo del componente web para el módulo Salud Materno-Infantil del SIAPS.

1.7.4 Eclipse Ganymede 3.4.2

Eclipse es un IDE (*Integrated Development Environment* por sus siglas en inglés) de código abierto y multiplataforma que ha alcanzado un alto grado de madurez en el desarrollo de lo que se conoce como “Aplicaciones de cliente enriquecido”. Cuenta con herramientas para desarrollar aplicaciones de consola, web y servicios web con diferentes servidores de aplicaciones tales como JBoss, Websphere y Glassfish. Fue desarrollado originalmente por IBM (*International Business Machines* por sus siglas en inglés) y su futuro está ahora en manos de la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

En cuanto a las aplicaciones clientes, Eclipse provee al programador con *frameworks* muy ricos para el desarrollo de aplicaciones gráficas, para la definición y manipulación de modelos de software, aplicaciones web, entre otros.

Eclipse Ganymede 3.4.2 asegura robustez y rendimiento. Mejora, respecto a versiones anteriores, su soporte de búsqueda con expresiones regulares. Cuenta con un visor de problemas en el código, con asistente para la conversión a *StringBuffer* y con mejoras en el *debugger*.

1.7.5 PgAdmin III 1.10.5

PgAdmin III es una herramienta para la administración gráfica de PostgreSQL. Funciona sobre casi todas las plataformas. Fue diseñado para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL a la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración. La aplicación también incluye un editor de la sintaxis SQL, un editor de código del lado del servidor, un agente para la programación de tareas «SQL/batch/shell» y soporte para el motor de replicación Slony-I.

PgAdmin III 1.10.5 es la herramienta seleccionada para la administración gráfica de PostgreSQL 8.4 en el desarrollo de la presente investigación.

1.7.6 ER/Studio 8.0

Es una herramienta de modelado de datos fácil de usar y multinivel, concebida para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los

administradores, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones grandes y complejas.

ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación en HTML y fácil creación de reportes.

1.8 Frameworks, librerías y componentes.

Los frameworks se pueden considerar como soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución). Aceleran el proceso de desarrollo, permiten reutilizar código ya existente y promover buenas prácticas como el uso de patrones.

A su vez, las librerías son un conjunto de subprogramas utilizados para desarrollar *software*. Contienen códigos que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos.

1.8.1 Java Server Faces

La tecnología JSF (*Java Server Faces* por sus siglas en inglés) constituye un *framework* para la creación de interfaces de usuario del lado del servidor, dirigido a aplicaciones web basadas en tecnología Java y en el patrón de arquitectura Modelo Vista Controlador.

Los principales componentes de la tecnología Java Server Faces son: [22]

- Una API y una implementación de referencia para:
 - Representar componentes de interfaz de usuario y manejar su estado.
 - Manejar eventos, validar en el lado del servidor y convertir datos.
 - Definir la navegación entre páginas.
 - Soportar internacionalización y accesibilidad.
 - Proporcionar extensibilidad para todas estas características.
- Una librería de etiquetas JSP (*Java Server Pages* por sus siglas en inglés) personalizadas para dibujar componentes UI (*User Interface* por sus siglas en inglés) dentro de una página JSP.

Algunas de las ventajas de JSF son: [23]

- Permite conectar eventos generados en el cliente con código de la aplicación en el lado servidor.

- Posibilita mapear componentes UI a una página de datos en el lado del servidor.
- Permite construir una interfaz de usuario con componentes reutilizables y extensibles.

Desventajas de JSF:

- Existe poca documentación sobre este *framework*.
- Los validadores no son tan completos como en *Struts* (un *framework* con finalidad similar a la de JSF).
- No tiene validaciones del lado del cliente.
- No soporta el método GET para el intercambio de datos, lo cual da cierta seguridad; pero quita flexibilidad y en algunos casos “direccionalidad”, lo cual puede ser muy molesto.

1.8.2 RichFaces

RichFaces es una librería de código abierto basada en Java que permite crear aplicaciones web con AJAX (*Asynchronous JavaScript And XML* por sus siglas en inglés). Construye sobre el framework JSF. Sobre él implementa filtros para permitir peticiones en la página. La singularidad del planteamiento que ofrece es que la petición AJAX provoca una ejecución en el servidor y finalmente una renderización parcial o total de la página del navegador. Así el control de lo que sucede está en el servidor.

Richfaces 3.3.1 permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes listos para usar, altamente configurables en cuanto a temas y esquemas de colores predefinidos o desarrollados a conveniencia, lo que mejora grandemente la experiencia de usuario. Soporta Facelets y contiene un set de componentes visuales, los más comunes para el desarrollo de una aplicación web, que cubre casi todas las necesidades.

1.8.3 Asynchronous JavaScript And XML

AJAX (*Asynchronous JavaScript And XML* por sus siglas en inglés), no es un lenguaje de programación sino un conjunto de tecnologías tales como HTML, JavaScript, CSS (*Cascading Style Sheets* por sus siglas en inglés), DHTML (*Dynamic HTML* por sus siglas en inglés), PHP (*HyperText Preprocessor* por sus siglas en inglés), ASP.NET (*Application Service Providers* por sus siglas en inglés), JSP y XML (*Extensible Markup Language* por sus siglas en inglés) que permiten hacer páginas de Internet más interactivas. La característica fundamental de AJAX es que permite actualizar parte de una página con información que se

encuentra en el servidor sin tener que refrescar completamente la página. De modo similar su puede enviar información al servidor. [24]

Ventajas de AJAX: [25]

- Utiliza tecnologías ya existentes.
- Soportada por la mayoría de los navegadores modernos.
- Interactividad. El usuario no tiene que esperar hasta que lleguen los datos del servidor.
- Portabilidad. No requiere *plugins* como Flash y Applet de Java.
- Mayor velocidad debido a que no hay que retornar toda la página nuevamente.
- La página se asemeja a una aplicación de escritorio.

Desventajas en el uso de AJAX:

- En algunos casos es necesario incluir señales para que el usuario se percate que el servidor ha respondido.
- AJAX depende de JavaScript y el cliente puede tener desactivada la opción de ejecutar el código JavaScript.
- Explotar una aplicación con excesivos recursos AJAX puede degradar el rendimiento del sistema.
- Una mala elección de tratar múltiples conexiones asíncronas puede ocasionar problemas.

1.8.4 AJAX4JSF

AJAX4JSF es una librería *open source* o de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas, dotándolas con tecnología AJAX de forma limpia y sin añadir código JavaScript. Mediante esta librería se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones automáticas al servidor, controlar cualquier evento de usuario, entre otras funcionalidades. En definitiva AJAX4JSF permite dotar a una aplicación JSF de contenido mucho más ajustado a las necesidades del usuario, con muy poco esfuerzo.

1.8.5 Facelets.

Facelets es un framework simplificado de presentación, en donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y

mejora los informes de errores que tiene JSF. Permite la definición de disposición de páginas basada en plantillas, la composición de componentes, creación de etiquetas personalizadas, desarrollo amigable para el diseñador gráfico y creación de librerías de componentes.

Facelets 1.1 ha sido seleccionado para la concepción de plantillas en el desarrollo del componente web para el módulo Salud Materno-Infantil del SIAPS.

1.8.6 JBoss Seam.

JBoss Seam 2.1.1 es un potente framework para desarrollar aplicaciones Web 2.0 al unificar e integrar tecnologías como AJAX, JSF, EJB, Java Portlets y BPM (*Business Process Management* por sus siglas en inglés). Otra característica importante es que se pueden hacer validaciones en los POJOs (*Plain Old Java Objects* por sus siglas en inglés) además de manejar directamente la lógica de la aplicación y de negocio desde las *sessions beans*.

Seam provee una mayor granularidad de contextos de estado. La principal, quizás sea el contexto conversacional así como el asociado a procesos del negocio. Con estos se logra un uso más eficiente de la memoria evitando *memory-leaks*. Integra además el concepto de *workspaces* permitiendo que el usuario tenga en varios *tabs* o ventanas del navegador, actividades del negocio con contextos completamente aislados. Seam integra transparentemente la administración de procesos del negocio vía JBoss JBPM, haciendo muy fácil implementar y optimizar complejas colaboraciones (*workflows*) y complejas interacciones con el usuario (*pageflows*).

1.8.7 Hibernate.

Hibernate es una herramienta ORM (*Object-Relational Mapping* por sus siglas en inglés) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML o anotaciones en los *beans* de las entidades que permiten establecer estas relaciones.

Para lograrlo, permite al desarrollador, detallar las relaciones que existen y qué forma tienen. Con esta información Hibernate le ofrece a la aplicación la posibilidad de manipular los datos de la base operando sobre objetos, con todas las características de la Programación Orientada a Objetos (POO). Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Igualmente, 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 todos los motores de bases de datos con un ligero

incremento en el tiempo de ejecución. Hibernate 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. Tiene además, la funcionalidad de crear la base de datos a partir de la información disponible. Hibernate ofrece un lenguaje de consulta de datos llamado HQL (*Hibernate Query Language* por sus siglas en inglés), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "critería"). Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA (*Java Persistence API* por sus siglas en inglés), que es parte de esta plataforma.

Desventajas de Hibernate:

- Se necesita de un gran esfuerzo para comprender Hibernate.
- A veces la depuración y el rendimiento presentan dificultades.
- Hibernate es más lento que el puro JDBC (*Java DataBase Connectivity* por sus siglas en inglés) ya que genera muchas sentencias SQL en tiempo de ejecución.
- No es adecuado para el procesamiento por lotes. Es recomendable usar JDBC para dicho procesamiento.

Se escogió Hibernate 3.3 para ser utilizado en el desarrollo del componente web perteneciente a la presente investigación. Esta versión cuenta además con las siguientes características:

- Migración a un sistema de construcción con Maven.
- División del proyecto en varios módulos lo que facilita el ver y administrar las dependencias.
- Rediseño de las SPI (*Serial Peripheral Interface* por sus siglas en inglés) para el caché de segundo nivel.
- Integración con JBossCache 2.x como proveedor de caché de segundo nivel.

1.8.8 Java Persistence API

JPA (*Java Persistence API* por sus siglas en inglés), es la API de persistencia desarrollada para la plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE). Ha sido incluida en el estándar EJB 3.

La persistencia en este contexto cubre tres áreas:

- La API en sí misma, definida en `javax.persistence.package`
- La JPQL (*Java Persistence Query Language* por sus siglas en inglés)
- Metadatos objeto/relacional

El objetivo que persigue la creación de JPA es no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional), como sí pasaba con EJB2, y permitir usar objetos regulares (POJOs).

1.9 Tecnologías.

1.9.1 Cascading Style Sheets

CSS (*Cascading Style Sheets* por sus siglas en inglés) es la tecnología desarrollada por el W3C (*World Wide Web Consortium* por sus siglas en inglés) con el fin de separar la estructura de la presentación. Es fundamental para la creación de páginas web complejas. Se utiliza para dar estilo a documentos HTML y XML. Los estilos definen la forma de mostrar los elementos. Esta tecnología permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. [26]

1.9.2 HyperText Markup Language

HTML (*HyperText Markup Language* por sus siglas en inglés) es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Permite describir la estructura y el contenido en forma de texto, además de complementarlo con objetos tales como imágenes. Se describe mediante etiquetas. Por otra parte, el HTML permite incluir scripts (por ejemplo, de Javascript), códigos que pueden modificar el comportamiento de los navegadores web y de otros procesadores de HTML. [27]

1.9.3 JavaScript.

JavaScript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. No requiere de compilación ya que el lenguaje funciona del lado del cliente. Los navegadores son los encargados de interpretar estos códigos. Gran parte de la programación en este lenguaje está

centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas, entre otros. Tiene la ventaja de ser incorporado en cualquier página web y puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.

Ventajas de JavaScript:

- JavaScript ofrece a los creadores de páginas web la capacidad de escribir guiones que dentro de la página, interactúan con objetos como formularios, marcos, color de fondo, entre otros. Una de sus aplicaciones más útiles es la validación de formularios antes de su envío al guión CGI (*Common Gateway Interface* por sus siglas en inglés) en el servidor.
- JavaScript es el complemento ideal de HTML, al permitir a la página realizar algunas tareas por sí misma, sin necesidad de estar sobrecargando el servidor del cual depende.

Desventajas de JavaScript:

- JavaScript no puede interactuar con el servidor
- No puede crear gráficos
- JavaScript trabaja de forma diferente en navegadores distintos.

1.9.4 eXtensible HyperText Markup Language

XHTML (*eXtensible HyperText Markup Language* por sus siglas en inglés) es un lenguaje de programación pensado para sustituir a HTML. XHTML es la versión XML de HTML con las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML.

1.9.5 Extensible Markup Language

XML (*Extensible Markup Language* por sus siglas en inglés), es una especificación/lenguaje de programación desarrollada por el W3C. XML es una versión de SGML (*Standard Generalized Markup Language* por sus siglas en inglés), diseñado especialmente para los documentos de la web. Posibilita que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones. [28]

Ventajas de XML:

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.

- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de XML.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla.
- Mejora la compatibilidad entre aplicaciones. Las aplicaciones se pueden comunicar de distintas plataformas, sin que importe el origen de los datos.
- Se transforman datos en información, pues se le añade un significado concreto y se le asocia a un contexto, con lo cual se tiene flexibilidad para estructurar documentos.

1.9.6 Java.

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principio de la década de 1990. El lenguaje en sí mismo toma gran parte 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, como la manipulación directa de punteros o memoria.

El lenguaje Java tiene la ventaja de ser multiplataforma. El mismo se ha extendido y ha cobrado cada día más importancia tanto en el ámbito de Internet como en la Informática en general. Propicia la integración de diversos *frameworks* que ofrecen múltiples facilidades tales como: persistencia, seguridad, *logs*, internacionalización, BPM, *testing*, *web services*, pantallas con elementos ricos de interfaz y relativa facilidad de uso.

Ventajas del lenguaje Java:

- Más simple. Elimina la complejidad de los lenguajes como “C” y da paso al contexto de los lenguajes modernos orientados a objetos.
- Familiar. Como la mayoría de los programadores están acostumbrados a programar en C o en C++, les será fácil comprender a Java por su sintaxis similar a la de estos lenguajes.
- Robusto. Java se encarga de manejar la memoria de la computadora. No es necesaria la preocupación del programador por los apuntadores ni por la memoria que no se esté utilizando.

- Seguro. Cuenta con políticas que evitan se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los *applets*, que limitan lo que se puede y no se puede hacer con los recursos críticos de una computadora.
- Portable. Como el código compilado de Java (conocido como byte code) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java.
- Independiente de la arquitectura. Al ser un lenguaje interpretado solamente se debe implementar un intérprete para cada plataforma. De esta forma, Java no depende de una arquitectura computacional definida.

Desventajas del lenguaje Java:

- Velocidad. Al tener que ser ejecutado mediante JVM (*Java Virtual Machine* por sus siglas en inglés) esto hace que Java no sea tan rápida como otras tecnologías, por ejemplo C++.
- No es tan simple. Adquirir un estilo de programación orientada a objetos es difícil. Saber además sobre la filosofía y los componentes en las clases de librerías, es complejo y consume mucho tiempo.

1.10 Guía para el desarrollo del software.

CMMI (*Capability Maturity Model Integration* por sus siglas en inglés) es un modelo para la mejora y evaluación de procesos de desarrollo, mantenimiento y operación de sistemas de software. Las mejores prácticas CMMI se publican en documentos llamados modelos, los cuales contienen el conjunto de prácticas relacionadas que son ejecutadas de forma conjunta para conseguir determinados objetivos.

Así es como el modelo CMMI establece una medida del progreso, conforme al avance en niveles de madurez. Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. El hecho de alcanzar estas áreas se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables.

CMMI consta de 22 áreas de proceso distribuidas dentro de 5 niveles de madurez:

- Nivel 1: Inicial.
- Nivel 2: Administrado.

- Nivel 3: Definido.
- Nivel 4: Cuantitativamente administrado.
- Nivel 5: Optimizado.

Las organizaciones son evaluadas y reciben una calificación de nivel 1-5 siguiendo los niveles de madurez. Este enfoque se denomina “Representación Escalonada”.

Dichas organizaciones pueden también ser evaluadas por áreas de procesos en vez de por niveles de madurez, al adquirir los niveles de capacidad en cada una de ellas, obteniendo el "Perfil de Capacidad" de la organización. A esta visión de la organización se le conoce como “Representación Continua.”

Cada área de procesos dentro de los niveles de capacidad posee un conjunto de objetivos genéricos y específicos. Los objetivos genéricos asociados a un nivel de capacidad establecen lo que una organización debe alcanzar en ese nivel de capacidad. El logro de cada uno de esos objetivos en un área de proceso significa mejorar el control en la ejecución de la misma. Estos objetivos tienen un conjunto de prácticas genéricas que se aplican a cualquier área de procesos porque pueden mejorar el funcionamiento y el control de cualquier proceso.

Los objetivos específicos se aplican a una única área de procesos y localizan las particularidades que describen qué se debe implementar para satisfacer el propósito del área de procesos. A su vez, cada objetivo está formado por un conjunto de prácticas determinadas y subprácticas. Las prácticas son actividades que se consideran importantes en la realización del objetivo específico al cual están asociadas, describen las actividades esperadas para lograr dicho objetivo en un área de procesos y las subprácticas constituyen descripciones detalladas que sirven como guía para la interpretación de una práctica genérica o específica.

El objetivo de la Administración de Requisitos es gestionar los requisitos de los elementos del proyecto y sus componentes e identificar inconsistencias entre estos requisitos, el plan de proyecto y los elementos de trabajo. En este proceso se deben gestionar todos los requisitos del proyecto, tanto técnicos como no técnicos. Estos requisitos han de ser revisados conjuntamente con la fuente de los mismos así como con las personas que se encargarán del desarrollo posterior. Para llevar a cabo estas actividades y como resultado de que la Universidad de las Ciencias Informáticas (UCI) se encuentra desarrollando un proceso de mejora con el objetivo de obtener el Nivel 2 de CMMI, es creado y utilizado por la UCI el documento

IPP- 3510:2009 (Libro de Proceso para la Administración de Requisitos), cuyo objetivo es definir el proceso de administración de requisitos.

Este documento establece el ciclo de vida a seguir, asociado a los proyectos involucrados en el proceso de mejora, el cual consta de 9 fases y se establece por cada fase la relación con los subprocesos descritos en el libro de procesos específico del área Administración de Requisitos.

Ciclo de vida básico:

Estudio Preliminar: Se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo.

Modelación del Negocio: Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para la descripción y modelado de negocio pueden ser utilizadas diferentes técnicas como el Modelado de Casos de Uso del Negocio y BPMN.

Requisitos: Se desarrolla un modelo del sistema que se va a construir. Incluye un conjunto de casos de uso, servicios que describen todas las interacciones que tendrán los usuarios con el software, estos responden a los requisitos funcionales del sistema.

Análisis y Diseño: Se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos elaborados en esta etapa son más formales y específicos de una implementación. Durante esta fase son desarrollados el documento de arquitectura, diagramas de clases, diagramas de entidad relación, diagrama de despliegue entre otros.

Implementación: Se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares, a partir de los resultados del análisis y diseño.

Pruebas Internas: Se verifica el resultado de la implementación probando, según sea necesario, cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.

Pruebas de Liberación: Se llevan a cabo las pruebas diseñadas e implementadas por el Laboratorio Industrial de Pruebas de Software, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

Despliegue: Se procede a la entrega de la solución, así como a la instalación, configuración, prueba y puesta en marcha del software en el entorno real del cliente. También deben realizarse en este período la capacitación y acompañamiento a clientes para asegurar que adquieran los conocimientos necesarios en la manipulación del software.

Soporte: Se ofrece un servicio para resolver conflictos y problemas de usabilidad y rendimiento del software entregado al cliente, suministrándole actualizaciones y parches a errores.

La descripción del IPP-3510:2009 incluye la definición de roles, sus responsabilidades y las habilidades en la ejecución de las actividades de los distintos procesos y los productos típicos de trabajos que se obtienen como resultado de la ejecución de dichas actividades. La información contemplada en cada producto típico de trabajo está especificada teniendo como base la experiencia de uso de los expertos de la metodología RUP (*Rational Unified Process* por sus siglas en inglés).

RUP, es una metodología de desarrollo de software que proporciona un método disciplinado para asignar las tareas y responsabilidades dentro del equipo de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que resuelva las necesidades del usuario dentro de un cronograma predecible y al menor costo posible. Se caracteriza por ser iterativo e incremental, centrado en la arquitectura y guiado por casos de uso y dividir el proceso en ciclos de desarrollo que se agrupan en fases, en las cuales las actividades se distribuyen entre 9 flujos de trabajo. Cada fase finaliza con un hito donde se debe tomar una decisión importante.

Para la descripción de los productos de trabajo resultantes de cada uno de los flujos de trabajo de RUP así como de las fases del ciclo vida definido en el IPP- 3510:2009 se utiliza UML, con la especificación del uso del estándar BPMN para la descripción de los procesos de negocio.

1.11 Proceso de negocio

Un proceso de negocio es un conjunto de tareas relacionadas de forma lógica, llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas. [29]

1.11.1 BPM

BPM (*Business Process Management* por sus siglas en inglés), consiste en un conjunto de herramientas, tecnologías, técnicas, métodos y disciplinas de gestión para la identificación, modelación, análisis, ejecución, control y mejora de los procesos de negocio. Las mejoras incluyen tanto cambios de mejora continua como cambios radicales. [30]

1.11.2 Notación Utilizada para Modelar los Procesos del Negocio.

BPMN (*Business Process Modeling Notation* por sus siglas en inglés) es un nuevo estándar que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. BPMN proporciona a los negocios la capacidad de entender sus procedimientos internos en una notación gráfica, facilitando a las organizaciones la habilidad para comunicar esos procedimientos de una manera estándar. Su principal objetivo es proveer una notación que sea fácilmente entendida por todos los usuarios, desde el analista de negocio, el desarrollador técnico y los administradores del negocio.

1.11.3 Lenguaje Unificado de Modelado

UML (*Unified Modeling Language* por sus siglas en inglés), es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. Los estereotipos son el mecanismo de extensibilidad incorporado más utilizado dentro de UML. Un estereotipo representa una distinción de uso. Puede ser aplicado a cualquier elemento de modelado, incluyendo clases, paquetes, relaciones de herencia, entre otros [31]

En el capítulo, el estudio en profundidad del dominio del problema ha permitido comprender lo que se quiere alcanzar con la investigación. Esa claridad ha servido de base para la selección de las herramientas y tecnologías explicadas, las cuales han sido estudiadas y analizado el nivel de adecuación de cada una para dar respuesta a los requerimientos de la aplicación informática que se quiere lograr. Para ello se ha indagado en las tendencias actuales de la industria del *software* y se han estudiado diferentes soluciones existentes en Cuba y el resto del mundo relacionadas con la gestión de la información que se genera en las actividades de salud materno-infantil.

Capítulo 2. Características del Sistema

En el capítulo se argumentan los aspectos relacionados con el negocio y el sistema informático. Se describe con rigor el flujo actual de los procesos y actividades que deben ser informatizadas, se desglosan los requerimientos funcionales y no funcionales que cumplirá la aplicación para definir las funcionalidades que serán implementadas así como su nivel de importancia.

2.1 Flujo actual de los procesos involucrados en el campo de acción.

Los procesos de captación de la embarazada y el recién nacido, ingreso y atención prenatal de la gestante en el hogar materno constituyen el centro de atención de la investigación que se realiza.

El proceso de captación de la embarazada debe ocurrir antes de las catorce semanas y debe realizarse por el médico de familia en el consultorio médico. El mismo consiste fundamentalmente en la creación de la Historia Clínica de Atención Pre y Post-Natal y el Carné de la Embarazada, documentos que recogen los datos más importantes de la gestante durante los períodos de parto y posparto. En este proceso el médico asistencial comienza verificando el tiempo de gestación de la paciente y toma los datos generales, obstétricos y biopsicosociales de la misma, los que son reflejados en los documentos antes mencionados. Se consulta la Tarjeta de Vacunación para verificar la fecha de la última vacunación y decidir si planificarle o no, una nueva inmunización. De igual forma se revisa la Tarjeta de Citología para observar la fecha de la última prueba citológica y decidir si repetirla la misma o no.

El médico de familia actualiza el grupo dispensarial de la paciente y hace un resumen del interrogatorio a la embarazada. Luego realizar un examen físico general y por aparatos a la gestante a la vez que va actualizando con la información que ello genera, tanto la Historia Clínica de Atención Pre y Post-natal como el Carné de la Embarazada. En dependencia de los resultados de los mismos, se le indica un tratamiento a la paciente así como exámenes complementarios o solo estos últimos, para luego emitir una orden de análisis. Con posterioridad la paciente es remitida a consultas con especialistas, el médico de familia la orienta a partir de consejos profilácticos y la cita a una reconsulta.

En el proceso de ingreso de la embarazada en el hogar materno interviene el médico asistencial y el asistente de enfermería. La gestante entrega sus datos contenidos en el Carné de la Embarazada los cuales son reflejados por el médico de familia en el documento Resumen de Ingreso y Datos Generales. Éste hace un interrogatorio a la paciente y realiza un examen físico general y por sistemas a la gestante.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

La información obtenida es reflejada en dos documentos que se crean: Historia de la Enfermedad Actual y Examen Físico. Se confecciona el documento Discusión Diagnóstica y Evolución por parte del médico, en el cual refleja sus impresiones diagnósticas. Finalmente la asistente de enfermería realiza una valoración de la gestante y una impresión diagnóstica de la misma, creándose la Hoja de Evolución Diaria.

En el proceso de atención prenatal de la embarazada en el hogar materno intervienen el médico asistencial, el asistente de enfermería, el técnico de laboratorio y especialistas en Psicología, Obstetricia, Nutrición y el Clínico. El médico asistencial o la enfermera toman el peso y la talla de la gestante una vez a la semana y registran los datos. Los especialistas acuden al hogar materno a realizar consultas a las pacientes una vez a la semana. Cada especialista elabora un resumen de la evolución de la paciente. La especialista en nutrición indica la dieta correspondiente a la gestante y actualiza la Hoja de Vigilancia Nutricional específicamente además del resto de los documentos que le competen. El médico indica un conjunto de exámenes complementarios que debe realizarse semanalmente la gestante.

Los técnicos de laboratorio que acuden una vez a la semana al hogar materno, toman las muestras correspondientes según los tipos de análisis indicados a las gestantes y son llevadas al laboratorio para ser analizadas y emitir los resultados. El médico asistencial registra los resultados de los análisis, actualizando la Hoja de Complementarios. La enfermera toma diariamente los signos vitales de la paciente y registra los datos. El médico asistencial realiza un conjunto de controles obstétricos de forma diaria y registra los datos, e indica a modo de resumen, los procedimientos a ser tomados por la gestante.

La enfermera ejecuta diariamente las indicaciones médicas orientadas por el médico asistencial para las embarazadas y actualiza las Indicaciones Médicas y Cumplimiento de Enfermería. El médico asistencial o la enfermera evalúan el estado integral de la gestante diariamente y elaboran un resumen de la evaluación indicando quién la realiza. Si la misma ha superado el motivo de ingreso, el médico asistencial le da de alta.

La captación del recién nacido debe ocurrir antes de que transcurran las primeras cuarenta y ocho horas posteriores al parto y debe ser realizada por el médico asistencial. En caso de que el recién nacido aún se encuentre hospitalizado, se realizará en la instalación hospitalaria de lo contrario, se efectuará en su domicilio o en el consultorio médico de la familia previa citación realizada a la madre por parte del médico asistencial. Este proceso consiste en la creación de la Historia Clínica del Niño (HCN), en el cual

inicialmente se utiliza como documento de consulta el Carné de Salud Infantil (CSI), pues en él se recogen los primeros datos (nombre y apellidos del niño, nombre del padre y la madre, lugar de nacimiento, tipo y fecha de las primeras vacunas, entre otros) luego de finalizado el parto. El CSI es creado inmediatamente después del parto en el Departamento de Neonatología.

Con la creación de la HCN se recoge la información referente al seguimiento del embarazo, los datos perinatales y posnatales, como el tipo de parto, la atención prenatal, el tipo de embarazo y otros. La enfermera toma la talla y el peso del niño y el médico realiza un examen físico completo donde abarca desde los datos más generales del paciente, como el aspecto y la temperatura, hasta los correspondientes a cada uno de los sistemas (cardiovascular, digestivo, genitourinario, entre otros). La evaluación del seguimiento del desarrollo psicomotor y del funcionamiento familiar son otros de los tópicos que se describen en la captación.

En este proceso se indica la realización del examen de Fenilcetonuria y se actualiza la Historia de Salud Familiar (HSF) pues se produce la inserción de un nuevo miembro a la misma. Es el momento en que se incluye al recién nacido como un nuevo paciente del Sistema Nacional de Salud.

2.2 Objeto de automatización.

Los procesos de captación de la embarazada y el recién nacido, ingreso y atención prenatal de la gestante en el hogar materno que se realizan actualmente como parte del Programa de Atención Materno-Infantil, contemplan como principales problemáticas, el registro manual y en papel de la información obtenida, exponiéndose al deterioro y la consecuente pérdida.

La información que se registra es muy diversa si se tiene en cuenta la participación de varios roles en la actualización de un mismo documento y la carencia de modelos oficiales que estandaricen la estructura de la misma. No se garantiza, por tanto, que siempre aparezcan los datos necesarios y se corre el riesgo de malinterpretaciones o no entendimiento de los éstos.

De igual forma, resulta difícil acceder, desde una institución de salud y con la inmediatez que se requiere, a la información que sobre el paciente; posee otra institución de salud, por lo que existe una gran dependencia del interrogatorio al paciente y no de documentos oficiales preconcebidos. La mesurada atención que precisan el niño y la embarazada para la detección a tiempo de patologías que pongan en riesgo la vida de la gestante y el feto así como del lactante en su etapa madurativa, se ve afectada al no disponer de información que posea la veracidad e integralidad que se requiere.

Cuba no cuenta en la actualidad con un sistema informático que gestione la información relacionada con los procesos referidos y que logre el cumplimiento del objetivo planteado para esta investigación.

Teniendo en cuenta lo anteriormente descrito se propone el desarrollo de un componente web que permita la automatización de los procesos:

Captación de la embarazada: se desea gestionar los datos referentes a la gestante, obtenidos en este proceso. Entre las actividades que comprende se encuentran la creación de la Historia Clínica de Atención Pre y Post-natal y el Carné de la Embarazada y la actualización de la Historia de Salud Familiar.

Ingreso de la embarazada en el hogar materno: se desea gestionar los datos referentes al ingreso de la gestante en el hogar materno. Entre las actividades de este proceso se encuentran la creación de la Historia Clínica de Atención Pre y Post-natal en esa institución de salud, la Hoja de Evolución Diaria y la Discusión Diagnóstica y Evolución.

Atención prenatal de la embarazada en el hogar materno: se desea gestionar los datos referentes a la embarazada en el período de atención prenatal en el hogar materno en caso de ser ingresada. Entre las actividades de ese proceso se encuentran la creación de la Hoja de Balance Nutricional, Hoja de Vigilancia Nutricional y Hoja de Evolución.

Captación del Recién Nacido: Se desea gestionar la información referente al recién nacido. Entre las actividades de proceso se encuentran el registro de las vacunas, la creación de la Historia Clínica del Niño y la actualización del Carné de Salud Infantil.

2.3 Modelado del negocio.

El ciclo de desarrollo de software cuenta como primera fase con el Modelado de Negocio. El mismo es de suma importancia ya que durante éste, el equipo de trabajo se familiariza con el funcionamiento de la empresa y conoce sus procesos característicos. Además se logra entender la estructura y la dinámica de la organización para la cual el sistema va a ser desarrollado. Se tiene una visión un poco más amplia de cuáles son las actividades que son ejecutadas y la interrelación existente entre las mismas en cada uno de los momentos.

2.3.1 Diagramas de procesos del Negocio.

2.3.1.1 Proceso Captación de la Embarazada.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Objetivos:	<ol style="list-style-type: none"> 1 Mantener una mejor atención a la gestante cuando es detectado el embarazo a tiempo. 2 Evaluar todo el período de embarazo detectando a tiempo cualquier anomalía en el feto o problema en la gestante. 3 Suministrar atención preventiva y continua a la salud de la gestante y el feto. 4 Reducir el riesgo de morbilidad y mortalidad materna y fetal.
Evento(s) que lo generan:	Acude la Gestante al CMF.
Precondiciones:	No aplicable.
Poscondiciones:	Se creó el Carné de la Embarazada y la Historia Clínica de Atención Pre y Post-natal y se inició el seguimiento de la misma.
Reglas de Negocio:	<p>Regla 1. Ver APS_SIAPS_0116_RNeg_SaludMI_Wv1.0.</p> <p>Regla 2. Ver APS_SIAPS_0116_RNeg_SaludMI_Wv1.0.</p> <p>Regla 3. Ver APS_SIAPS_0116_RNeg_SaludMI_Wv1.0.</p> <p>Regla 4. Ver APS_SIAPS_0116_RNeg_SaludMI_Wv1.0.</p>
Responsables:	Médico asistencial.
Clientes internos:	CESIM
Clientes externos:	Gestantes.
Entradas:	<p>Tarjeta de Vacunación</p> <p>Tarjeta de Citología</p>

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Salidas:	Carné de Embarazada. Historia Clínica de Atención Pre y Post-natal Remisión. Historia de Salud Familiar. Orden de Análisis. Tarjeta de Vacunación (si procede). Tarjeta de Citología (si procede). Receta (si procede). Método (si procede).
Actividades:	Verificar tiempo de gestación. Tomar datos generales, obstétricos y biopsicosociales. Verificar fecha de última vacunación. Verificar fecha de última prueba citológica. Registrar fecha de la última vacunación. Registrar fecha de la última prueba citológica. Planificar Vacunación. Realizar Prueba Citológica. Actualizar grupo dispensarial de la gestante Elaborar el resumen del interrogatorio. Realizar examen físico completo. Indicar tratamiento. Indicar exámenes complementarios.

	Remitir a consultas con especialistas. Brindar consejos profilácticos. Citar a reconsulta.
--	--

2.4 Propuesta del sistema.

2.4.1 Requerimientos.

El objetivo principal de un sistema de información es automatizar tareas o actividades de un proceso de negocio, permitiendo a los actores organizacionales alcanzar sus metas particulares, así como las metas generales. Esta es la razón por la que el estudio del ambiente organizacional en el que se implementará el producto de software ha sido reconocido como una parte fundamental de la ingeniería de requisitos. El propósito de la ingeniería de requisitos es hacer que los mismos alcancen un estado óptimo antes de arribar a la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables, sin ambigüedades o contradicciones.

2.4.2 Especificación de los requerimientos de software

El análisis de requisitos es el conjunto de técnicas y procedimientos que permite conocer los elementos necesarios para definir un proyecto de *software*. Los requisitos funcionales son las capacidades o condiciones con las que debe contar un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente como reflejo de las exigencias del cliente. Los requisitos no funcionales son las cualidades que un producto informático debe poseer.

Para realizar bien el desarrollo de un *software* es esencial que se asegure una especificación completa de los requerimientos del mismo. Independientemente de lo bien diseñado o codificado que esté, un programa pobremente especificado decepcionará al usuario y hará fracasar el desarrollo.

2.4.3 Requerimientos funcionales

Los requerimientos funcionales determinan la capacidad que el sistema deberá tener para satisfacer las exigencias del cliente. Como se define en la ingeniería de requisitos, los requisitos funcionales establecen los comportamientos del sistema. Su núcleo es la descripción del comportamiento requerido, que debe ser

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

clara y concisa y proviene generalmente de reglas organizacionales o del negocio, o ser descubiertas por interacción con usuarios, inversores y otros expertos en la organización.

Luego de realizarse un estudio de los procesos del negocio, se determinaron los siguientes requisitos funcionales:

Comunes:	RF 10 Insertar datos del ingreso.
RF 1 Buscar paciente.	Proceso Atención Prenatal en Hogar Materno:
RF 2 Mostrar datos del paciente.	RF 11 Buscar pacientes ingresadas.
RF 3 Buscar clasificación CIE.	RF 12 Atender paciente.
RF 4 Buscar clasificación CIAP	RF 13 Actualizar datos del ingreso.
Proceso Captación de la embarazada:	RF 14 Gestionar Hoja de Discusión Diagnóstica.
RF 5 Gestionar datos de captación de embarazada.	RF 14.1 Buscar datos de discusión diagnóstica.
RF 5.1 Insertar datos de captación de embarazada.	RF 14.2 Insertar datos de discusión diagnóstica.
RF 5.2 Mostrar datos de captación de embarazada.	RF 14.3 Mostrar datos de discusión diagnóstica.
RF 6 Mostrar datos Tarjeta de Vacunación.	RF 15 Gestionar Hoja de Evolución Diaria.
RF 7 Mostrar datos Tarjeta de Citología.	RF 15.1 Buscar datos de evolución diaria.
RF 8 Actualizar datos de HSF.	RF 15.2 Insertar datos de evolución diaria.
Proceso Ingreso en Hogar Materno:	RF 15.3 Mostrar datos de evolución diaria.
RF 9 Gestionar remisiones pendientes.	RF 16 Insertar Hoja de Balance Nutricional.
RF 9.1 Listar remisiones pendientes.	RF 17 Gestionar seguimiento nutricional
RF 9.2 Mostrar datos de remisión pendiente.	RF 17.1 Insertar seguimiento nutricional.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RF 17.2 Mostrar seguimiento nutricional.	RF 20.1 Insertar datos del egreso.
RF 17.3 Buscar seguimiento nutricional.	RF 20.2 Mostrar datos del egreso.
RF 18 Gestionar Hoja de Evolución para Especialistas.	Proceso Captación del recién nacido:
RF 18.1 Buscar datos de evolución para especialistas.	RF 21 Gestionar datos de captación del recién nacido.
RF 18.2 Insertar datos de evolución para especialistas.	RF 21.1 Insertar datos de captación del recién nacido.
RF 18.3 Mostrar datos de evolución para especialistas.	RF 21.2 Mostrar datos de captación del recién nacido.
RF 19 Gestionar Hoja de Signos Vitales.	RF 22 Buscar dirección.
RF 19.1 Buscar datos de signos vitales.	
RF 19.2 Insertar datos de signos vitales.	
RF 19.3 Mostrar datos de signos vitales.	
RF 20 Gestionar Hoja de Egreso.	

2.4.4 Requerimientos no funcionales.

Los requerimientos no funcionales especifican las propiedades del sistema, como confiabilidad y seguridad. Contribuyen considerablemente a la aceptación del producto por parte del cliente. Normalmente están vinculados a requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

No	Categoría	Requisitos no funcionales
RNF 1	Usabilidad.	El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

		<ul style="list-style-type: none">➤ Usuarios normales: 20 días➤ Usuarios avanzados: 30 días
RNF 2	Confiabilidad	Se garantizarán políticas de respaldo a toda la información, evitando pérdidas en caso de desastres ajenos al sistema.
RNF 3	Confiabilidad	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 4	Confiabilidad	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 5	Confiabilidad	Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
RNF 6	Confiabilidad	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 7	Confiabilidad	Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para este, el elemento ya no exista.
RNF 8	Confiabilidad	Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo.
RNF 9	Eficiencia	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.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RNF 10	Eficiencia	El sistema minimizará el volumen de datos en las peticiones y optimizará el uso de recursos críticos como la memoria.
RNF 11	Eficiencia	Se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.
RNF 12	Eficiencia	El sistema adoptará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.
RNF 13	Soporte	Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación.
RNF 14	Soporte	Las notificaciones de las deficiencias detectadas en la aplicación desplegada deberán realizarse por escrito.
RNF 15	Soporte	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 16	Restricciones de diseño.	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 departamento de Atención Primaria de Salud perteneciente al CESIM.
RNF 17	Restricciones de diseño.	Las interfaces se realizarán siguiendo el estándar definido en el departamento de Atención Primaria de Salud perteneciente al CESIM.
RNF 18	Requerimientos de hardware	Las estaciones de trabajo deben tener como mínimo 256 Mb de memoria RAM y un microprocesador de 2.0 Hz.
RNF 19	Requerimientos de hardware	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

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

		sistema operativo GNU/Linux.
RNF 20	Requerimientos de hardware	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 GNU/Linux.
RNF 21	Requerimientos de software	El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL).
RNF 22	Requerimientos de software	El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2.

Con la descripción de los procesos involucrados en la presente investigación, se alcanzó un mayor entendimiento del negocio, lo que contribuyó sobremanera a definir lo que se debe automatizar y cómo hacerlo. Los modelos de procesos ilustran con claridad, la lógica de acciones dirigidas a la atención a la embarazada y al recién nacido. A partir del entendimiento entre cliente y desarrolladores acerca de lo que el sistema debe hacer y las propiedades con que debe contar, se determinaron los requisitos funcionales y no funcionales, los que sin dudas, marcan una meta a alcanzar. Los aspectos recogidos en este capítulo, dejan el escenario listo para un correcto diseño del sistema.

Capítulo 3. Diseño del Sistema

Para realizar la construcción exitosa de un producto de *software* resulta de vital importancia cumplir cabalmente con cada uno de los pasos dentro del ciclo de desarrollo del proyecto. La fase de “Análisis y Diseño” proporciona una vista más cercana al sistema que se quiere lograr. El diseño específicamente, tiene como objetivo principal, modelar el sistema de manera que soporte todos los requisitos tanto funcionales como no funcionales.

En este capítulo se especifican los diagramas de clases del diseño con el objetivo de describir la transformación de los requisitos funcionales en un diseño de clases, teniéndose en cuenta además, los aspectos relacionados con los requisitos no funcionales. Igualmente se definen los principios y patrones del diseño.

3.1 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Los patrones de diseño se clasifican en:

- Patrones Creacionales: Inicialización y configuración de objetos.
- Patrones Estructurales: Separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes.
- Patrones de Comportamiento: Más que describir objetos o clases, describen la comunicación entre ellos.

Los patrones de diseño son independientes del lenguaje en el que se utilicen (siempre y cuando el lenguaje sea orientado a objetos). Generalmente se presentan como diagramas UML.

Los patrones GRASP realizan una descripción de los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Son de vital importancia en el diseño orientado a objeto ya que es en esta fase donde se realiza la creación de las clases y las relaciones entre ellas. Constituyen un apoyo para la enseñanza. Ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Entre estos tipos de patrones se pueden destacar los siguientes:

- **Experto:** Tiene como objetivo la asignación de la responsabilidad de realizar una tarea determinada a aquel objeto que tiene la información (atributos) necesaria para ello. Expresa la intuición común de que los objetos hacen el trabajo relacionado con la información que tienen. No siempre la solución que proporciona es la más conveniente ya que normalmente ocasiona problemas de acoplamiento y de cohesión por lo que hay que tener en cuenta que este patrón es aplicable mientras se estén considerando los mismos aspectos del sistema.
- **Creador:** Este patrón, como su nombre lo indica, es el que crea y guía la asignación de responsabilidades relacionadas con la creación de objetos. Asigna la responsabilidad de que una clase B cree un objeto de la clase A. Según este patrón las relaciones *contiene* y *registra*, son las mejores candidatas para asignar la responsabilidad de crear objetos.
- **Alta cohesión:** Tiene como objetivo que cada elemento realice una labor única dentro del sistema, de forma tal que esta labor no sea desempeñada por el resto de los elementos. Asigna una responsabilidad de manera que la cohesión permanezca alta.
- **Bajo acoplamiento:** Propone que el nivel de acoplamiento entre los objetos sea bajo, entendiendo por acoplamiento el número de elementos (clases, subclasses, sistemas) a los que un objeto está conectado. Propone el diseño de clases que son más independientes, lo que reduce el impacto del cambio y facilita la reutilización de los sistemas.
- **Controlador:** Asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, entre otras). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

- **Polimorfismo:** Se basa en asignar la responsabilidad para el comportamiento, utilizando el polimorfismo, cuando las alternativas y comportamientos relacionados varíen según el tipo de clases.
- **Fabricación pura:** Desarrolla clases que se encarguen de construir objetos adecuados en cada momento (factorías). Estas son clases artificiales que no representan ningún concepto del dominio del problema. Tienen un conjunto de responsabilidades altamente cohesivas y con bajo acoplamiento y consiguen un diseño limpio y puro.

3.2 Modelo de Datos

El modelo de datos es uno de los artefactos más importante dentro del diseño. Es la combinación de una colección de estructuras de datos, operadores o reglas de inferencia y de reglas de integridad, las cuales definen un conjunto de estados consistentes. Éste puede ser usado como una herramienta para especificar los tipos de datos y la organización de los mismos así como la manipulación de consultas y datos.

En la investigación, para el control de la información se utiliza la herramienta Hibernate, la cual proporciona el control de la accesibilidad a los datos almacenados así como la realización de rápidas consultas. Hibernate, usada con otras herramientas, reduce significativamente el tiempo de desarrollo. Al tener menos líneas de código hace que el sistema sea más comprensible y que el desarrollador se centre en la lógica del negocio en lugar de la conexión. Además implementa una serie de patrones para la administración del modelo de datos, entre ellos están los siguientes:

- **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). Además sirve para mapear relaciones de muchos a muchos; en una base de datos esta relación se resuelve introduciendo una tercera tabla que contenga las claves primarias de ambas entidades involucradas en la relación. [32]
- **Lazy Load:** Carga los distintos componentes de una clase a medida que son solicitados, de modo que si los componentes no son solicitados se ahorra el tiempo que se hubiese consumido en encontrarlos.

- **Query Object:** Permite 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.

3.3 Modelo de Diseño

El Modelo de Diseño es una abstracción del Modelo de Implementación y su código fuente. Su principal objetivo es la construcción de un modelo lógico del sistema que se desea implementar. Es usado como entrada esencial en las actividades relacionadas con la implementación.

Su contenido lo conforman diagramas, clases, paquetes, subsistemas, cápsulas, protocolos, interfaces, relaciones, colaboraciones, atributos, entre otros, que se puedan considerar para el sistema en desarrollo. Esta estructura es muy significativa para la arquitectura en general debido a que no solo permite visualizar, especificar y documentar modelos estructurales; sino que también posibilita la construcción de sistemas ejecutables, aplicando ingeniería directa e inversa. [33]

3.3.1 Definiciones de elementos del diseño

En el Modelo de Diseño se establece la realización de las funcionalidades en clases incluyendo una orientación hacia el entorno de implementación. Está constituido por los diagramas de clases. Se realizará un diagrama de clases por cada funcionalidad simple y compleja en el caso de los codificadores.

La representación de estas clases está asociada al uso de UML para el modelamiento de aplicaciones y son identificadas de la siguiente manera: Página Servidor (en inglés *Server Page*), Página Cliente (en inglés *Client Page*) y Formulario (en inglés *Form*).

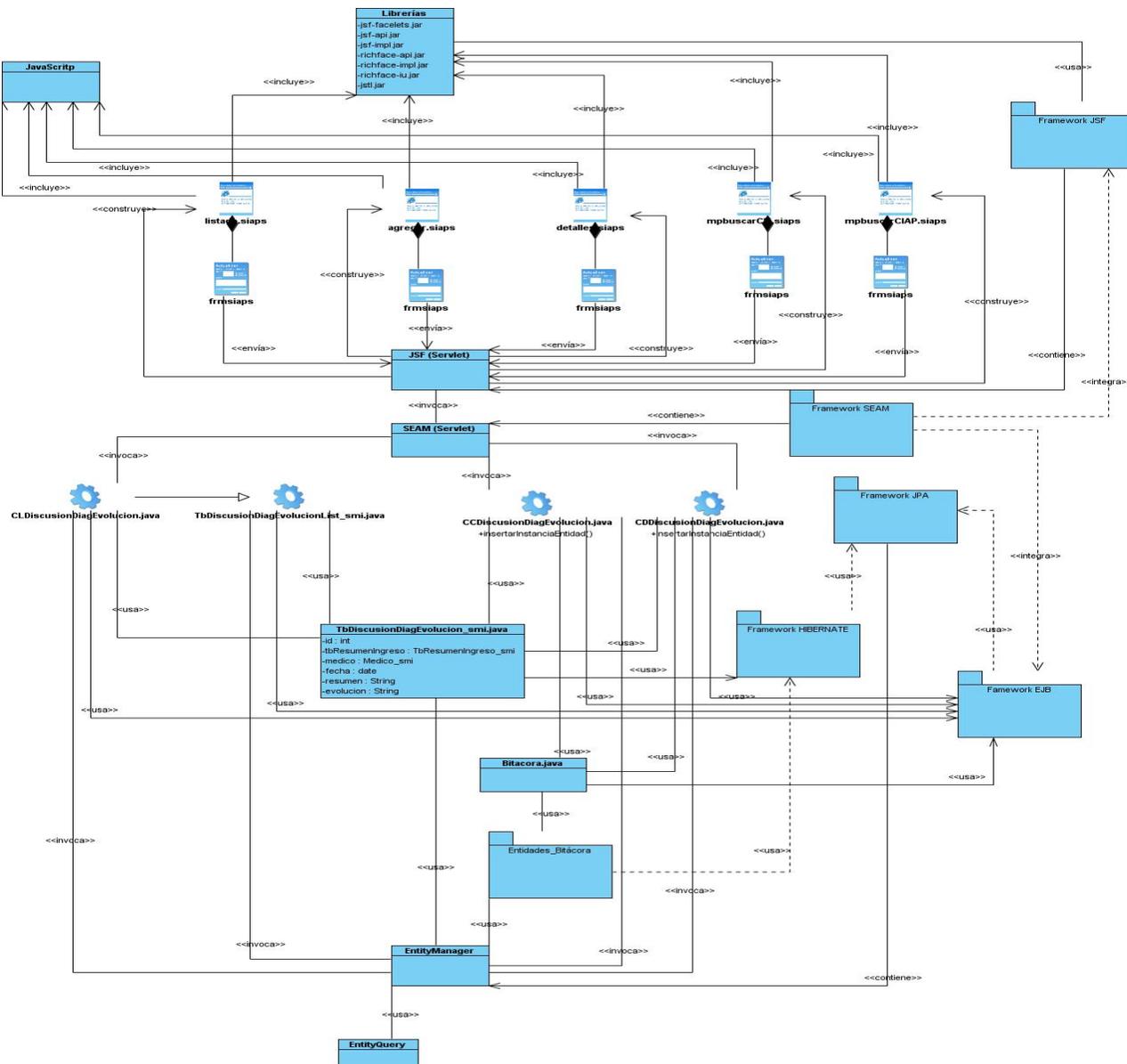
En el modelado de los diagramas de paquetes y las clases del diseño, es usada la siguiente nomenclatura: Diagrama de paquetes (Diagrama de paquetes_<Nombre del módulo>), Diagramas de clases del diseño (DCD_<Nombre de la funcionalidad>).

La clase contenedora del *framework* JSF tiene la función de generar el código de las páginas clientes (<<construye>>). Los formularios componen las páginas clientes permitiendo que entre las páginas clientes y sus formularios, exista una relación de composición y a través de estos se muestran e introducen los datos necesarios. Dichos datos son enviados hacia el contenedor del *framework* JSF que construyó la página cliente asociada (<<envía>>). Las páginas clientes incluyen clases con ficheros JavaScript y librerías (<<incluyen>>).

Con el fin de representar las clases anteriormente mencionadas, se procede a usar la siguiente nomenclatura: frmsiaps<Nombre de la vista>.siaps, 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 tales como los *frameworks* JSF 1.2, EJB 3.0, JasperReport, JPA, Hibernate 3.3 y Seam 2.1.1. Además, se cuenta con un conjunto de clases entre las que se identifican librerías, JavaScript, JSF (*Servlets*), Seam (*Servlets*), Bitácora, entre otras.

3.3.2 Diagramas de clases del diseño. Gestionar Hoja de Discusión Diagnóstica.



3.3.3 Descripciones textuales

Capa de Presentación	
Nombre:	Propósito:
<div style="text-align: center;">  <p>listado.siaps</p> <p>Figura 1. listado.siaps</p> </div>	<p>Proveer la interacción con el usuario.</p>
Descripción:	
<p>La clase listado.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite obtener el listado de los pacientes. Posee un conjunto de validaciones en JavaScript que posibilita no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el <i>framework</i> JSF.</p>	
Nombre:	Propósito:
<div style="text-align: center;">  <p>frmsiaps</p> <p>Figura 2. 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</p>	

página cliente. Sus atributos son los elementos de entrada del formulario (inputText, textTarea, checkBox, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.

Nombre:	Propósito:
 <p>agregar.siaps</p> <p>Figura 3. agregar.siaps</p>	<p>Proveer la interacción con el usuario.</p>

Descripción:

La clase agregar.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite insertar todos los datos necesarios referentes a una hoja de discusión diagnóstica del paciente. 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>frmsiaps</p> <p>Figura 4. frmsiaps</p>	<p>Enviar los datos a las páginas servidoras.</p>

Descripción:

La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la

página cliente. Sus atributos son los elementos de entrada del formulario (inputText, textTarea, checkBox, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.

Nombre:

Propósito:



detalles.s_iaps

Figura 5. detalles.s_iaps

Proveer la interacción con el usuario.

Descripción:

La clase detalles.s_iaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite mostrar todos los datos referentes a la Hoja de Discusión Diagnóstica de un paciente. 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:



frmsiaps

Figura 6. 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. Sus atributos son los elementos de entrada del formulario (inputText, textTarea, checkBox, entre otros). No tienen operaciones, el método para el paso de los parámetros es

\$_POST.	
Nombre:	Propósito:
 <p>mpbuscar CIE.s siaps</p> <p>Figura 7. mpbuscarCIE.s siaps</p>	<p>Proveer la interacción con el usuario.</p>
Descripción:	
<p>La clase mpbuscarCIE.s iaps es una página web (ventana emergente) que se ejecuta del lado del cliente sobre un navegador. Permite obtener el listado de las enfermedades y subcategorías de la Clasificación Internacional de Enfermedades (CIE). Posee un conjunto de validaciones en JavaScript que permiten no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el <i>framework</i> JSF.</p>	
Nombre:	Propósito:
 <p>frmsiaps</p> <p>Figura 8. frmsiaps</p>	<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. Sus atributos son los elementos de entrada del formulario (inputText, textTarea, checkBox, entre otros). No tienen operaciones, el método para el paso de los parámetros es</p>	

\$_POST.	
Nombre:	Propósito:
 <p>mpbuscar CIAP.s siaps</p> <p>Figura 9. mpbuscarCIAP.s siaps</p>	<p>Proveer la interacción con el usuario.</p>
Descripción:	
<p>La clase mpbuscarCIAP.s siaps es una página web (ventana emergente) que se ejecuta del lado del cliente sobre un navegador. Permite obtener el listado de las enfermedades de la Clasificación Internacional para la Atención Primaria (CIAP). Posee un conjunto de validaciones en JavaScript que permiten no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el <i>framework</i> JSF.</p>	
Nombre:	Propósito:
 <p>frmsiaps</p> <p>Figura 10. frmsiaps</p>	<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. Sus atributos son los elementos de entrada del formulario (inputText, textTarea, checkBox, entre otros). No tienen operaciones y el método para el paso de los parámetros es</p>	

\$_POST.	
Capa de Negocio	
Nombre:	Propósito:
 CCDiscussionDiagEvolucion.java +insertarInstanciaEntidad() Figura11. CCDiscusionDiagEvolucion.java	Proveer una respuesta a las peticiones realizadas en la vista.
Descripción:	
La clase CCDiscusionDiagEvolucion.java se ejecuta del lado del servidor. Permite dar respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contiene. Se encarga de gestionar la acción de insertar, solicitada sobre la página cliente correspondiente. Hace uso del <i>framework</i> EJB que encapsula la lógica de negocio, integrándose con la vista a través del <i>framework</i> SEAM.	
Nombre:	Propósito:
 TbDiscussionDiagEvolucionList_smi.java Figura12. TbDiscussionDiagEvolucionList_smi .java	Proveer una respuesta a las peticiones realizadas en la vista.
Descripción:	
La clase TbDiscussionDiagEvolucionList_smi se genera de forma automática al realizarse la	

ingeniería inversa y se ejecuta del lado del servidor. Permite dar respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contiene. Se encarga de gestionar la acción de listar, solicitada 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:
 CLDiscussionDiagEvolucion.java	Proveer una respuesta a las peticiones realizadas en la vista

Figura13.

CLDiscussionDiagEvolucion.java

Descripción:

La clase CLDiscussionDiagEvolucion.java se ejecuta del lado del servidor. Hereda todas las funcionalidades de la clase autogenerada TbDiscussionDiagEvolucionList_smi.java. Se encarga de gestionar la acción de buscar o listar, solicitada sobre la página cliente correspondiente. Permite guardar los cambios originales de la misma, lo que evita que se pierdan al realizarse la ingeniería inversa. Hace uso del *framework* EJB que encapsula la lógica de negocio, integrándose con la vista a través del *framework* SEAM.

Nombre:	Propósito:
 CDDiscusionDiagEvolucion.java +insertarInstanciaEntidad()	Proveer una respuesta a las peticiones realizadas en la vista.

Figura14.

CDDiscusionDiagEvolucion.java

Descripción:	
<p>La clase CDDiscusionDiagEvolucion.java se ejecuta del lado del servidor. Permite dar respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contiene. Se encarga de gestionar la acción de mostrar, solicitada sobre la página cliente correspondiente. Hace uso del <i>framework</i> EJB que encapsula la lógica de negocio, integrándose con la vista a través del <i>framework</i> SEAM.</p>	
Capa de Datos	
Nombre:	Propósito:
<div style="border: 1px solid black; background-color: #d9e1f2; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center; margin: 0;">TbDiscusionDiagEvolucion_smi.java</p> <pre style="margin: 0;">-id : int -tbResumenIngreso : TbResumenIngreso_smi -medico : Medico_smi -fecha : date -resumen : String -evolucion : String</pre> </div> <p>Figura 15. TbDiscusionDiagEvolucion_smi.java</p>	<p>Proveer el mapeo con la base de datos.</p>
Descripción:	
<p>La clase TbDiscusionDiagEvolucion_smi.java se ejecuta del lado del servidor. Representa una tabla en el modelo de datos relacional que contiene la información referente a la Hoja de Discusión Diagnóstica. 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 <i>framework</i> Hibernate y JPA.</p>	
Nombre:	Propósito:

<pre>TrDiscusionDiagPsalud_smi.java -id : int -tnCieSubcategoria : TnCieSubcategoria_smi -tnCieEnfermedad : TnCieEnfermedad_smi -tnCpaps : TnCpaps_smi -tbDiscusionDiagEvolucion : TbDiscusionDiagEvolucion_smi -tnCiap : TnCiap_smi</pre>	Proveer el mapeo con la base de datos.
Descripción:	
La clase TrDiscusionDiagPsalud_smi.java se ejecuta del lado del servidor. Representa una tabla en el modelo de datos relacional que contiene la información referente a los problemas de salud asociados a una discusión diagnóstica 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 <i>framework</i> Hibernate y JPA.	

En el capítulo se pudo constatar cómo el diseño del sistema contribuye a tener una visión más cercana al producto que se quiere lograr a partir de los diagramas y sus descripciones, permitiendo que se entienda la estructura y las diversas interacciones entre las clases. De esta forma, se dejan las condiciones creadas para realizar una implementación que permita desarrollar la aplicación de acuerdo a las exigencias del cliente, la cual debe ser capaz de garantizar un apego a los criterios de diseño definidos.

Capítulo 4. Implementación

En el capítulo se tratan los aspectos correspondientes a la fase de Implementación. Se explica cómo se implementaron en términos de componentes, las clases y subsistemas obtenidos en el Diseño. Se argumenta la necesidad de integración del módulo Salud Materno-Infantil con otros sistemas y se representa la misma, se muestra el Diagrama de Despliegue y se exponen los estándares de diseño, codificación y tratamiento de errores aplicados en el desarrollo del componente web.

4.1 Integración con otros sistemas.

En la Informática resulta poco usual concebir un sistema de forma aislada, es decir, sin la interacción con otros sistemas para la prestación de servicios entre ellos. El componente web que se desarrolla con esta investigación, perteneciente al módulo Salud Materno-Infantil, se integra al SIAPS como plataforma única para la gestión, procesamiento y transmisión de la información clínica en el Sistema Nacional de Salud; a través de los siguientes módulos:

➤ **Módulo Clínico Quirúrgico del SIAPS**

El componente web para el módulo Salud Materno-Infantil hace peticiones al módulo Clínico Quirúrgico para obtener la información referente a los problemas de salud de un paciente.

➤ **Módulo de Enfermería del SIAPS**

El componente web para el módulo Salud Materno-Infantil consulta la información referente a la vacunación de cada paciente así como a las pruebas citológicas, contenida en las tablas de la base de datos pertenecientes al módulo de Enfermería.

➤ **Módulo de Medicina Familiar del SIAPS**

El componente web para el módulo Salud Materno-Infantil actualiza el grupo dispensarial de la embarazada cuando es captada y asigna la Historia de Salud Familiar correspondiente al recién nacido que se capta, para lo cual se necesita la integración con el módulo de Medicina Familiar.

➤ **Módulo de Configuración del SIAPS**

El módulo Salud Materno-Infantil se encuentra inevitablemente integrado al módulo Configuración del SIAPS pues éste último se encarga de gestionar todas las configuraciones de temas, roles, usuarios,

funcionalidades, módulos, unidades de salud y áreas de salud, personal de salud, ubicación, gestión de codificadores o estándares internacionales, nomencladores médicos, las poblaciones o zonas de APS, así como la seguridad del SIAPS.

4.2 Implementación.

La Implementación comienza con el resultado del Diseño y se define cómo se organizan las clases y objetos en términos de componentes, cuáles nodos se utilizarán y la ubicación en ellos, de los componentes y la estructura de capas de la aplicación. Uno de los principales propósitos de esta fase consiste en desarrollar la arquitectura y el sistema como un todo.

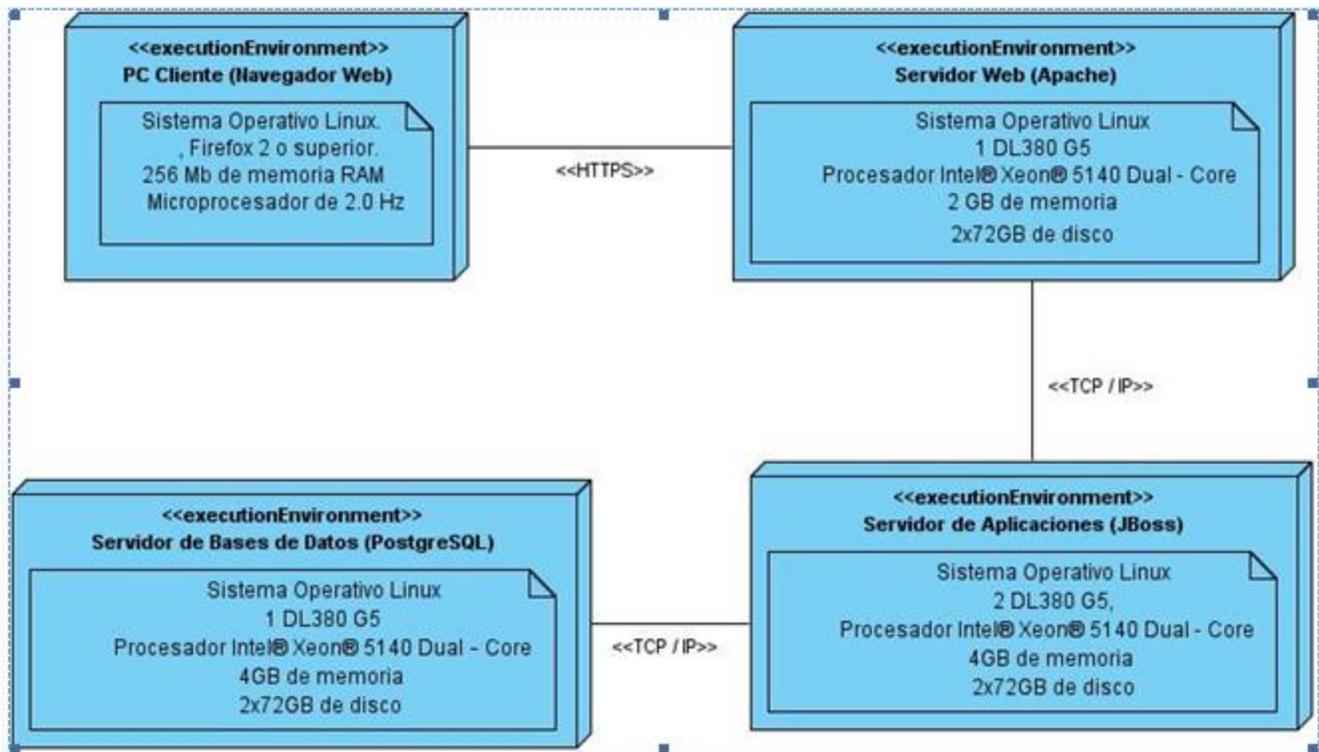
4.2.1. Descripción de clase.

La clase **CCDiscusionDiag.java** se encarga de gestionar la información perteneciente a la Hoja de Discusión Diagnóstica como parte del proceso de atención prenatal de la embarazada en el hogar materno. Tiene como atributos principales el **resumen** (recoge las consideraciones emitidas por el médico durante la discusión diagnóstica), la **fecha** (recoge la fecha en la cual es realizada la discusión) y la **evolución** (se refiere a la información que expresa el estado de la paciente transcurrido un período de tiempo). Cuenta con el método **insertarInstanciaEntidad()** el cual se encarga de insertar en la base de datos la información contenida en la hoja antes mencionada. Hace uso de los métodos de la interfaz **EntityManger** para la persistencia y posterior utilización de los datos.

De igual forma contiene el método **setListaidEnf(param)** que tiene la responsabilidad de agregar a una lista, las enfermedades seleccionadas dentro de la Clasificación Internacional de Enfermedades (CIE), para luego ser insertadas en la base de datos. El método garantiza que, de ser seleccionado algún problema dentro de la clasificación CIE no se seleccionen problemas dentro de la Clasificación Internacional para la Atención Primaria (CIAP), lo cual constituye una regla del negocio definida previamente.

4.3 Diagrama de Despliegue.

Los diagramas de despliegue muestran la disposición física de los distintos nodos que entran en la composición de un sistema y el reparto de los programas ejecutables sobre estos nodos. Los nodos representan recursos de cómputo: procesadores o dispositivos de hardware. Los nodos se interconectan mediante soportes bidireccionales (en principio) que pueden a su vez estereotiparse.



4.4 Estándares de codificación y tratamiento de excepciones.

Con el propósito de que exista homogeneidad entre las aplicaciones que se encuentran integradas al Sistema Integral para la Atención Primaria de Salud, se han definido una serie de estándares tanto para el diseño como para la codificación, además de establecer cómo se hará el tratamiento de errores al componente para que los códigos fuentes de las aplicaciones y los mensajes que se emitan mantengan una uniformidad.

4.4.1 Estándares de codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre al aspecto práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de *software*, es necesario establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada.

Las convenciones de código o estándares de codificación son importantes para los programadores por un gran número de razones:

- El 80% del coste del código de un programa va a su mantenimiento.
- Casi ningún *software* es mantenido toda su vida por el autor original.
- Las convenciones de código mejoran la lectura del *software* lo que permite entender código nuevo de manera más óptima y rápida.
- Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.

Sobre los estándares de codificación definidos para el SIAPS:

- Idioma: Se debe utilizar como idioma el español, las palabras no se acentuarán.
- Comentarios, líneas y espacios en blanco.

Todos los ficheros fuentes deben comenzar con un comentario en el que se lista el nombre de la clase, información de la versión, fecha y copyright.

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 la siguiente circunstancia:

- 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 clases o métodos 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 de peso "\$", aunque ambos están permitidos por el lenguaje.
- 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 aparecer totalmente en mayúscula 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 mayúscula. 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 forman en mayúscula.

4.4.2 Tratamiento de excepciones.

Durante el tiempo de ejecución de un sistema pueden fracasar diferentes rutinas, es a esto a lo que comúnmente se le llama excepción. Las excepciones son situaciones anómalas que requieren un

tratamiento especial. No tienen por qué ser errores. Si se consigue dominar su programación, la calidad de las aplicaciones que se desarrollen aumentará considerablemente. Con ellas se obtiene un sistema más robusto y fiable.

En el sistema propuesto se utilizan todas las facilidades que brinda la plataforma para el tratamiento de excepciones. Para cada fragmento de código donde se espere una situación anómala, se definen las excepciones correspondientes para luego ser tratadas evitando la interrupción del sistema. De igual forma se emplean excepciones predefinidas por los marcos de trabajo que se utilizan 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 el envío de información correcta, al servidor.

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.

La implementación del sistema que se ha descrito en el presente capítulo, ha permitido demostrar que es posible lograr una aplicación con las características que se determinaron previamente. Puede afirmarse que después de la implementación, ocurre un considerable acercamiento al cumplimiento del objetivo general por el que ha surgido esta investigación. Se expuso una idea acabada de los elementos necesarios a tener en cuenta para el despliegue de la aplicación informática y se pudo comprender que el componente desarrollado no es un ente aislado sino que se integra con otros sistemas para el logro de prestaciones de servicios más integrales que garanticen una atención de excelencia al paciente.

Conclusiones

Luego de realizada la investigación, se pudo arribar a las siguientes conclusiones:

- En la atención a la salud del niño y la embarazada resulta esencial el enfoque preventivo de las acciones, para lo cual se precisa del registro y análisis de grandes volúmenes de información.
- La indagación sobre aplicaciones informáticas desarrolladas para el nivel primario de atención, tanto en Cuba como en el resto del mundo, permitió identificar la necesidad de realización de la presente investigación; al no contarse con un componente web que gestione integralmente la información generada en los procesos de captación de la embarazada y el recién nacido, ingreso y atención prenatal de la gestante en el hogar materno en la APS.
- El conjunto de tecnologías empleadas, sobre la base de la arquitectura y patrones de diseño descritos, facilitaron la obtención de los artefactos requeridos e incorporaron calidad y buenas prácticas al resultado de la investigación.
- La identificación del flujo de negocio asociado a cada una de las áreas relacionadas con el campo de acción de la investigación, permitió caracterizar de forma detallada el objeto de automatización.
- El componente web perteneciente al módulo Salud Materno-Infantil del Sistema Integral para la Atención Primaria de Salud desarrollado, permite gestionar con mayor facilidad la información asociada a los procesos de captación de la embarazada y el recién nacido, ingreso y atención prenatal de la gestante en el hogar materno.

Recomendaciones

Para contribuir al éxito en la continuidad de la investigación, se hacen las siguientes recomendaciones:

- Desarrollar el resto de las funcionalidades del módulo Salud Materno-Infantil del Sistema Integral para la Atención Primaria de Salud.
- Concluir la informatización del proceso atención prenatal de la gestante en el hogar materno.
- Investigar con mayor detenimiento, sobre las prestaciones y particularidades del Sistema de Gestión Hospitalaria y otros sistemas informáticos de gestión de información en el sector de la salud, para lograr la necesaria integración con éstos.

Referencias Bibliográficas

1. **Hardy, Hellen y otros.** Biblioteca Virtual en Salud. *Masculinidad y Género*. [En línea] 2001. [Citado el: 10 de noviembre de 2010] http://bvs.sld.cu/revistas/spu/vol27_2_01/spu01201.htm.
2. **Elizalde Cremonte, Alejandra Dra.** Universidad Nacional del Nordeste. *Historia Clínica Perinatal. Control Prenatal. Diagnóstico de Riesgo*. [En línea] [Citado el: 10 de noviembre de 2010.] <http://med.unne.edu.ar/posgrado/cursomedgral/clases/30092006ec.pdf>.
3. scielo. *Resúmenes sobre salud materno infantil*. [En línea] 2009. [Citado el: 12 de noviembre de 2010.] http://scielo.sld.cu/scielo.php?pid=S0864-21252009000300016&script=sci_arttext.
4. **Halkyer, Percy Dr.** *Conceptos sobre temas de salud y propuesta de texto*. [En línea] 2006. [Citado el: 17 de enero de 2011.] <http://saludpublica.vbsp.org.bo>
5. *Atención Primaria de Salud*. [En línea] [Citado el: 10 de noviembre de 2010.] <http://www.argentina.gov.ar/argentina/portal/paginas.dhtml?pagina=113>.
6. **Delgado Ramos, Ariel Msc. y otros.** *Informática en la Salud Pública Cubana*. [En línea] [Citado el: 15 de enero de 2011.] http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm.
7. **Pompa Sourd, Frank Ing. y otros.** *Sistema Informático para la Atención Primaria de Salud*. [En línea] [Citado el: 18 de noviembre de 2010.] <http://www.hab2001.sld.cu/arrepdf/00196.pdf>.
8. Alert Life Sciences Computing. *ALERT@ CARE es seleccionado por seis ciudades en Brasil*. [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.alert-online.com/es/news/company/alert-care-es-seleccionado-por-seis-ciudades-en-brasil>.
9. *Definición de Internet*. [En línea] [Citado el: 20 de noviembre de 2010.] <http://definicion.de/internet/>.
10. La Definición de Software Libre. [En línea] [Citado el: 20 de enero de 2011.] <http://www.gnu.org/philosophy/free-sw.es.html>.
11. **Casanovas, Josep.** *Usabilidad y arquitectura del software*. [En línea] 9 de septiembre de 2004. [Citado el: 24 de enero de 2011.] <http://www.desarrolloweb.com/articulos/1622.php>.
12. La arquitectura cliente servidor. [En línea] [Citado el: 18 de enero de 2011.] <http://temariotic.wikidot.com/la-arquitectura-cliente-servidor>.

13. *Modelo Vista Controlador – Definición y Características*. [En línea] 13 de noviembre de 2010. [Citado el: 24 de enero de 2011.] <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>.
14. **Casal Terreros, Julio**. Desarrollo de software basado en componentes. [En línea] [Citado el: 20 de enero de 2011] <http://msdn.microsoft.com/es-es/library/bb972268.aspx>.
15. **Vallecillo, Antonio**. El-Golem. Qué es la Programación Orientada a Componentes. [En línea] 30 de agosto de 2010. [Citado el: 24 de enero de 2011.] <http://www.el-golem.com.ar/index.php/poc/22-que-es-la-programacion-orientada-a-componentes>.
16. La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. [En línea] diciembre 2006. [Citado el: 20 de enero de 2011.] <http://www.google.com/cu/#hl=es&biw=1024&bih=576&q=arquitectura+orientada+a+servicios&aq=f&aqi=g10&aql=&oq=&fp=71d0260c96497ec0>.
17. Software: Beneficios de la Arquitectura Orientada a Servicios. [En línea] febrero de 2008. [Citado el: 3 de diciembre de 2010.] <http://www.channelplanet.com/?idcategoria=19497>.
18. *¿Qué Es Un Servidor De Aplicaciones?* [En línea] 25 de diciembre de 2007. [Citado el: 29 de enero de 2011.] <http://www.editum.org/Que-Es-Un-Servidor-De-Aplicaciones-p-473.html>.
19. ¿ Qué es un Sistema Gestor de Base de Datos o SGBD. [En línea] [Citado el: 3 de diciembre de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
20. **Ballina Sánchez, Francisco Salvador**. Taller de Base de Datos. [En línea] [Citado el: 20 de enero de 2011.] <http://www.slideshare.net/Mayra00/taller-de-base-de-datos>.
21. **González, Carlos D**. Curso Base de Datos PostgreSQL, SQL avanzado y PHP. [En línea] febrero 2011. [Citado el: 20 de marzo de 2011.] <http://www.usabilidadweb.com.ar/postgre.php>.
22. Tutorial de JavaServer Faces. [En línea] [Citado el: 25 de noviembre de 2010.] <http://www.sicuma.uma.es/sicuma/Formacion/documentacion/JSF.pdf>.
23. **Dorr, Paul M**. Desarrollando Ideas. Empezando Con JSF (Java Server Faces). [En línea] 25 de mayo de 2008. [Citado el: 18 de febrero de 2011.] <http://desarrollandoideas.com.ar/2008/05/25/empezando-con-jsf-java-server-faces/>.

24. AJAX Ya. *Qué es AJAX?* [En línea] [Citado el: 22 de enero de 2010.]
<http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>.
25. AJAX Ya. *Ventajas y desventajas de AJAX.* [En línea] [Citado el: 26 de noviembre de 2010.]
<http://www.ajaxya.com.ar/temarios/descripcion.php?cod=34&punto=2>.
26. Concepto de CSS. [En línea] [Citado el: 20 de enero de 2011.]
<http://www.creargratisunapaginaweb.com/CSS-Hojas-de-estilo-en-cascada/Concepto-de-CSS->
27. Definición de html. [En línea] [Citado el: 26 de noviembre de 2010] <http://definicion.de/html/>.
28. ¿Qué es XML? - Definición de XML. [En línea] [Citado el: 25 de noviembre de 2010.]
<http://www.masadelante.com/faqs/xml>.
29. Definición de proceso. [En línea] [Citado el: 20 de enero de 2011.] <http://definicion.de/proceso/>.
30. ¿Qué es BPM? [En línea] 3 de noviembre de 2009. [Citado el: 28 de noviembre 2010.] <http://www.club-bpm.com/ApuntesBPM/ApuntesBPM01.pdf>.
31. Definición de UML. [En línea] [Citado el: 3 de diciembre de 2010.]
<http://www.mastermagazine.info/termino/7006.php>.
32. **Paredes, Adrián M.** Blog de Informática, Arquitectura, Diseño. *Patrones Enterprise (parte 2).* [En línea] 9 de noviembre de 2010. [Citado el: 20 de marzo de 2011.]
<http://elblogdelfrasco.blogspot.com/2008/12/patrones-enterprise-parte-2.html>.
33. **López Hernández, Yurién Ing.** *Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS.* La Habana : s.n., 2010.

Bibliografía

1. **Hardy, Hellen y otros.** Biblioteca Virtual en Salud. Masculinidad y Género. [En línea] 2001. [Citado el: 10 de noviembre de 2010] http://bvs.sld.cu/revistas/spu/vol27_2_01/spu01201.htm.
2. **Elizalde Cremonte, Alejandra Dra.** Universidad Nacional del Nordeste. Historia Clínica Perinatal. Control Prenatal. Diagnóstico de Riesgo. [En línea] [Citado el: 10 de noviembre de 2010.] <http://med.unne.edu.ar/posgrado/cursomedgral/clases/30092006ec.pdf>.
3. **scielo.** Resúmenes sobre salud materno infantil. [En línea] 2009. [Citado el: 12 de noviembre de 2010.] http://scielo.sld.cu/scielo.php?pid=S0864-21252009000300016&script=sci_arttext.
4. **Halkyer, Percy Dr.** Conceptos sobre temas de salud y propuesta de texto. [En línea] 2006. [Citado el: 17 de enero de 2011.] <http://saludpublica.vbsp.org.bo>
5. Atención Primaria de Salud. [En línea] [Citado el: 10 de noviembre de 2010.] <http://www.argentina.gov.ar/argentina/portal/paginas.dhtml?pagina=113>.
6. **Delgado Ramos, Ariel Msc.** y otros. Informática en la Salud Pública Cubana. [En línea] [Citado el: 15 de enero de 2011.] http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm.
7. **Pompa Sourd, Frank Ing.** y otros. Sistema Informático para la Atención Primaria de Salud. [En línea] [Citado el: 18 de noviembre de 2010.] <http://www.hab2001.sld.cu/arrepdf/00196.pdf>.
8. Alert Life Sciences Computing. ALERT® CARE es seleccionado por seis ciudades en Brasil. [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.alert-online.com/es/news/company/alert-care-es-seleccionado-por-seis-ciudades-en-brasil>.
9. Definición de Internet. [En línea] [Citado el: 20 de noviembre de 2010.] <http://definicion.de/internet/>.
10. La Definición de Software Libre. [En línea] [Citado el: 20 de enero de 2011.] <http://www.gnu.org/philosophy/free-sw.es.html>.
11. Casanovas, Josep. Usabilidad y arquitectura del software. [En línea] 9 de septiembre de 2004. [Citado el: 24 de enero de 2011.] <http://www.desarrolloweb.com/articulos/1622.php>.
12. La arquitectura cliente servidor. [En línea] [Citado el: 18 de enero de 2011.] <http://temariotic.wikidot.com/la-arquitectura-cliente-servidor>.

13. Modelo Vista Controlador – Definición y Características. [En línea] 13 de noviembre de 2010. [Citado el: 24 de enero de 2011.] <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>.
14. Casal Terreros, Julio. Desarrollo de software basado en componentes. [En línea] [Citado el: 20 de enero de 2011] <http://msdn.microsoft.com/es-es/library/bb972268.aspx>.
15. **Vallecillo, Antonio**. El-Golem. Qué es la Programación Orientada a Componentes. [En línea] 30 de agosto de 2010. [Citado el: 24 de enero de 2011.] <http://www.el-golem.com.ar/index.php/poc/22-que-es-la-programacion-orientada-a-componentes>.
16. La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. [En línea] diciembre 2006. [Citado el: 20 de enero de 2011.] <http://www.google.com/cu/#hl=es&biw=1024&bih=576&q=arquitectura+orientada+a+servicios&aq=f&aqi=g10&aql=&oq=&fp=71d0260c96497ec0>.
17. Software: Beneficios de la Arquitectura Orientada a Servicios. [En línea] febrero de 2008. [Citado el: 3 de diciembre de 2010.] <http://www.channelplanet.com/?idcategoria=19497>.
18. ¿Qué Es Un Servidor De Aplicaciones? [En línea] 25 de diciembre de 2007. [Citado el: 29 de enero de 2011.] <http://www.editum.org/Que-Es-Un-Servidor-De-Aplicaciones-p-473.html>.
19. ¿ Qué es un Sistema Gestor de Base de Datos o SGBD. [En línea] [Citado el: 3 de diciembre de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
20. **Ballina Sánchez, Francisco Salvador**. Taller de Base de Datos. [En línea] [Citado el: 20 de enero de 2011.] <http://www.slideshare.net/Mayra00/taller-de-base-de-datos>.
21. **González, Carlos D**. Curso Base de Datos PostgreSQL, SQL avanzado y PHP. [En línea] febrero 2011. [Citado el: 20 de marzo de 2011.] <http://www.usabilidadweb.com.ar/postgre.php>.
22. Tutorial de JavaServer Faces. [En línea] [Citado el: 25 de noviembre de 2010.] <http://www.sicuma.uma.es/sicuma/Formacion/documentacion/JSF.pdf>.
23. **Dorr, Paul M**. Desarrollando Ideas. Empezando Con JSF (Java Server Faces). [En línea] 25 de mayo de 2008. [Citado el: 18 de febrero de 2011.] <http://desarrollandoideas.com.ar/2008/05/25/empezando-con-jsf-java-server-faces/>.

24. AJAX Ya. Qué es AJAX? [En línea] [Citado el: 22 de enero de 2010.] <http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>.
25. AJAX Ya. Ventajas y desventajas de AJAX. [En línea] [Citado el: 26 de noviembre de 2010.] <http://www.ajaxya.com.ar/temarios/descripcion.php?cod=34&punto=2>.
26. Concepto de CSS. [En línea] [Citado el: 20 de enero de 2011.] <http://www.creargratisunapaginaweb.com/CSS-Hojas-de-estilo-en-cascada/Concepto-de-CSS->
27. Definición de html. [En línea] [Citado el: 26 de noviembre de 2010] <http://definicion.de/html/>.
28. ¿Qué es XML? - Definición de XML. [En línea] [Citado el: 25 de noviembre de 2010.] <http://www.masadelante.com/faqs/xml>.
29. Definición de proceso. [En línea] [Citado el: 20 de enero de 2011.] <http://definicion.de/proceso/>.
30. ¿Qué es BPM? [En línea] 3 de noviembre de 2009. [Citado el: 28 de noviembre 2010.] <http://www.club-bpm.com/ApuntesBPM/ApuntesBPM01.pdf>.
31. Definición de UML. [En línea] [Citado el: 3 de diciembre de 2010.] <http://www.mastermagazine.info/termino/7006.php>.
32. **Paredes, Adrián M.** Blog de Informática, Arquitectura, Diseño. Patrones Enterprise (parte 2). [En línea] 9 de noviembre de 2010. [Citado el: 20 de marzo de 2011.] <http://elblogdelfrasco.blogspot.com/2008/12/patrones-enterprise-parte-2.html>.
33. **López Hernández, Yurién Ing.** Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alasSIAPS. La Habana : s.n., 2010.
34. **Villanueva Rozas, Luis Dr.** Niveles de Atención en Salud. [En línea] [Citado el: 20 de noviembre de 2010.] <http://www.google.com.cu/#hl=es-419&biw=1264&bih=676&q=niveles+de+atencion+de+la+salud&aq=o&aql=&oq=&fp=1&cad=b>
35. **Díaz, Rodrigo Dr.** El rol de la atención primaria en los seguros de salud. [En línea] [Citado el: 10 de enero de 2011.] <http://www.saluddealtura.com/seguros-salud-ecuador/sistemas-seguros-salud/atencion-primaria-seguros-salud/>.
36. Nutrinet. Programa Nacional Materno Infantil. [En línea] [Citado el: 11 de enero de 2011.] <http://cuba.nutrinet.org/servicios/noticias-articulos/articulos-recomendados/423-programa-nacional-materno-infantil>.

37. **Delgado García, Gregorio Dr.** Biblioteca Virtual en Salud. Los hogares maternos: su fundación en Cuba y objetivos propuestos desde su creación. [En línea] [Citado el: 12 de enero de 2011.] http://bvs.sld.cu/revistas/his/his_%20101/his12101.htm.
38. EcuRed. [En línea] [Citado el: 12 de enero de 2011.] http://www.ecured.cu/index.php/Consultorio_M%C3%A9dico_de_la_Familia_Mariana_Grajales_%28Cauto_Cristo%29.
39. Brasilia tiene uno de Mayores Proyectos de Informatización de Salud Pública del Mundo. [En línea] [Citado el: 15 de enero de 2011.] <http://www.intersystemschile.cl/iscch/publicador/imgpublicador/casocli41.pdf>.
40. APUS. Sistema Informático para la Gestión Médica en la APS. [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.sld.cu/instituciones/cedisap/atepri1.htm>.
41. CAIS 2010 Congreso Argentino de Informática y Salud. [En línea] [Citado el: 15 de noviembre de 2010.] http://www.managementensalud.com.ar/Eventos2010/CAIS/Programa_CAIS_39AIO.pdf.
42. **Montesinos Alonso, Amparo.** Los profesionales sanitarios de atención primaria ante las nuevas tecnologías. [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.medynet.com/elmedico/publicaciones/centrosalud6/353-356.pdf>.
43. **Blanquer Gregori, José J y otros.** *La informatización, un elemento más en la formación del médico de familia.* [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.svmfyc.org/Revista/05/originales2.asp>.
44. *Atención Primaria en el INSALUD: Diecisiete años de experiencia.* [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.ingesa.msc.es/estadEstudios/documPublica/pdf/Completo.pdf>.
45. **Pompa Sourd, Frank Ing. y otros.** *Sistema Informático para la Atención Primaria de Salud.* [En línea] [Citado el: 18 de noviembre de 2010.] <http://www.hab2001.sld.cu/arrepdf/00196.pdf>.
46. **Curioso, Walter H. y otros.** *Historia Clínica Electrónica. Experiencia en un Hospital Nacional. Satisfacción por parte del personal de salud y pacientes.* [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.enlacesmedicos.com/hcefulttextcurioso.pdf>.

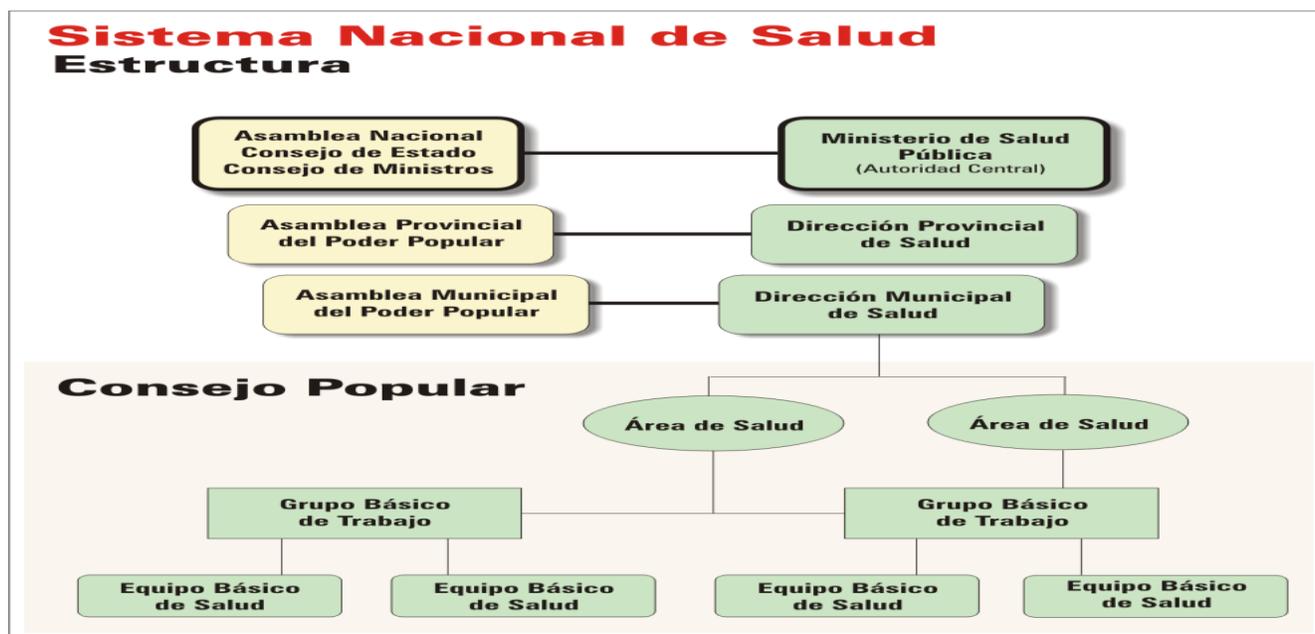
47. **Herrero Santoja, Joaquín.** Informática en salud 2007. *SLD102-La experiencia Kewan - Cosmosalud.* [En línea] [Citado el: 15 de noviembre de 2010.] <http://www.informatica2007.sld.cu/Members/jherreroypf/la-experiencia-kewan-cosmosalud/>.
48. **Tackett, Jack.** Características generales de Unix/Linux. [En línea] [Citado el: 20 de enero de 2011.] <http://www.fismat.umich.mx/~elizalde/tesis/node24.html>.
49. Arquitectura 3 Capas. [En línea] 6 de febrero de 2009. [Citado el: 3 de diciembre de 2010.] <http://kernelerror.net/programacion/php/arquitectura-3-capas/>.
50. Desarrollo de Software basado en Componentes. [En línea] [Citado el: 25 de enero de 2011.] <http://www.lambdasi.com.ar/textocomp.asp?id=9>.
51. JBoss Application Server. [En línea] [Citado el: 21 de enero de 2011.] http://www.dosideas.com/wiki/JBoss_Application_Server.
52. ¿Qué es PostgreSQL? [En línea] [Citado el: 22 de enero de 2011.] http://danielpecos.com/docs/mysql_postgres/x15.html.
53. **Sánchez Suárez, José Manuel.** Introducción a RichFaces. [En línea] 2 de enero de 2010. [Citado el: 20 de enero de 2011.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflIntro>.
54. **Pérez Valdés, Damián.** maestros del web. *¿Qué es Javascript?* [En línea] [Citado el: 20 de enero de 2011.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
55. Framework de desarrollo AMAP. [En línea] 2008. [Citado el: 25 de noviembre de 2010.] <http://amap.cantabria.es/confluence/display/BASE/Facelets>.
56. ¿Qué es CSS? [En línea] [Citado el: 20 de noviembre de 2011.] <http://www.maestrosdelweb.com/editorial/introcss/>.
57. ¿Qué es xhtml? . [En línea] [Citado el: 18 de enero de 2011.] <http://www.chesco.info/curso/xhtmll.htm>.

Glosario de Términos

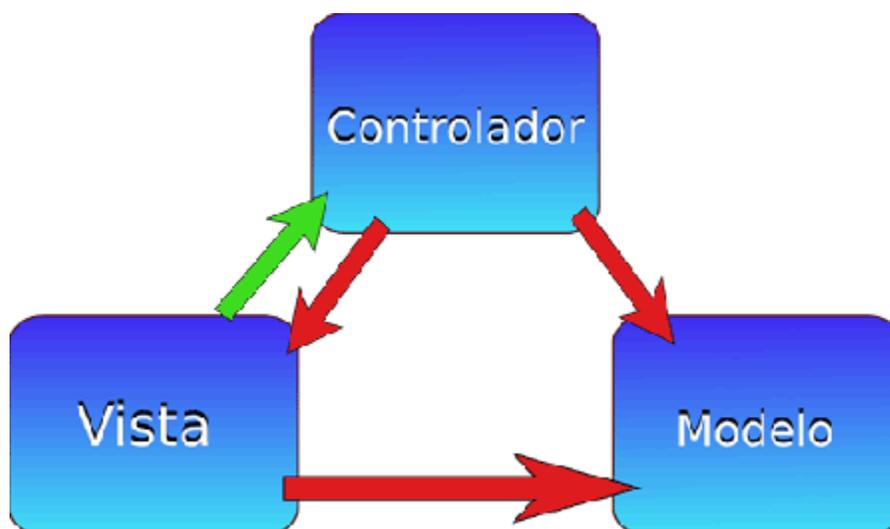
- 1 Diagnóstico: en la medicina, un diagnóstico es el acto de conocer la naturaleza de una enfermedad a través de la observación de sus síntomas y signos. También es el nombre que recibe la calificación que da el médico a la enfermedad según los signos que advierte.
- 2 Gestante: sinónimo de embarazada.
- 3 Indentación: anglicismo (de la palabra inglesa *indentation*) de uso común en Informática que significa mover un bloque de texto hacia la derecha, insertando espacios o tabuladores para separarlo del texto adyacente.
- 4 Morbilidad: se refiere a la cantidad de personas o individuos que son considerados enfermos o que son víctimas de enfermedad en un espacio y tiempo determinados.
- 5 Obstétrico: referente a la Obstetricia la cual es una rama de las ciencias de la salud que se ocupa del embarazo, parto y puerperio, comprendiendo también los aspectos psicológicos y sociales de la maternidad.
- 6 Patología: enfermedad física o mental.
- 7 Precoz: se refiere a lo prematuro o temprano.
- 8 Prenatal: referente al período de tiempo de embarazo que culmina justamente con el parto.
- 9 Púérpera: se refiere a la mujer que se encuentra transitando por el puerperio, período que transcurre desde el parto hasta que los órganos genitales y el estado general de la mujer recuperan la situación que tenían antes del embarazo.

Anexos

Anexo 1: Estructura del Sistema Nacional de Salud



Anexo 2: Esquema del patrón de arquitectura Modelo Vista Controlador.



Anexo 3: Interfaces correspondientes al proceso de captación del recién nacido

Captación del recién nacido Buscar...

Datos generales **Examen físico**

Datos generales

Nombre(s):	Primer apellido:	Segundo apellido:
<input type="text"/>	<input type="text"/>	<input type="text"/>
No. CI:	Dirección:	Teléfono:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Área salud:	Consultorio:	No. HCl:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Fecha nacimiento:	Sexo:	Grupo sanguíneo:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Factor RH:	Color piel:	No. Historia de Salud Familiar:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Fecha actual 22/01/11 15:00	Fecha de captación:	
<input type="text"/>	<input type="text"/>	

Datos del padre



Datos de la madre



Captación del recién nacido Buscar..

Datos generales **Examen físico**

Generales **Regional** **Por sistemas**

Aspecto del paciente >

Piel >

Temperatura, fascias y TSC >

Fanículo adiposo >

Mucosas >

Observaciones

Especifique:

Aceptar **Cancelar**

Anexo 4: Interfaces correspondientes al proceso de captación de la embarazada

Buscar paciente

Criterios de búsqueda

Nombre: Primer apellido: Segundo apellido:

Carné de identidad: Sexo:

 Exportar

No. HCl	No. CI	Nombre y apellidos de la paciente	Sexo
112009061212099	22222222222	Epitafio Estrada Mayor	M
112009061212001	22222222222	Matha Estrada Jimenez	F
112009061212078	77777777777	Marlene Estrada Mayor	F
112009060312001	11111111111	Johander Leon Garces	M
112009061212023	88101801221	Alberto Ortega Palacios	M
112009061212070	22222222222	Sofia Adulto Mayor	-
112009061212045	22222222222	Yaneslaser Batista Barro	M
112009061212001	44444444444	Humberto Ruiz Salcedo	M
112009061212078	22222222222	Joseta Adulto Mayor	-
112009061212078	22222222222	Frumencia Adulto Mayor	F

◀ 1 ▶

1 10/24

Captar embarazada

Datos generales de la paciente



Nombre: Sofía Primer apellido: Adulto Segundo apellido: Mayor
 Dirección: Cepero Bonilla Zona 1 Primer apellido: Manzana23422
 Policlínico: Policlínico 1ro de Mayo Hospital: Hermanos Ameijeiras Unidad de salud: Hogar Materno Che Guevara
 Médico: Enrique Castillo Esposo: Reinedio Vega Edad: 96 año(s)
 Escolaridad: Pre Universitario Ocupación: Estado civil:

Datos del interrogatorio

Consejos profilácticos

Interrogatorio

Generales

Embarazo Actual

Fecha captación:

Talla:

Valoración ponderal inicial:

Fórmula menstrual (duración):

Ver tarjeta de vacunación

Peso inicial:

IMC:

Menstruación regular:

Toma de muestra de colpocitología:

Peso habitual:

Peso ideal:

Fórmula menstrual (ciclo):

Ver tarjeta de citología

Resumen de interrogatorio

Actualización de grupo dispensarial

Aceptar

Cancelar

Anexo 5: Interfaces correspondientes al proceso de ingreso de la embarazada en el hogar materno.

Nuevo Ingreso Q Buscar...

Datos generales de la paciente

Nombre: Reynier Rodriguez Ronquillo Carné identidad: 86041020423 Edad: 25 año(s)
 Sexo: M Color de la piel: blanco Estado conyugal: Pendiente BD
 Ocupación: Centro de trabajo: Pendiente BD Dirección del centro: PendienteBD

Resumen de ingreso Enfermedad actual Examen físico

Lugar de nacimiento

Lugar de residencia

Datos de los padres

Avisar en caso de emergencia

Ingreso actual

Resumen historia de la enfermedad actual

Datos de interés del examen físico

Complementarios

Evolución y tratamiento

Diagnósticos y operaciones

Causa de accidente o traumatismo

Datos del egreso

Aceptar Cancelar

Nuevo Ingreso Q Buscar...

Datos generales de la paciente

Nombre: Reynier Rodriguez Ronquillo Carné identidad: 86041020423 Edad: 25 año(s)
 Sexo: M Color de la piel: blanco Estado conyugal: Pendiente BD
 Ocupación: Centro de trabajo: Pendiente BD Dirección del centro: PendienteBD

Resumen de ingreso Enfermedad actual Examen físico

Historia enf. actual

Motivo de ingreso Antecedentes patológicos Intervenciones quirúrgicas Interrogatorio

Listado de problemas de salud

+ Búsqueda en CIE-10 + Búsqueda en CIAP

Listado de problemas de salud

No se han seleccionado problemas de salud.

Datos ginecológicos

Hábitos tóxicos

Aceptar Cancelar

Anexo 6: Interfaces correspondientes al proceso de atención prenatal de la gestante en el hogar materno

Listado de pacientes ingresadas Q Buscar...

Criterios de búsqueda

Numero H. Clínica

 Exportar

Listado de pacientes

Numero H. Clínica	Cama	Nombre	
6565556565	3	Reynier Rodriguez Ronquillo	 
33333333333	2	Reynier Rodriguez Ronquillo	 
22222222222	1	Reynier Rodriguez Ronquillo	 

Atender paciente Q Buscar...

Datos generales de la paciente

Nombre: Mairéys Mendoza Santana. Carné identidad: 45052145632. Hrc. historia clínica: 22543263900745
 Servicio: Obstetricia. Ubicación: SP3HP4. Cama: C3

 Datos del ingreso
 Discusión diagnóstica
 Hoja de Indicaciones médicas
 Data ta

 Hoja de signos vitales
 Hoja de evaluación para especialistas
 Hoja de complementarios

 Hoja de balance nutricional
 Hoja de evolución diaria
 Hoja de controles obstétricos

Listado de discusiones diagnósticas Q Buscar...

Datos generales de la paciente

Nombre: Mairéys Mendoza Santana. Carné identidad: 84523657412. Número de historia clínica: 5648087541236
 Servicio: Obstetricia. Ubicación: S1HP2. Cama: C1

Criterios de búsqueda

Fecha

 Exportar  Agregar

Listado de discusiones diagnósticas realizadas

Fecha	Realizado por
2010-01-15	Ramon de la O