

Universidad de las Ciencias Informáticas Facultad 7



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

Desarrollo del Módulo Enfermería del Subsistema Web del
Sistema Integral para la Atención Primaria de Salud alas SIAPS

Autores: Yilianne Arais Crombet Pérez
Yurien López Hernández

Tutores: Ing. Yoelvis Osés Sosa
Ing. Ariel Pons Alonso

Ciudad de La Habana, julio del 2010
“Año 52 de la Revolución”

DATOS DE CONTACTO

Ing. Yoelvis Osés Sosa, Instructor recién graduado en el año 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor de la Facultad # 7. Ha impartido las asignaturas Historia de la Informática, Software Libre y Seguridad Informática. Se ha desempeñado como Asesor Comercial del Centro de Informática Médica (CESIM). Ha presentado trabajos en UClencia 2008 y 2010, en la XIII Convención y Feria Internacional Informática 2009, posee publicaciones en las memorias de dichos eventos. Ha sido tutor de otros trabajos de diploma.

e-mail: yoses@uci.cu

Ing. Ariel Pons Alonso, Instructor recién graduado en el año 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor de la Facultad # 7. Ha impartido las asignaturas Máquinas Computadoras I y II. Se ha desempeñado líder de desarrollo en el Departamento Atención Primaria de Salud del Centro de Informática Médica (CESIM). Ha presentado trabajos en UClencia 2008 y en la XIII Convención y Feria Internacional Informática 2009, posee publicaciones en las memorias de dichos eventos.

e-mail: apons@uci.cu

Resumen

La Atención Primaria de Salud está constituida por una variedad de instituciones encargadas de brindar atención integral de salud a la población. El personal de enfermería que labora en dichas instituciones médicas desempeña una función esencial en el cuidado, rehabilitación, promoción y prevención de pacientes, garantizándoles una mayor calidad de vida y potenciando al máximo sus capacidades de autocuidado. En el proceso de atención a los pacientes las enfermeras gestionan grandes volúmenes de información de forma manual y en papel, lo que implica que disminuya la frecuencia con que están en contacto directo con los pacientes, afectándose la calidad de los servicios de enfermería proporcionados a los mismos.

La presente investigación propone obtener un sistema informático que facilite la gestión de la información de los servicios de Enfermería en la Atención Primaria de Salud, basado en tecnologías libres, multiplataforma y de código abierto.

Se emplea para el desarrollo de la aplicación la Plataforma Java Enterprise Edition 5.0. Como lenguaje de programación se utiliza Java, además PostgreSQL como Sistema Gestor de Bases de Datos, Hibernate como herramienta ORM (Object Relational Mapping) para la persistencia de los datos y el framework integrador Seam.

Con el Módulo de Enfermería del Subsistema Web del SIAPS se espera aumentar la calidad de la atención de enfermería brindada a los pacientes, además de contribuir a mejorar los procesos relacionados con la vacunación y la esterilización, permitiendo que la información que estos generan, sea actualizada y accesible en tiempo real.

PALABRAS CLAVE: Atención Primaria de Salud, Enfermería, Esterilización, Vacunación.

TABLA DE CONTENIDO

| | |
|---|-----------|
| Introducción | 0 |
| Capítulo 1.Fundamentación Teórica | 5 |
| 1.1 Sistema Nacional de Salud..... | 5 |
| 1.2 Marco Conceptual | 6 |
| 1.3 La Informática en la Enfermería | 6 |
| 1.4 Análisis de las soluciones existentes..... | 7 |
| 1.5 Tendencias y Tecnologías..... | 9 |
| 1.6 Herramientas..... | 15 |
| Capítulo 2. Características del Sistema | 16 |
| 2.1 Modelo del Negocio..... | 16 |
| 2.2 Diagramas de procesos de negocio | 17 |
| 2.3 Flujo actual de los procesos involucrados en el campo de acción | 20 |
| 2.4 Propuesta del Sistema | 23 |
| Capítulo 3. Diseño del Sistema | 28 |
| 3.1 Modelos de diseño | 28 |
| 3.2 Patrones de diseño | 28 |
| 3.3 Definición de Elementos de Diseño | 30 |
| Capítulo 4. Implementación | 52 |
| 4.1 Propuesta de integración entre módulos | 52 |
| 4.2 Propuesta de seguridad del módulo | 52 |
| 4.3 Implementación..... | 53 |
| 4.4 Diagrama de Despliegue | 53 |
| 4.5 Estándares de codificación y tratamiento de excepciones..... | 54 |
| Conclusiones | 58 |
| Recomendaciones | 59 |
| Referencias Bibliográficas | 60 |
| Bibliografía | 63 |
| Anexos | 66 |

Introducción

Actualmente el desarrollo tecnológico muestra una convergencia cada vez mayor entre las Tecnologías de la Información y las Comunicaciones (TIC), teniendo una alta incidencia en la modernización y eficiencia de todos los sectores de la sociedad. Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social estas tecnologías; y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitará a nuestra sociedad acercarse más hacia el objetivo de un desarrollo sostenible. [1]

La salud es una de las primeras esferas que se ha visto beneficiada con el auge vertiginoso de la informática. La necesidad de mejorar los servicios de atención sanitaria y los recursos en todos los niveles de atención médica, han hecho necesaria la integración de las nuevas tecnologías a este sector.

Desde los primeros años del triunfo de la Revolución Cubana, fue una estrategia política e interés del gobierno revolucionario y el Ministerio de Salud Pública (MINSAP), el estudio y procesamiento de los hechos vitales y sanitarios dentro del Sistema Nacional de Salud (SNS), inicialmente de forma manual y después con equipos de cómputo. En años posteriores se introdujeron las primeras mini computadoras cubanas y se construyó el primer centro de cálculo en salud pública. [2]

El país comienza a llevar a cabo diversas acciones con el propósito de utilizar los recursos y la información de salud disponible, permitiendo enlazar a todo el sistema de salud para dar una respuesta más eficiente en la esfera de la información científica a los profesionales, técnicos de la salud y a la situación sanitaria del país.

En 1992 se crea la Red Telemática de la Salud, INFOMED, dentro de la estructura del Centro Nacional de Información de Ciencias Médicas, con una acertada visión de la influencia que las Tecnologías de la Información y las Comunicaciones ejercerán en la esfera de la información y el conocimiento.

Durante los últimos años un grupo de instituciones cubanas y el propio MINSAP han desarrollado sistemas encaminados a lograr la informatización de la salud. En todos los casos el objetivo ha sido proveer al SNS de información para la toma de decisiones y el mejoramiento de los procesos médicos asistenciales. [3]

INTRODUCCIÓN

Con el objetivo de garantizar la calidad en la prevención, cuidado, rehabilitación de los pacientes y especializar los servicios de salud, el SNS comprende tres niveles de atención médica organizados en: Atención Médica Primaria, Atención Médica Secundaria y Atención Médica Terciaria.

El eje fundamental y centro del proceso de informatización del sector lo constituye el paciente, quién será el principal beneficiado al garantizar las aplicaciones, la calidad, oportunidad y consistencia de la información, lo que incrementará la efectividad y eficiencia de los procesos relacionados con la salud, que en última instancia gravitarán en un incremento continuo y sostenido de la calidad en la atención médica. [4]

En todo el proceso de informatización del SNS, se hace un énfasis particular en la Atención Primaria de Salud (APS), como nivel conductor de la estrategia sanitaria.

La informatización de la APS tiene como objetivo principal la creación de sistemas informáticos que permitan gestionar toda la información referente a este nivel de atención médica, permitiendo el flujo de los antecedentes personales, familiares y de la comunidad hacia los niveles superiores de toma de decisiones.

La APS es, para las personas, las familias y la comunidad, el primer nivel de contacto con el SNS, que pone los cuidados de salud lo más cerca posible de los lugares en que vive y trabaja la población. [5]

En esta esfera las enfermeras constituyen un elemento esencial, integran los Grupos Básicos de Trabajo (GBT) y los Equipos Básicos de Salud (EBS), actúan en policlínicos, consultorios del médico de la familia, centros educacionales y laborales, desempeñándose fundamentalmente en las áreas de Consulta de Enfermería, Vacunación y Esterilización, haciendo énfasis en las dos últimas.

En todo sistema de atención de salud basado en la atención primaria, la función de las enfermeras figura en un lugar prominente, siendo protagonistas en el cuidado de pacientes de todas las edades, ya sean sanos o enfermos. Estas se encargan de que se fomenten y se mantengan los vínculos entre las personas, las familias, las comunidades y el resto del sistema de atención de salud. Además ellas trabajan de manera autónoma y en colaboración para prevenir enfermedades y discapacidades, además de promover, mejorar, mantener y restaurar la salud.

Las enfermeras se convierten en el centro de la prestación de cuidados en la APS, garantizando un mejor acceso a estos, con lo que se logra una mayor eficiencia en los costos y una mejor aceptación de los servicios de enfermería por parte del paciente.

INTRODUCCIÓN

Según estadísticas del Consejo Internacional de Enfermería (CIE), entre el 60 y el 80 % de la APS prestada tradicionalmente por médicos, puede ser dispensada por enfermeras a un costo menor y resultados similares. [6]

Teniendo en cuenta lo antes descrito se identifica como **Situación Problemática** lo siguiente:

En la actualidad la Atención de Enfermería se encuentra distribuida por todas las áreas de la APS, aumentando considerablemente el flujo de información que las enfermeras manipulan. Esta se utiliza para valorar, diagnosticar, planificar, intervenir y evaluar las condiciones de los pacientes.

La recopilación de la información se realiza de forma manual y en papel, demandando una porción muy significativa de tiempo y recursos. Según estudios realizados, se demostró que por cada hora de cuidado directo sobre el paciente se generaba entre 30 y 60 minutos de registro en papel [7], aumentando la cantidad de documentación acumulada. Lo que provoca que las enfermeras empleen menos tiempo estando en contacto directo con el paciente, agudizándose la carencia que existe de estos profesionales en la actualidad.

Esta escasez de personal genera diferentes problemas, por un lado pocas enfermeras para cuidar muchos pacientes y por el otro un aumento del horario laboral de las enfermeras dispuestas u obligadas a cubrir esos puestos.

Las constantes limitaciones en la gestión de información a las que se enfrentan las enfermeras dificultan el desempeño de sus responsabilidades y pone en riesgo la salud del paciente.

Las campañas de vacunación realizadas a la población son guiadas por las enfermeras. En estos momentos se les hace difícil acceder a la información referente a la Tarjeta de Vacunación de los pacientes de forma rápida y oportuna, por lo que se torna engorroso llevar el control de las personas que deben vacunarse así como de la dosis respectiva que les corresponde, pudiéndoseles administrar la dosis o vacuna incorrecta.

Las enfermeras se encargan de controlar los materiales esterilizables para su desinfección, empaquetamiento y esterilización. Si existe algún error o duplicado de registro en este proceso y por consiguiente gran cantidad de información confusa, conllevaría a que algún área no reciba la cantidad de materiales necesarios ni con la higiene requerida para su correcto funcionamiento.

INTRODUCCIÓN

Después de analizar la situación antes expuesta, se plantea como **Problema a Resolver** la siguiente interrogante: ¿Cómo facilitar la gestión de la información relacionada con la atención de enfermería en la Atención Primaria de Salud?

Se define como **Objeto de Estudio** el proceso de gestión de la información en la Atención Primaria de Salud.

El **Campo de Acción** se centra en el proceso de gestión de la información de Enfermería en la Atención Primaria de Salud.

Para dar solución a la problemática planteada se ha definido como **Objetivo General** de la investigación, desarrollar el Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS, que facilite la gestión de la información de los servicios de Enfermería en la Atención Primaria de Salud.

Para dar cumplimiento al objetivo general se plantean las siguientes **Tareas de la Investigación:**

- ✓ Realizar la fundamentación teórica como resultado de la revisión bibliográfica e investigación del estado del arte relacionado con la atención de enfermería en la APS.
- ✓ Realizar el estado del arte de las tendencias actuales en el mundo y en la nación, relacionadas con los Sistemas de Atención de Enfermería en la APS.
- ✓ Analizar la arquitectura definida en el Centro de Informática Médica (CESIM) para aplicaciones web.
- ✓ Realizar la preparación y capacitación correspondiente para el uso de la arquitectura definida y aprobada por el CESIM para el desarrollo de las aplicaciones web.
- ✓ Elaborar la documentación y artefactos correspondientes mediante la metodología definida para los flujos de trabajo: Modelado de Negocio, Requerimientos, Análisis y Diseño e Implementación de los procesos de las áreas de Vacunación y Esterilización en la APS.
- ✓ Implementar los procesos de las áreas de Vacunación y Esterilización en la APS.

En este sentido se puede destacar que el desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS, proporcionará un grupo de **Aportes Prácticos** entre los que pueden ser mencionados los siguientes:

INTRODUCCIÓN

- ✓ Proponer modelos oficiales para el control de los procesos de recibo, esterilización y entrega de los materiales esterilizables.
- ✓ Proveer al personal de enfermería de Registros Informatizados que gestionen todo el proceso de vacunación a la población.
- ✓ Actualizar de forma automática la Tarjeta de Vacunación de los pacientes.
- ✓ Consultar y actualizar en tiempo real el Esquema Nacional de Vacunación.

El presente documento está estructurado en cuatro capítulos, que incluyen todo lo relacionado con el trabajo investigativo, así como el diseño e implementación del sistema.

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

En el **Capítulo II** Características del Sistema: Se refleja la investigación realizada con los procesos que tienen lugar en el negocio como objeto de estudio. Se describe la propuesta del sistema, se aborda lo referente al funcionamiento del negocio, sus reglas, descripción y las mejoras que propone el mismo. Se describe además la solución propuesta, utilizando los requerimientos funcionales y no funcionales, los procesos de negocio, el diagrama de procesos de negocio y prototipos de interfaz de usuario.

En el **Capítulo III** Diseño del Sistema: Se realiza el diseño del sistema a desarrollar, con el propósito de refinar y estructurar los requisitos obtenidos con anterioridad para facilitar la comprensión, preparación, modificación y mantenimiento de los mismos. Describe los aspectos relacionados al diseño de la solución propuesta, se modelan los diagramas de clases del diseño.

En el **Capítulo IV** Implementación: Trata los aspectos relacionados con la construcción de la solución propuesta, se modela el diagrama de despliegue, aborda la descripción de los estándares de diseño, codificación, seguridad y además del tratamiento de errores en la solución del sistema.

Capítulo 1. Fundamentación Teórica

Este capítulo está dedicado a analizar conceptos que se deben conocer para tener un dominio básico de la situación planteada. Además de realizar un análisis detallado del estado del arte en cuanto a sistemas existentes, tecnologías, metodologías y herramientas utilizadas para el desarrollo del sistema propuesto.

1.1 Sistema Nacional de Salud

La forma y los métodos que sirven de base para la organización de la atención a la salud en un país determinado, es lo que se conoce como Sistema Nacional de Salud (SNS). La Organización Mundial de la Salud (OMS) lo define como: “Un sistema que engloba todas las organizaciones, instituciones y recursos cuyo principal objetivo es llevar a cabo actividades encaminadas a mejorar la salud”. [8]

De acuerdo con la complejidad de las acciones preventivas, curativas y de rehabilitación, así como la especialización de los servicios de salud brindados, los diferentes niveles de atención médica se han organizado en:

- ✓ **Atención Primaria de Salud (APS):** Da solución aproximadamente al 80 % de los problemas de salud de la población y que correspondan con las acciones de promoción y protección de la salud. Aunque sus actividades se realizan en cualquier unidad del SNS, están relacionados fundamentalmente con las que se realizan en clínicas Urbanas o Rurales, Dispensarios y Postas Médicas.
- ✓ **Atención Secundaria de Salud (ASS):** Este nivel da cobertura acerca del 15 % de los problemas de salud, su función fundamental es tratar a la persona ya enferma, tanto desde el punto de vista individual como colectivo, pero también desempeña funciones de rehabilitación, promoción y prevención de la salud. Se llevan a cabo acciones de salud más complejas y especializadas. Comprende la atención médica brindada en los distintos Hospitales.
- ✓ **Atención Terciaria de Salud (ATS):** 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.

1.2 Marco Conceptual

1.2.1 Atención de Enfermería

La atención de enfermería es considerada una valiosa herramienta en el desempeño del profesional de enfermería, que permite prestar cuidados de una forma racional, lógica y sistemática para obtener información e identificar los problemas del individuo, la familia y la comunidad con el fin de planear, ejecutar y evaluar el cuidado de enfermería. [9]

Enfermería

Es la ciencia o disciplina que se encarga del estudio de las respuestas reales o potenciales de la persona, familia o comunidad tanto sana como enferma en los aspectos biológicos, psicológicos, sociales y espirituales.

Esterilización

Es llevada a cabo por las enfermeras, este proceso consiste en destruir todo tipo de microbios, bacterias y hongos, que se encuentran en los instrumentos y materiales de cura.

Vacunación

Es el proceso que realizan las enfermeras para prevenir las enfermedades a la población, por medio de las vacunas.

1.3 La Informática en la Enfermería

Las computadoras constituyen la herramienta tecnológica más poderosa que ha transformado más velozmente la profesión de enfermería en el nuevo siglo. Hoy y mañana, la informática y la Internet serán esenciales para todos los escenarios donde se desempeñe la actividad de enfermería.

La informática en enfermería integra las ciencias de la enfermería, las ciencias de la computación y las ciencias de la información para manejar y comunicar datos. Integra información y conocimiento en la práctica de enfermería ayudando a los pacientes, enfermeras y médicos en la toma de decisiones. Esta ayuda es realizada a través del uso de estructuras, procesos y tecnologías de información constituyendo un nuevo paradigma presentado a las enfermeras como resultado de los impactos producidos por los avances de la tecnología computacional.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el año 2002 se publicó un estudio cualitativo realizado sobre 26 entrevistas a personal de salud tratando de indagar, qué herramientas tecnológicas podrían ayudar a resolver los problemas en la gestión de información relacionados con la enfermería. Partiendo del proceso de atención de enfermería como resultado de las entrevistas se determinaron 10 puntos donde la informática podría ayudar a mejorar la seguridad del paciente y la satisfacción de las enfermeras con el trabajo diario. [10]

Las enfermeras colectan y utilizan mucha información en el cuidado de pacientes, por lo que los sistemas de información pueden reducir el tiempo de documentación en un 30%, proveer acceso a información, mejorar la adherencia a estándares así como asegurar que toda la información es almacenada y recolectada una sola vez. Esto permitiría que las enfermeras dispongan de más tiempo para el cuidado de los pacientes, mejorando la calidad de la información del paciente.

1.4 Análisis de las soluciones existentes

En la actualidad existen diversos sistemas informáticos que abarcan el espectro de la APS. Debido a la variedad que existe entre los sistemas de salud a escala mundial, cada uno de ellos está enfocado a las necesidades particulares de la institución para la cual se hayan implementado.

Existe una solución española llamada **LogicPAE** que cuenta con un generador de inteligencia artificial incorporado que permite gestionar completamente el proceso de atención de enfermería de las distintas unidades o servicios que disponga un centro de salud.

LogicPAE es un sistema inteligente de escritorio, monoplataforma que soporta solamente el sistema operativo Windows sobre la versión XP o superiores. Es un software propietario construido sobre plataforma .Net. Existen actualmente tres versiones del producto (Estudiantes, Profesional y Enterprise) cuyas descargas pueden realizarse de manera gratuita, pero están limitadas las funcionalidades más importantes.

Para la adquisición de una de las versiones del software con el paquete completo de funcionalidades es necesario el pago de varias tarifas (en euros) por concepto de licencias. Además el costo de formación de los usuarios para interactuar con el sistema está dado por otras tarifas bastante elevadas que varían según la cantidad de usuarios, el tiempo de capacitación y el país donde se realice la misma.

En el ámbito nacional han existido desde décadas anteriores intentos de fomentar el desarrollo de la informática en el campo de la enfermería. El 16 de junio del 2000 con la creación del Grupo Nacional de

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Informática de Enfermería (GINENF) comenzaron a darse los primeros pasos en la informatización del sector en el país. La primera aplicación creada por esta organización fue una base de datos utilizando el procesador de Excel, que era enviada por correo electrónico, para la recogida estandarizada de los datos, lo que facilitó el procesamiento de la información y al mismo tiempo permanecía el registro por cada año.

Más tarde se oficializó el software de Proceso de Atención de Enfermería (PAE) elaborado por el CIREN (Centro Internacional de Restauración Neurológica), el cual se valida en todos los servicios de la propia institución por el personal de enfermería que lo utiliza. En el hospital Hermanos Ameijeiras, también existía otra versión del PAE, pero este era utilizado sólo en algunos servicios del centro.

Utilizando el lenguaje de programación Delphi y sustentados en SQL Server como gestor de base de datos se comenzaron a confeccionar dos software, uno de Esterilización y otro de Gerencia, cuyas implementaciones no se llegaron a materializar.

En Cuba los sistemas existentes no cubren las necesidades del personal de enfermería debido a que la información que gestionan no está acorde al flujo actual de datos que se procesan en los departamentos de enfermería en la APS. Además, los mismos fueron desarrollados para una institución en específico por lo que no pueden extenderse hacia todos los servicios de enfermería de la APS.

En la esfera internacional el sistema mencionado muestra funcionalidades de relevancia en el campo de la enfermería, pero presenta como desventaja que es una aplicación de escritorio lo que eleva considerablemente los costos y dificulta el proceso de despliegue al tener que instalarlos en cada puesto de trabajo. Además, no es un sistema multiplataforma por lo que restringe al usuario final a utilizar un sistema operativo específico.

También se debe destacar la falta de compatibilidad e intercambio de información clínica con otros sistemas, lo que impide que todo el cúmulo de información que maneja tribute a otros niveles de atención médica y que pueda integrarse para brindar un mejor servicio de salud.

Los sistemas anteriormente expuestos no cumplen con los requisitos de integración plasmados en la estrategia de informatización de la salud en Cuba, lo cual imposibilita la uniformidad en cuanto al uso del mismo sistema, y el acceso a ellos por el personal de enfermería de la Atención Primaria para desarrollar sus actividades de la misma forma.

1.5 Tendencias y Tecnologías

Durante el desarrollo de este epígrafe se tratarán una serie de conceptos muy utilizados en el campo de la informática, los mismos se encuentran relacionados con la arquitectura, tecnologías, herramientas y metodologías que con su interconexión permiten la obtención del Módulo de Enfermería del SIAPS.

1.5.1 Sistemas Distribuidos. Modelo Cliente-Servidor

Un sistema distribuido se define como una colección de ordenadores autónomos conectados por una red, y con el software distribuido adecuado para que el sistema sea visto por los usuarios como una única entidad capaz de proporcionar facilidades de computación. Se podría ver un sistema distribuido como un grupo de ordenadores independientes que son percibidos por los usuarios como un único ordenador. [11]

Una de las arquitecturas más difundidas en el desarrollo de los sistemas distribuidos, es la arquitectura cliente-servidor. La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

1.5.2 Patrones de arquitectura

De forma general un patrón es un modelo a seguir para darle solución a un determinado problema, estos surgen de la experiencia de los seres humanos al tratar de lograr ciertos objetivos, capturando la experiencia existente y probada para promover las buenas prácticas. Tienen como objetivo la creación de un lenguaje común para la comunidad de desarrolladores permitiéndole su uso miles de veces sin hacerlo siquiera en dos ocasiones de la misma forma. Existen varios tipos de patrones, dependiendo del contexto particular en la cual sean aplicados o de la etapa en el proceso de desarrollo, algunos de estos tipos son: de Diseño, de Arquitectura, de Negocios, de Análisis, para ambientes distribuidos, etc.

El 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 y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista. [12]

Este modelo de arquitectura presenta varias ventajas:

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- ✓ Es posible tener diferentes vistas para un mismo modelo.
- ✓ Es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.
- ✓ Proporciona un mecanismo de configuración a componentes complejos mucho más tratables que el puramente basado en eventos.

Si se diseña un sistema con este patrón, se puede construir por separado las piezas del programa y luego unirlos en tiempo de ejecución. Así mismo cuando alguna de las piezas se vea afectada, conllevando a que el funcionamiento de un componente no sea el más óptimo, esta puede cambiarse por otra.

1.5.3 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 de desarrollo como el uso de patrones.

Java Server Faces (JSF)

La tecnología Java Server Faces es un marco de desarrollo de componentes de interfaz de usuario del lado del servidor y es válido para todas aquellas aplicaciones web basadas en la tecnología JAVA. JSF traduce las distintas acciones del usuario en eventos que son respondidos por el servidor regenerando la página original, reflejando los cambios necesarios para la acción realizada. El objetivo del desarrollo de aplicaciones mediante JSF, es construir aplicaciones web que se parezcan a las aplicaciones de escritorio.

RichFaces

RichFaces es un framework de código abierto que añade capacidad JavaScript asíncrono y XML (Ajax) dentro de aplicaciones JSF existentes sin recurrir a JavaScript. RichFaces incluye ciclo de vida, validaciones, conversores y la gestión de recursos estáticos y dinámicos. Los componentes de RichFaces están contruidos con soporte Ajax y un alto grado de personalización que puede ser fácilmente incorporado dentro de las aplicaciones JSF. [13]

Permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes altamente configurables en cuanto a temas y esquemas de colores predefinidos por el propio framework o desarrollados a conveniencia, lo que mejora grandemente la experiencia de usuario.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Ajax4JSF

Ajax4jsf es una librería open source 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.

Ajax4jsf presenta mejoras sobre los propios beneficios del framework JSF incluyendo el ciclo de vida, validaciones, facilidades de conversión y el manejo de recursos estáticos y dinámicos. Permite definir un evento en una página que invoca una petición Ajax y luego las áreas de la página deberían sincronizarse con el árbol de componentes JSF después de que la petición Ajax cambie los datos en el servidor.

Facelets

Facelets es un framework simplificado de presentación, 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 crear plantillas para construir un árbol de componentes de forma que puedan definirse como composición de otros.

Jboss Seam

JBoss Seam es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB). Con seam basta agregar anotaciones propias de éste a los objetos Entidad y Session de EJB, logrando escribir menos código Java y XML.

Otra característica importante es que se pueden hacer validaciones en los POJOs (Plain Old Java Objects) como además manejar directamente la lógica de la aplicación y de negocios desde los session beans.

Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional 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) que permiten establecer estas relaciones. Genera sentencias SQL (Structure Query Language) a partir de sentencias HQL (Hibernate Query Language). A través de la implementación del estándar JPA (Java Persistence API) que provee Hibernate 3.3, se puede realizar el acceso a datos.

[14]

1.5.4 Lenguaje de Programación

Se utiliza Java como lenguaje de programación orientado a objetos, el cual fue desarrollado por Sun Microsystems a principio de los años noventa. Java hace uso de la sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir muchos errores, como la manipulación directa de punteros o memoria.

Tiene como ventaja que es un lenguaje multiplataforma, que se ha extendido cobrando cada día más importancia tanto en el ámbito de Internet como en la informática en general.

Java permite programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema.

1.5.5 Servidor de Aplicaciones

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0, y esto hace que el desarrollo de las aplicaciones sean mucho más simples. [15]

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

1.5.6 Sistema Gestor de Base de Datos

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Berkeley Software Distribution) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. [16]

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. [17]

1.5.7 Metodología de Desarrollo de Software

Capability Maturity Model Integration (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Las mejores prácticas de 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 alcanzar estas áreas o estadios se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables.

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

- ✓ Nivel 1: Inicial.
- ✓ Nivel 2: Administrado.
- ✓ Nivel 3: Definido.
- ✓ Nivel 4: Cuantitativamente administrado.
- ✓ Nivel 4: Optimizado.

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

La Universidad de las Ciencias Informáticas se encuentra desarrollando un proceso de mejora con el objetivo de obtener el Nivel 2 de CMMI. Las áreas de procesos que lo forman son:

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

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Dentro de las áreas de procesos mencionadas anteriormente la Universidad se encuentra enfocada en la Administración de Requisitos.

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 es utilizado el documento IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos realizado por la universidad y 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 (Estudio Preliminar, Modelación del Negocio, Requisitos, Análisis y Diseño, Pruebas Internas, Pruebas de Liberación, Despliegue y Soporte). En este se establece por cada fase la relación con los subprocesos descritos en el libro de procesos específico para el área de Administración de Requisitos.

La descripción del IPP-3510:2009 Libro de Proceso para la Administración de Requisitos 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.

Proceso Unificado de Desarrollo de Software (RUP)

El Proceso Unificado de Desarrollo de Software (RUP), es una metodología de desarrollo de software orientada a objetos 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, 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 Libro de Proceso para la Administración

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

de Requisitos se utiliza UML, con la especificación del uso del estándar BPMN para la descripción de los procesos de negocio.

Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

Notación Utilizada para Modelar los Procesos del Negocio

La Notación para el Modelado de Procesos de Negocio (BPMN) 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.

1.6 Herramientas

Toda metodología de desarrollo de sistemas debe sustentarse sobre la base de una herramienta CASE que garantice un mejor análisis del sistema y la calidad del resultado final, a su vez estas herramientas proporcionan el potencial para mejorar la productividad del analista. La herramienta seleccionada para el sustento de la metodología fue Visual Paradigm suite 6.4, esta Herramienta CASE da soporte al modelado visual de UML 2.1 y BPMN.

Para el desarrollo de la aplicación web, se tiene al Entorno de Desarrollo Integrado (IDE) Eclipse SDK en su versión 3.4, con la colección de plug-ins de JBoos Tools, necesarios para utilizar los frameworks de desarrollo que se proponen. Además, se utiliza el pgAdmin III para la administración de la base de datos.

En este capítulo se expusieron los requisitos y premisas definidos para la informatización del sector de la salud en el país. Se profundizó en el conocimiento de algunos conceptos necesarios así como en los antecedentes históricos de la atención de enfermería en la APS en los intentos de informatizarla, tanto en el ámbito nacional como internacional. Se realizó un análisis de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto.

Capítulo 2. Características del Sistema

En este capítulo se aborda todo lo referente al modelado del negocio y el sistema del Módulo Enfermería que se desarrollará, mostrando claramente los aspectos fundamentales que darán paso a la descripción de la solución que se propone. Se detallan ampliamente todos los procesos involucrados, haciendo un profundo y extenso análisis de las actividades que serán objeto de automatización. Además, se especifican los requerimientos funcionales y no funcionales.

2.1 Modelo del Negocio

Para el desarrollo de cualquier sistema un paso fundamental es el análisis y la comprensión de los procesos existentes en el negocio. Un proceso de negocio no es más que un conjunto completo o colección de actividades relacionadas lógicamente, que toman uno o más tipos de entradas y crean uno o más resultados que producen un valor para la organización, sus inversores o sus clientes. [18]

El modelado del negocio es una técnica para comprender los procesos del negocio de la organización. Los propósitos que se persiguen al realizarse este modelo son: entender la estructura y la dinámica de la organización, entender los problemas actuales e identificar mejoras potenciales, asegurarse de que los clientes, usuarios finales y desarrolladores tengan una idea común de la organización y derivar los requerimientos del sistema a partir del modelo de negocio que se obtenga.

Para realizar el modelado del negocio se utiliza la notación BPMN. La misma aporta una mayor visibilidad de las actividades que se realizan y ayuda a lograr un mejor entendimiento del flujo de trabajo existente entre las áreas, lo que permite definir con claridad las actividades innecesarias a la hora de automatizar el negocio.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.2 Diagramas de procesos de negocio

2.2.1 Vacunación

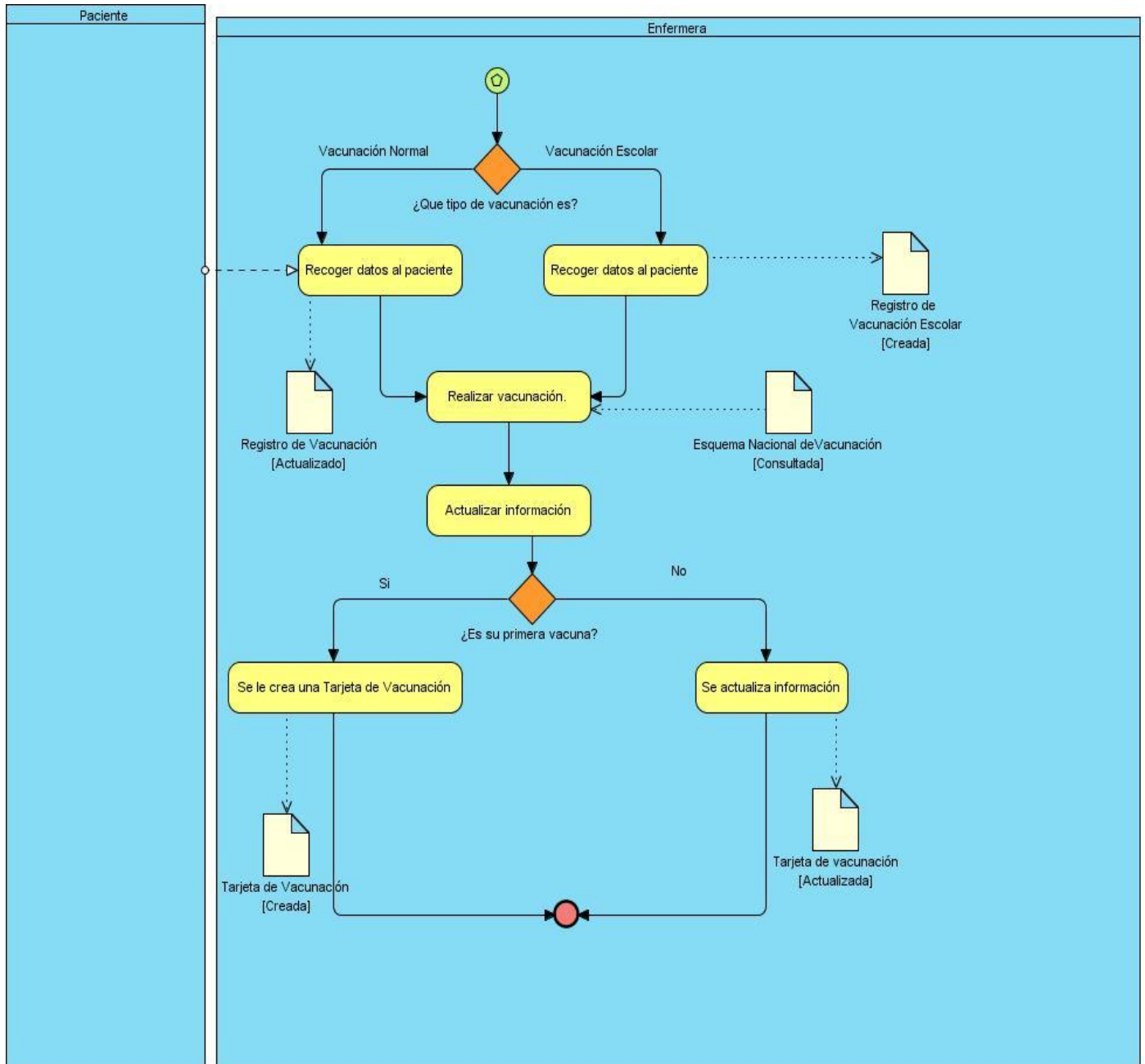


Figura 1. Diagrama de Procesos del Negocio de Vacunación. Realizar Vacunación.

2.2.2 Esterilización

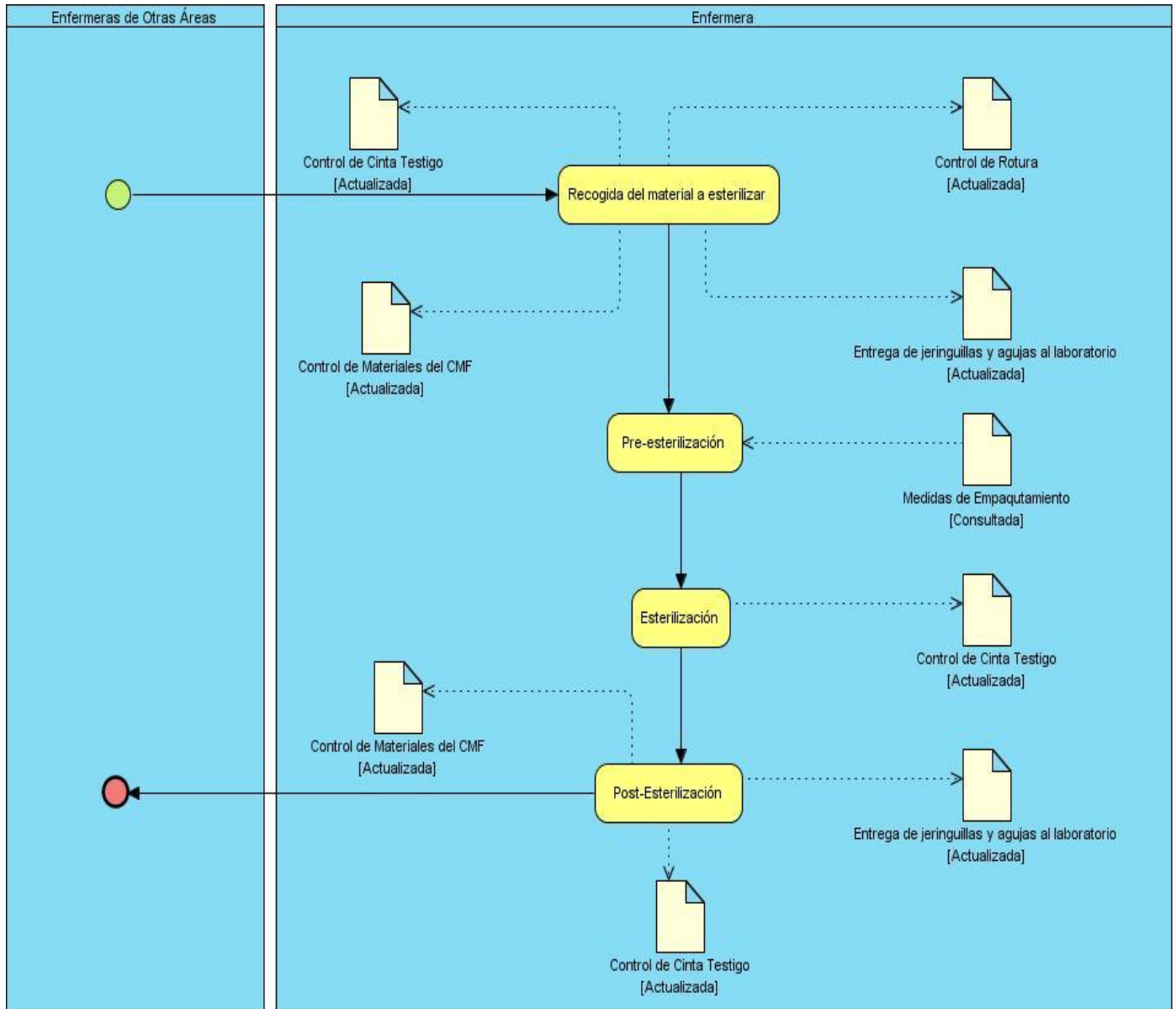


Figura 2. Diagrama de Procesos del Negocio de Esterilización. Esterilizar Material.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

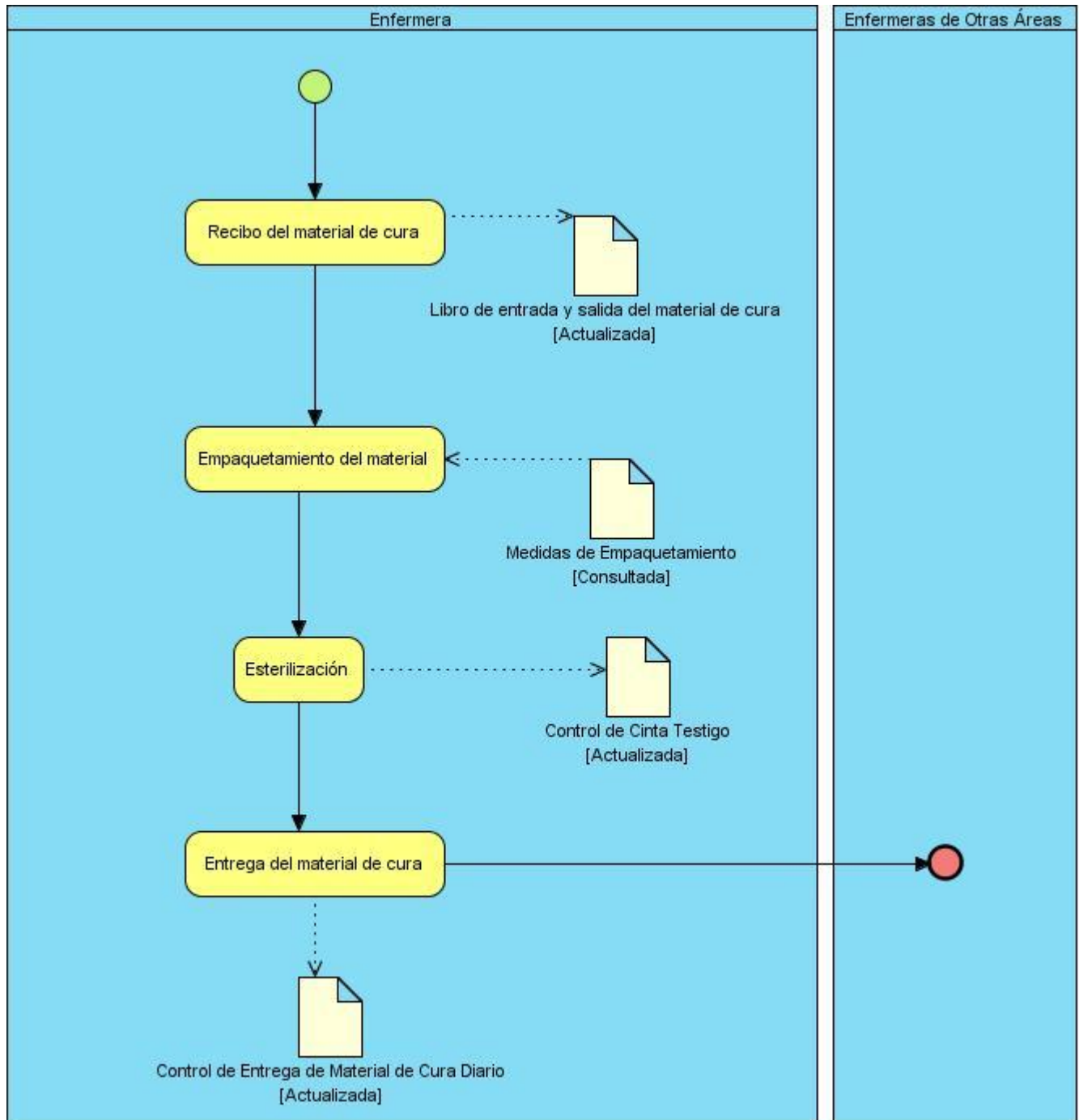


Figura 3. Diagrama de Procesos del Negocio de Esterilización. Empaquetar Materiales de Cura.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.3 Flujo actual de los procesos involucrados en el campo de acción

Dentro de los procesos que se contemplan en la Atención de Enfermería en la APS existen dos que tienen un gran peso dado su vital importancia: Vacunación y Esterilización.

La Vacunación es la forma de prevención de ciertas enfermedades infecciosas que consiste en provocar en un sujeto la aparición de anticuerpos específicos contra estas enfermedades, por administración de sustancias antigénicas denominadas vacunas. Son inyecciones de antitoxinas como medio de protección contra bacterias causantes de enfermedades.

Tiene como objetivo fundamental evitar que la población padezca cualquiera de las diferentes patologías que se pueden prevenir con este proceso y cuyo propósito no es otro que el de mantener a la comunidad saludable.

La Esterilización tiene como objetivo lograr una desinfección o esterilización previa de los instrumentos para evitar el contagio de una persona a otra, de cualquier tipo de enfermedad a través de algún instrumento médico, además del empaquetamiento y preparación del material utilizado en curas. Este proceso tiene como propósito cuidar y proteger la salud de la población y proporcionar al material de curas un empaquetamiento adecuado para garantizar que el mismo no esté propenso a infecciones.

2.3.1 Procesos de Realizar Vacunación

| | |
|---------------------------|--|
| Nombre: | P1_Realizar vacunación |
| Objetivos: | El objetivo de este proceso es prevenir que el paciente padezca de algunas de las diferentes patologías. |
| Evento(s) que lo generan: | <ul style="list-style-type: none">• Que llegue algún paciente que le toque vacunarse con alguna de las vacunas según el Esquema de Vacunación de la República de Cuba.• Que se esté desarrollando alguna campaña de vacunación para prevenir alguna enfermedad infecto-contagiosa |
| Precondiciones: | <ul style="list-style-type: none">• Que halla algún paciente para vacunar. |
| Poscondiciones: | <ul style="list-style-type: none">• Que se actualice la Tarjeta de Vacunación con la vacuna administrada.• Que se actualice el Registro de Vacunación o el Registro de Vacunación Escolar. |
| Reglas de Negocio: | <ul style="list-style-type: none">• Debe haber un paciente al cual vacunar. |

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

| | |
|--------------------|---|
| | <ul style="list-style-type: none">• Debe existir la vacuna que se va a poner.• Se debe tener conocimiento acerca de las vacunas y las dosis correspondientes.• La tarjeta de vacunación personal debe ser actualizada después de cada vacuna. |
| Responsables: | Enfermera |
| Clientes internos: | Paciente |
| Clientes externos: | CESIM |
| Entradas: | <ul style="list-style-type: none">• Esquema Nacional de Vacunación |
| Salidas: | <ul style="list-style-type: none">• Registro de Vacunación.• Registro de Vacunación Escolar.• Tarjeta de Vacunación. |
| Actividades: | <ul style="list-style-type: none">• Recoger datos al paciente• Realizar vacunación.• Actualizar información. |

2.3.1.1 Realizar Vacunación. Descripción Textual.

La Vacunación comienza con el proceso de **Realizar Vacunación** que tiene las siguientes actividades:

Al **Recoger los datos al paciente** la enfermera del vacunatorio o departamento de vacunación llenará el Registro de Vacunación (Modelo 18-30-01) Ver Anexo 1, según se administren las vacunas a los pacientes. Todos los pacientes poseen una Tarjeta de Vacunación. (Modelo 84-20-02) Ver Anexo 2. La vacunación se realiza en departamentos específicos existentes en los policlínicos (Vacunatorio ó departamento de vacunación) a donde acuden los pacientes, una vez que han sido citados por su médico y/o enfermera de la familia. Todos los pacientes cada 10 años deben pasar por su Consultorio del Médico de la Familia (CMF) para vacunarse contra las diferentes enfermedades.

En el caso de los pacientes desde 0 a 12 meses de nacidos además de seguir estos pasos también se le habilita una tarjeta (Carné del Recién Nacido) con los datos generales, la información del nacimiento, donde aparecen además el esquema de ablactación (Término utilizado para describir el inicio de alimentos diferentes de la leche) del lactante, tablas sobre el desarrollo normal pondoestatural y el esquema

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

nacional de inmunización vigente. En este carné se irá reflejando la información correspondiente a cada vacunación del paciente, también se reflejará cualquier condición que modifique el proceso de vacunación, como es el caso de madres antígeno positivo.

El Carné del Recién Nacido acompañará al recién nacido durante todo su período de vacunación y será el documento que mostrará la madre en el vacunatorio o lugar donde se realice la vacunación, con el objetivo de que cada vez que se vacune el niño la información de este carné sea actualizada. Cuando este llega al área de salud debe ser dispensarizado antes de las 72 horas. En este momento se le confecciona la Historia Clínica Individual y se le crea la tarjeta de vacunación.

La vacunación en escuelas, también es llevada a cabo por las enfermeras, para lo cual también es llenado el Registro de Vacunación (Modelo 84-19-01) Ver Anexo 1.

Luego se procede a **Realizar la Vacunación**, donde se selecciona la vacuna según el paciente, se verifica que no esté en tiempo de vencimiento, se verifican los implementos para vacunar, se limpia con yodo o alcohol el área a vacunar y se procede con la inyección, una vez vacunado el paciente se le explica a su acompañante todas las reacciones o efectos adversos que podría ocasionar la vacuna y en caso de presentar algún síntoma de reacción a ésta, acudir de inmediato al médico o cuerpo de guardia.

En el caso de los infantes se les ponen las vacunas que son obligatorias por el esquema nacional de vacunación.

La vacunación escolar puede ser por campaña o porque en ciertos grados, a los escolares les corresponden vacunas según lo planificado en el Esquema Nacional de Vacunación, Ver Anexo 3.

A las embarazadas se vacunan en caso de que tengan más de 10 años sin vacunarse con alguna vacuna específica que le corresponda por el Esquema Nacional de Vacunación, aunque entre las 22 y las 24 semanas se les pone la vacuna del Tétano, esta es obligatoria en esta fecha, sí como se decía ya hubieran pasado los 10 años sin ponérsela.

Existen vacunas que se administran a grupos específicos de población de riesgo, que no están contempladas en el Esquema Nacional de Inmunización (vacuna contra la leptospirosis, vacuna antigripal, vacuna antirrábica)

Una vez vacunado el paciente se procede a **Actualizar la información**, la enfermera del vacunatorio reflejará en el Carné del Recién Nacido la fecha, dosis, lote y vacuna administrada, para que

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

posteriormente la enfermera de la familia actualice esta información en la Tarjeta de Vacunación del paciente y el médico actualice esta información en la Historia Clínica Individual.

A medida de que se van vacunando los escolares se llena el Registro de Vacunación Escolar (Modelo 84-19-01) Ver Anexo 4, y se actualiza la Tarjeta de vacunación del paciente, notificando a los consultorios correspondientes que sus escolares fueron vacunados.

Para el resto de los pacientes solo se procede a la actualización de la Tarjeta de vacunación y de la Historia Clínica Individual (HCI).

Para obtener información de los procesos del negocio de las áreas de Vacunación y Esterilización, remitirse al Expediente de Proyecto. [19]

2.4 Propuesta del Sistema

2.4.1 Requerimientos

La Metodología RUP propone como uno de sus flujos de trabajo el de Gestión de Requerimientos, el cual tiene su mayor peso en la fase de inicio, teniéndose como artefactos fundamentales de este flujo, la especificación de requisitos, actores y casos de usos del sistema.

2.4.2 Especificación de los Requerimientos de Software

Los requerimientos de software son condiciones o capacidades que tiene que alcanzar o poseer un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Estos posibilitan determinar la solución tecnológica a algún problema existente, logrando como resultado, un informe detallado de los requerimientos necesarios para la solución a dicho problema, que permita alcanzar los objetivos y solucionar los problemas existentes. [20]

Los requisitos se pueden clasificar en: funcionales y no funcionales, donde la primera clasificación corresponde a las especificaciones del sistema, mientras que los últimos expresan las cualidades que el producto debe tener.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.4.3 Requerimientos Funcionales

Los requerimientos funcionales especifican capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. [21]

Después de analizados los procesos del negocio, se han definido los siguientes requisitos funcionales:

RF 1 Gestionar el Codificador Vacuna.

- RF 1.1 Buscar Vacuna.
- RF 1.2 Agregar Vacuna.
- RF 1.3 Modificar Vacuna.
- RF 1.4 Eliminar Vacuna.
- RF 1.5 Detalles Vacuna

RF 2 Gestionar el Codificador Dosis.

- RF 2.1 Buscar Dosis.
- RF 2.2 Agregar Dosis.
- RF 2.3 Modificar Dosis.
- RF 2.4 Eliminar Dosis.

RF 3 Gestionar el Codificador Región Anatómica.

- RF 3.1 Buscar Región Anatómica.
- RF 3.2 Agregar Región Anatómica.
- RF 3.3 Modificar Región Anatómica.
- RF 3.4 Eliminar Región Anatómica.

RF 4 Gestionar el Codificador Inmunización.

- RF 4.1 Buscar Inmunización.
- RF 4.2 Agregar Inmunización.
- RF 4.3 Modificar Inmunización.
- RF 4.4 Eliminar Inmunización.

RF 8 Gestionar Registro de Vacunación Escolar.

- RF 8.1 Registrar de Vacunación Escolar.
- RF 8.2 Registro de Vacunación Escolar.
- RF 8.3 Modificar Registro de Vacunación Escolar.
- RF 8.4 Mostrar Registro de Vacunación Escolar
- RF 8.5 Eliminar Efectos adversos asociados a un paciente.

RF 9 Mostrar Tarjeta de Vacunación.

RF 10 Mostrar Esquema Nacional de Vacunación.

RF 11 Gestionar el Codificador Material.

- RF 11.1 Buscar Material.
- RF 11.2 Agregar Material.
- RF 11.3 Modificar Material.
- RF 11.4 Eliminar Material.

RF 12 Gestionar el Codificador Medidas de Empaquetamiento.

- RF 12.1 Buscar Medidas de Empaquetamiento.
- RF 12.2 Agregar Medidas de Empaquetamiento.
- RF 12.3 Modificar Medidas de Empaquetamiento.
- RF 12.4 Eliminar Medidas de Empaquetamiento.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RF 5 Gestionar el Codificador Grado de Escolaridad.

- RF 5.1 Buscar Grado de Escolaridad.
- RF 5.2 Agregar Grado de Escolaridad.
- RF 5.3 Modificar Grado de Escolaridad.
- RF 5.4 Eliminar Grado de Escolaridad.

RF 6 Gestionar el Codificador Efectos Adversos.

- RF 6.1 Buscar Efectos Adversos.
- RF 6.2 Agregar Efectos Adversos.
- RF 6.3 Modificar Efectos Adversos.
- RF 6.4 Eliminar Efectos Adversos.

RF 7 Gestionar Registro de Vacunación.

- RF 7.1 Registrar de Vacunación.
- RF 7.2 Registro de Vacunación.
- RF 7.3 Modificar Registro de Vacunación.
- RF 7.4 Mostrar Registro de Vacunación.
- RF 7.5 Eliminar Efectos adversos asociados a un paciente.

RF 13 Gestionar Registro de Materiales Entregados.

RF 13.1 Entregar Materiales.

RF 13.2 Modificar Registro Materiales Entregados

RF 13.3 Registro de Materiales Entregados

RF 13.4 Eliminar Materiales Entregados

RF 14 Gestionar Registro de Materiales Recibidos.

RF 14.1 Recibir Materiales.

RF 14.2 Modificar Registro Materiales Recibidos

RF 14.3 Registro de Materiales Recibidos

RF 14.4 Eliminar Materiales Recibidos

RF 15 Gestionar Materiales a Esterilizar.

RF 15.1 Buscar Materiales a Esterilizar

RF 15.2 Esterilizar Materiales.

RF 16 Gestionar Registro de Materiales Esterilizados.

RF 16.1 Modificar Registro Materiales Esterilizados.

RF 16.2 Registro de Materiales Esterilizados

RF 17 Buscar pacientes.

2.4.3.1 Requerimientos No Funcionales

Los requerimientos no funcionales expresan las propiedades o cualidades que el producto debe tener, las características que hacen al producto atractivo, usable, rápido o confiable. Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Seguridad de acceso y administración de usuarios

Se mantendrá la seguridad y control entre usuarios, garantizando su acceso sólo a los niveles establecidos de acuerdo con la función que realizan. Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema.

Se mantendrá un segundo nivel de seguridad entre estaciones de trabajo, garantizando únicamente la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión. Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.

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

Restricciones de diseño

La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio. La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario. La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

Interfaz

Interfaces de usuario

Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con teclas de función y menús desplegables que faciliten y aceleren su utilización. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español.

Interfaces software

Se interactuará con el sistema alas HIS para realizar solicitudes y obtener resultados relacionados con la hospitalización del Paciente.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Requerimientos de hardware

Estaciones de trabajo

En la solución se incluyen estaciones de trabajo para las consultas del Sistema para la Atención Primaria alas SIAPS, las que necesitan una capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web, se recomienda IE 7, Firefox 2 o versiones superiores. Por lo que se escogieron estaciones de trabajo de 256 MB de memoria RAM y un microprocesador de 2.0 Hz con sistema operativo Linux.

Servidores

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

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). El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2 o versiones superiores de estos.

En este capítulo, se realizó la descripción y modelación de los procesos del negocio asociados al campo de acción, obteniendo una perspectiva del sistema que se desea desarrollar partiendo de los requerimientos funcionales y no funcionales, los cuales definen las capacidades y cualidades que la aplicación debe tener. Con la culminación de este capítulo se sentaron las bases para el exitoso desarrollo de las disciplinas de Diseño e Implementación del Módulo de Enfermería.

Capítulo 3. Diseño del Sistema

El Diseño de un sistema de software da una mayor comprensión de los aspectos relacionados con los requerimientos no funcionales y contribuye a la definición de una arquitectura estable y sólida, creando un plano del modelo de implementación.

En este capítulo se realiza el diseño de la solución propuesta con anterioridad, se describe la concepción arquitectónica del Sistema Integral para la Atención Primaria de Salud alas SIAPS y su aplicación en el Módulo de Enfermería, con el objetivo de entender la estructura y dinámica de la organización del mismo.

Se especifica el modelo de diseño del sistema mediante diagramas de clases poniendo de manifiesto el flujo de procesos. Seguidamente se describen cada una de las clases y diagramas de clases, correspondientes al proceso involucrado, justificándose además el uso de los diferentes patrones.

3.1 Modelos de diseño

Los Modelos de Diseño son una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas con la implementación.

Estos pueden contener: 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.

3.2 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 un modelo que se puede seguir para la realización de una actividad determinada, es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características.

CAPÍTULO 3. DISEÑO DEL SISTEMA

El modelo de datos se encarga del almacenamiento de los datos, su estructura y las relaciones entre los mismos. Para la administración de los datos se utiliza el framework Hibernate. El cual tiene como ventajas el control de la accesibilidad a la información almacenada, así como la realización de rápidas consultas. Hibernate implementa una serie de patrones para la administración del modelo de datos, entre ellos están los siguientes:

- ✓ **Active record:** Permite la asociación de filas únicas de la base de datos con objetos del lenguaje de programación usado.
- ✓ **Identity map:** Mantiene referencias a los objetos cargados y devuelve las mismas cada vez que se solicita un objeto ya cargado. De esta forma, actúa como un caché dentro de la transacción, lo cual también contribuye a reducir el número de accesos a la base de datos.
- ✓ **Identity field:** Permite mapear las relaciones entre referencias entre objetos y claves entre tablas.
- ✓ **Foreign key mapping:** Permite mapear las relaciones de asociación de clave foránea.
- ✓ **Association Table mapping:** Permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, lo cual crea una base de datos orientada a objetos virtuales, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).
- ✓ **Lazy load:** Carga los distintos componentes de una clase a medida que son solicitados.
- ✓ **Query object:** 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.

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que 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:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.

CAPÍTULO 3. DISEÑO DEL SISTEMA

- ✓ **Creador:** Este patrón como su nombre lo indica es el que crea, el que 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.
- ✓ **Alta cohesión:** La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.
- ✓ **Bajo acoplamiento:** El acoplamiento de una clase es el conjunto de dependencias que tiene con otras clases. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la menor repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.
- ✓ **Controlador:** Asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

En la realización del presente modelo de diseño y con el objetivo de mejorar la calidad de los diagramas correspondientes a esta disciplina fueron aplicados los patrones mencionados anteriormente.

3.3 Definición de Elementos de Diseño

La representación de las clases del diseño está asociada al uso de UML para el modelado de Aplicaciones WEB, siendo identificadas las siguientes clases: Página Servidor (Server Page), Página Cliente (Client Page) Formulario (Form), para la representación de las clases contenedoras del código de las páginas servidoras, clientes y los formularios respectivamente.

Para modelar los diagramas de paquetes y los de clases del diseño se utilizará la siguiente nomenclatura:

Diagrama de paquetes: Diagrama de paquetes_<Nombre del módulo>, Diagramas de clases del diseño: DCD_<Nombre de la funcionalidad>.

CAPÍTULO 3. DISEÑO DEL SISTEMA

La clase contenedora del framework JSF se encarga de generar el código de las páginas clientes (<<construye>>), los formularios componen las páginas clientes (permite que entre las páginas clientes y sus formularios exista la relación de composición), a través de los cuales se muestran e introducen los datos necesarios, los cuales 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>>).

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

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

3.3.1 Diagramas de Clases del Diseño

Gestionar Registro de Materiales Esterilizados

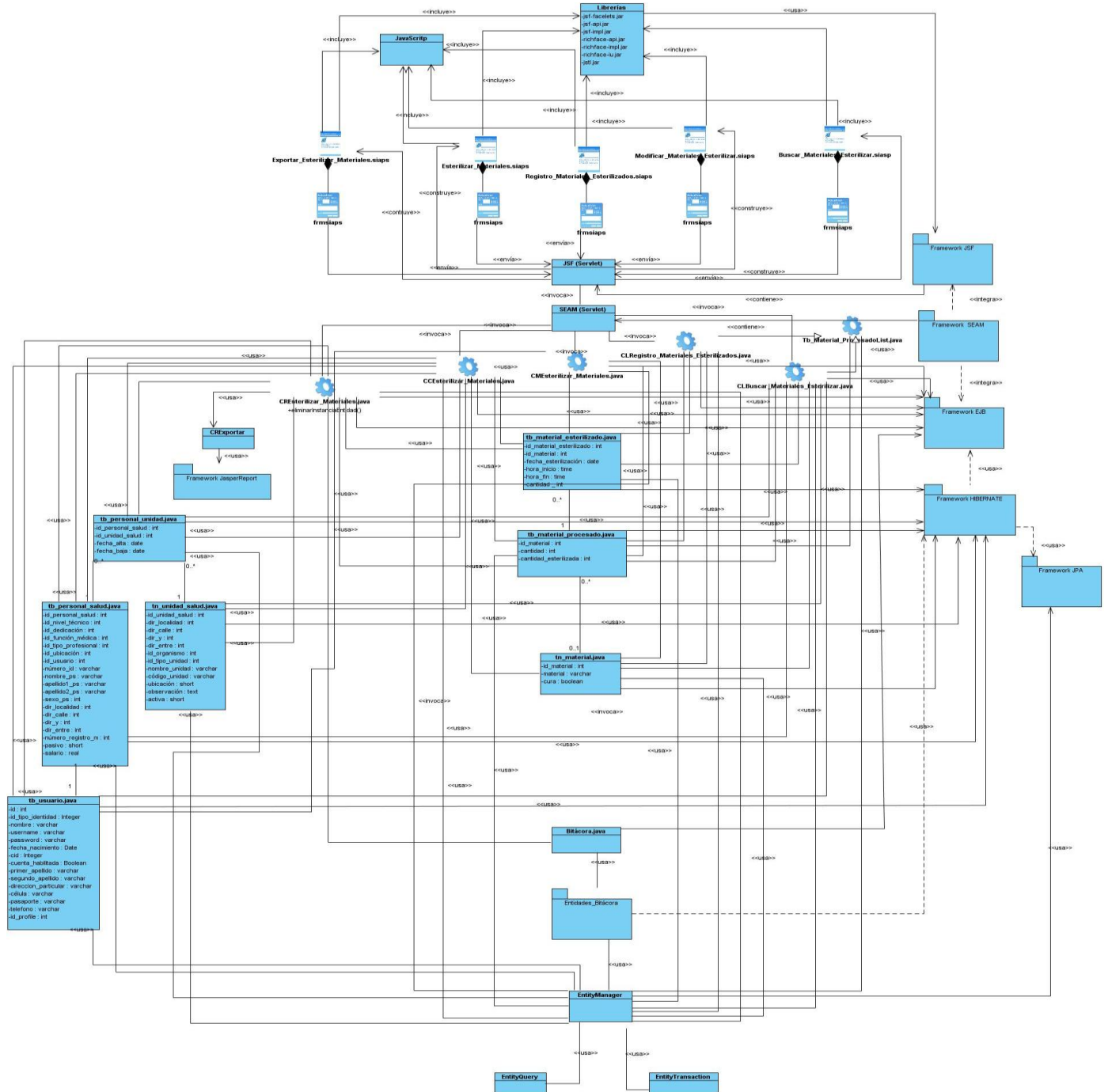





Figura 4. Diagrama de clases del diseño. Gestionar Registro de Materiales Esterilizados.

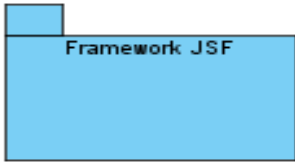
3.3.2 Descripciones textuales

3.3.2.1 Clases comunes. Descripción

| Capa de Presentación | |
|--|--|
| Nombre: | Propósito: |
|  <p>Figura 6. Clase JavaScript</p> | <p>Proveer el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.</p> |
| Descripción: | |
| <p>Permite realizar las validaciones del lado del cliente y construir páginas más dinámicas integradas a un navegador web</p> | |
| Nombre: | Propósito: |
|  <p>Figura 7. Clase Librerías</p> | <p>Permite la creación de interfaces.</p> |
| Descripción: | |
| <p>Contiene un conjunto de librerías que permiten construir una interfaz de usuario con componentes reutilizables y extensibles. Reducen significativamente la carga de construir y mantener aplicaciones web con componentes de interfaz del lado del servidor.</p> | |
| Nombre: | Propósito: |
|  <p>Figura 8. Clase JSF Servlet</p> | <p>Interceptar las peticiones a las páginas JSF.</p> |
| Descripción: | |
| <p>Es el controlador de JSF que intercepta las peticiones de las páginas clientes, asociándoles a estas páginas, clases java que recogen la información introducida y que disponen de métodos que responden a</p> | |


CAPÍTULO 3. DISEÑO DEL SISTEMA

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

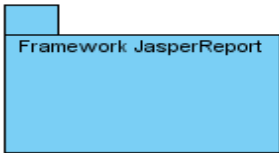
| Nombre: | Propósito: |
|--|--|
|  <p>Figura 9. Paquete Framework JSF</p> | Simplificar el desarrollo de las interfaces de usuario en aplicaciones java basadas en el patrón Modelo-Vista-Controlador. |

Descripción:
La tecnología Java Server Faces es un framework de los componentes de la interfaz de usuario y es válido para todas aquellas aplicaciones web basadas en la tecnología JAVA, está basado en el patrón MVC. Forma parte de la especificación JEE 5.

Capa de Negocio

| Nombre: | Propósito: |
|--|--|
|  <p>Figura 10. Clase CRExportar</p> | Proveer las funcionalidades para exportar a diferentes formatos. |

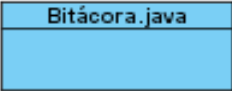
Descripción:
Permite realizar las funciones para exportar a diferentes tipos de archivos mediante el uso de las librerías del Framework JasperReports.

| Nombre: | Propósito: |
|---|---|
|  <p>Figura 11. Paquete Framework JasperReports</p> | Añadir características de generación de reportes a aplicaciones java. |

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

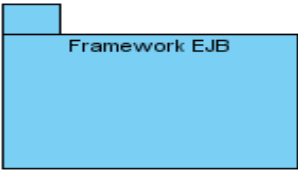
CAPÍTULO 3. DISEÑO DEL SISTEMA

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

| | |
|---|---|
| Nombre: | Propósito: |
|  <p>Figura 12. Clase Bitácora.java</p> | Proveer las funcionalidades para realizar las auditorias del sistema. |

Descripción:


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

| | |
|--|--|
| Nombre: | Propósito: |
|  <p>Figura 13. Paquete Framework EJB</p> | Encapsular la lógica de negocio que cumplimenta el propósito de la aplicación. |
| Observaciones: | |
| El Framework EJB está incluido en las capas de Negocio y Persistencia. | |

Descripción:

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



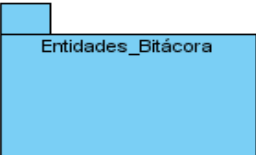
Capa de Datos

| | |
|---|---|
| Nombre: | Propósito: |
|  <p>Figura 14. Clase EntityManager</p> | Gestionar las entidades proveyendo servicios de persistencia. |

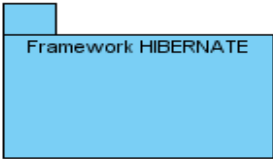
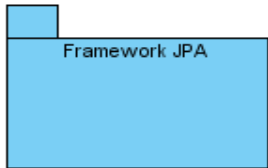
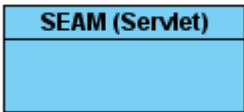
Descripción:

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

CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|---|---|
| Nombre: | Propósito: |
|  <p>Figura 15. Clase EntityManager</p> | Agregar consultas que pueden aplicarse a las entidades del modelo. |
| Descripción: | |
| Permite encontrar objetos persistentes manejando cierto criterio de búsqueda. Permite realizar peticiones a la base de datos y controla cómo se ejecuta dicha petición. Se utiliza para enlazar los parámetros de la petición, limitar el número de resultados devueltos por la petición y para ejecutar dicha petición. | |
| Nombre: | Propósito: |
|  <p>Figura 16. Clase EntityManager</p> | Agrupar las operaciones sobre datos persistentes en una unidad transaccional. |
| Descripción: | |
| Permite realizar operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos. | |
| Nombre: | Propósito: |
|  <p>Figura 17. Clase Entidades Bitácora</p> | Agrupar las entidades que contienen la información de las auditorías del sistema. |
| Descripción: | |
| Contiene el conjunto de entidades que poseen la información de los usuarios y sus trazas en cuanto a sesión utilizada, módulos accedidos, funcionalidades permitidas, acciones realizadas y atributos modificados. Son utilizadas por la clase Bitácora.java para realizar las auditorías del sistema. | |
| Nombre: | Propósito: |



CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|--|--|
|  <p>Figura 18. Paquete Framework HIBERNATE</p> | <p>Proveer el mapeo objeto/relacional con la base de datos.</p> |
| <p>Descripción:</p> | |
| <p>Conjunto de clases agrupadas en componentes que constituyen una herramienta de Mapeo objeto/relacional ó ORM de código abierto (Object Relational Mapping) y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada permite generar Bases de Datos en cualquiera de los entornos soportados: Oracle, PostgreSQL, DB2, MySQL, entre otras.</p> | |
| <p>Nombre:</p> | <p>Propósito:</p> |
|  <p>Figura 19. Paquete Framework JPA</p> | <p>Unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.</p> |
| <p>Descripción:</p> | |
| <p>Conjunto de clase agrupadas en componentes que constituyen la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. Permite unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.</p> | |
| <p>Nombre:</p> | <p>Propósito:</p> |
|  <p>Figura 20. Clase SEAM Servlet</p> | <p>Proveer la interacción de la capa de presentación con la de negocio.</p> <p>Observaciones:</p> <p>No se encuentra en ninguna capa en específico, sino que se encuentra ubicado entre la capa de Presentación y la de</p> |





CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|--|----------|
| | Negocio. |
| Descripción: | |
| Es el controlador de SEAM que capta las peticiones derivadas de la interacción del usuario después de interactuar con el Servlet de JSF. Enrutar las peticiones hacia los Beans que posibilitarán darle respuesta a la petición solicitada. Interviene en la integración de las capas de presentación y negocio. | |




3.3.2.2 Gestionar Registro de Vacunación. Descripción

| Capa de Presentación | |
|--|--|
| Nombre: | Propósito: |
|  <p>Figura 21. Clase RegistrarVacunación.siaps</p> | Proveer la interacción con el usuario. |
| Descripción: | |
| La clase RegistrarVacunación.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá insertar todos los datos necesarios para proceder con la vacunación del paciente y que todo quede correctamente registrado. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF. | |
| Nombre: | Propósito: |
|  <p>Figura 22. frmsiaps</p> | Enviar los datos a las páginas servidoras. |
| Descripción: | |
| La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para insertar todos los datos necesarios para proceder con la vacunación y que todo quede correctamente registrado. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST. | |
| Nombre: | Propósito: |

CAPÍTULO 3. DISEÑO DEL SISTEMA


| | |
|---|---|
|  <p style="text-align: center;">Figura 23. Clase Listado.shaps</p> | <p>Proveer la interacción con el usuario.</p> |
| <p>Descripción:</p> | |
| <p>La clase Listado.shaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá buscar cualquier registro hecho en alguna fecha o por alguna enfermera en específico. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p> | |
| <p>Nombre:</p> | <p>Propósito:</p> |
|  <p style="text-align: center;">Figura 24. frmsiaps</p> | <p>Enviar los datos a las páginas servidoras.</p> |
| <p>Descripción:</p> | |
| <p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para listar todos los datos necesarios de los Registros de Vacunación de cada paciente. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p> | |
| <p>Nombre:</p> | <p>Propósito:</p> |
|  <p style="text-align: center;">Figura 25. Clase ModificarRegistroVacunación.shaps</p> | <p>Proveer la interacción con el usuario.</p> |
| <p>Descripción:</p> | |
| <p>La clase ModificarRegistroVacunación.shaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá acceder a modificar cualquier registro hecho con anterioridad. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p> | |
| <p>Nombre:</p> | <p>Propósito:</p> |
|  <p style="text-align: center;">Figura 26. frmsiaps</p> | <p>Enviar los datos a las páginas servidoras.</p> |

CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|--|---|
| Descripción: | |
| <p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para modificar todos los datos necesarios de cualquier registro hecho con anterioridad. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p> | |
| Nombre: | Propósito: |
|  <p style="text-align: center;">Figura 27. Clase Detalles.siaps</p> | <p>Proveer la interacción con el usuario.</p> |
| Descripción: | |
| <p>La clase Detalles.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá desde la cual podemos ver los detalles de cualquier registro escogido previamente, en la cual se mostrará una serie de datos que no se muestran en el registro. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p> | |
| Nombre: | Propósito: |
|  <p style="text-align: center;">Figura 28. 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 para mostrar todos los datos necesarios de un Registro de Vacunación. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p> | |
| Nombre: | Propósito: |
|  <p style="text-align: center;">Figura 29. Clase Exportar.siaps</p> | <p>Proveer la interacción con el usuario.</p> |
| Descripción: | |
| <p>La clase Exportar.siaps es una página web que se ejecuta del lado del cliente sobre un navegador.</p> | |

CAPÍTULO 3. DISEÑO DEL SISTEMA


Permitirá exportar los datos del Registro de Vacunación, es decir el listado de pacientes atendidos en el vacunatorio, da la opción de escoger la cantidad de páginas que deseas exportar y además te brinda la opción de exportar tanto en formato WORD, como en PDF. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.

| | |
|--|--|
| Nombre: | Propósito: |
|  <p>Figura 30. frmsiaps</p> | Enviar los datos a las páginas servidoras. |

Descripción:

La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para exportar todos los datos del Registros de Vacunación. Sus atributos son los elementos de entrada del formulario (input boxes, text areas, check boxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.

Capa de Negocio




| | |
|---|--|
| Nombre: | Propósito: |
|  <p>CRegistrarVacunación.java</p> <p>Figura 31. Clase CRegistrarVacunación.java</p> | Proveer una respuesta a las peticiones realizadas en la vista. |

Descripción:

La clase CRegistrarVacunación.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de insertar los datos sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.


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

CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|---|--|
|  TbRegistroVacunaList.java Figura 32. Clase TbRegistroVacunaList.java | Proveer una respuesta a las peticiones realizadas en la vista. |
| Descripción: | |
| <p>La clase TbRegistroVacunaList.java es una clase que se genera de forma automática el realizar la ingeniería inversa y que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de listar los datos solicitados sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.</p> | |
| Nombre: | Propósito: |
|  CLBuscarRegistroVacunación.java Figura 33. CLBuscarRegistroVacuna.java | Proveer una respuesta a las peticiones realizadas en la vista. |
| Descripción: | |
| <p>La clase CLBuscarRegistroVacuna.java es una clase que se ejecuta del lado del servidor. Hereda todas las funcionalidades de la clase autogenerada TbRegistroVacunaList.java y permite guardar los cambios originales de la misma evitando que los mismos se pierdan al realizar la ingeniería inversa. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.</p> | |
| Nombre: | Propósito: |
|  CRegistroVacunación.java Figura 34. Clase CRegistroVacunación.java | Proveer una respuesta a las peticiones realizadas en la vista. |
| Descripción: | |


CAPÍTULO 3. DISEÑO DEL SISTEMA

La clase CRegistroVacunación.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de modificar los datos de cualquier registro hecho con anterioridad que se solicite sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

| Nombre: | Propósito: |
|--|--|
|  CDRegistroVacunación.java Figura 35. Clase CDRegistroVacunación.java | Proveer una respuesta a las peticiones realizadas en la vista. |

Descripción:



La clase CDRegistroVacunación.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de mostrar los datos de la Tarjeta de Vacunación de cualquier paciente escogido con anterioridad que es solicitado desde la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

| Nombre: | Propósito: |
|--|--|
|  CRRegistroVacunación.java Figura 36. Clase CRRegistroVacunación.java | Proveer una respuesta a las peticiones realizadas en la vista. |

Descripción:

La clase CRRegistroVacunación.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de exportar solicitada sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|--|--|
| Nombre: | Propósito: |
|  TbPacienteList.java Figura 37. Clase TbPacienteList.java | Proveer una respuesta a las peticiones realizadas en la vista. |
| Descripción: | |
| La clase TbPacienteList.java es una clase que se genera de forma automática el realizar la ingeniería inversa y que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de listar los datos solicitados sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM. | |
| Nombre: | Propósito: |
|  CLPaciente.java Figura 38. Clase CLPaciente.java | Proveer una respuesta a las peticiones realizadas en la vista. |
| Descripción: | |
| La clase CLPaciente.java es una clase que se ejecuta del lado del servidor. Hereda todas las funcionalidades de la clase autogenerada TbPacienteList.java y permite guardar los cambios originales de la misma evitando que los mismos se pierdan al realizar la ingeniería inversa. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM. | |
| Capa de Datos | |
| Nombre: | Propósito: |
| <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p>TbModeloVacunación.java</p> <p>-idRegVacunación : int -fechaVacunación : date -vacunador : int -idUnidad : int</p> </div> Figura 39. Clase TbModeloVacunación.java | Proveer el mapeo con la base de datos. |

CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|--|---|
| Descripción: | |
| <p>La clase TbModeloVacunación.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los modelos de vacunación diarios de cada enfermera. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p> | |
| Nombre: | Propósito: |
| <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;">EnfermeralnEntidad_Configuracion.java</p> <pre style="margin: 0;">-id : int -id_enfermero : int -version : int -eliminado : boolean -cid : int -id_entidad : int</pre> </div> <p style="text-align: center;">Figura 40. Clase EnfermeralnEntidad_Configuración.java</p> | <p>Proveer el mapeo con la base de datos.</p> |
| Descripción: | |
| <p>La clase EnfermeralnEntidad_Configuración.java es una clase que se ejecuta del lado del servidor. En ella se almacenan la relación entre las unidades a las que pertenece un personal de salud determinado. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p> | |
| Nombre: | Propósito: |
| <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;">Enfermera_Configuracion.java</p> <pre style="margin: 0;">-id : int -matriculaColegioEnfermeria : varchar -version : int -eliminado : boolean -cid : int</pre> </div> <p style="text-align: center;">Figura 41. Clase Enfermera_Configuración.java</p> | <p>Proveer el mapeo con la base de datos.</p> |
| Descripción: | |
| <p>La clase Enfermera_Configuración.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de del personal de salud de la Atención Primaria de salud (APS). Representa una</p> | |

CAPÍTULO 3. DISEÑO DEL SISTEMA

tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.

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

| Entidad_Configuracion.java |
|---|
| -id : int -version : int -nombre : varchar -direccion : varchar -telefono : varchar -correo : varchar -fechaApertura : date -idTipoHospital : int -idEstado : int -idMunicipio : int -idLocalidad : int -camasArquitectonicas : int -camasPresupuestadas : int -esPublico : boolean -eliminado : boolean -cid : int -logo : varchar -perteneceARhio : boolean -activa : boolean |

Proveer el mapeo con la base de datos.

Figura 42. Clase
Entidad_Configuración.java

| Descripción: | |
|--------------|--|
|--------------|--|

La clase Entidad_Configuración.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de cada una de las unidades de salud. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.

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

CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|--|---|
| <div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 10px;"> Usuario_Configuracion.java </div> <pre style="font-family: monospace; font-size: 0.9em; margin: 0;"> -id : int -idTipoidentidad : Integer -nombre : varchar -username : varchar -password : varchar -fechaNacimiento : Date -cid : Integer -cuentaHabilitada : Boolean -primerApellido : varchar -segundoApellido : varchar -direccionParticular : varchar -célula : varchar -pasaporte : varchar -telefono : varchar -idProfile : int -version : int -eliminado : boolean </pre> <p style="text-align: center; margin-top: 10px;">Figura 43. Clase Usuario_Configuración.java</p> | <p>Proveer el mapeo con la base de datos.</p> |
| Descripción: | |
| <p>La clase Usuario_Configuración.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos referentes a los usuarios con acceso al sistema. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p> | |
| Nombre: | Propósito: |
| <div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 10px;"> TbRegistroVacunas.java </div> <pre style="font-family: monospace; font-size: 0.9em; margin: 0;"> -idRegistroVacunas : int -idPaciente : int -idDosis : int -idRegVacunación : int -idVacuna : int -embarazada : boolean -campaña : boolean -observación : varchar </pre> <p style="text-align: center; margin-top: 10px;">Figura 44. Clase TbRegistroVacunas.java</p> | <p>Proveer el mapeo con la base de datos.</p> |
| Descripción: | |
| <p>La clase TbRegistroVacunas.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los identificadores de los pacientes vacunados, la vacuna y dosis administradas y las observaciones. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad</p> | |

CAPÍTULO 3. DISEÑO DEL SISTEMA

corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.

Nombre:

Propósito:

```
TbPaciente.java  
-idPaciente : int  
-idHsf : int  
-idNivelEducativo : int  
-idZonaAps : int  
-idOcupación : int  
-idFactorRh : int  
-idGrupoDispensarial : int  
-idColorPiel : int  
-idGrupo : int  
-idSexo : int  
-idUnidad : int  
-fechaNacimiento : date  
-idProfesión : int  
-nroHci : int  
-fechaCaptación : date  
-ci : varchar  
-nombrePaciente : varchar  
-primerApellido : varchar  
-segundoApellido : varchar  
-jefeNúcleo : int
```

Proveer el mapeo con la base de datos.

Figura 45. Clase TbPaciente.java

Descripción:

La clase TbPaciente.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de los pacientes. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.

Nombre:

Propósito:

```
TrVacunaDosis.java  
-idVacuna : int  
-volumenDosis : float  
-idDosis : int  
-idVacuna_dosis : int  
-edadPersona : int
```

Proveer el mapeo con la base de datos.

Figura 46. Clase TrVacunaDosis.java

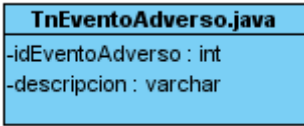
Descripción:

La clase TrVacunaDosis.java es una clase que se ejecuta del lado del servidor. En ella se almacenan la relación entre las vacunas y sus dosis correspondientes. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y

CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|---|---|
| JPA. | |
| Nombre: | Propósito: |
| <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p style="text-align: center; margin: 0;">TnDosis.java</p> <p style="margin: 0;">-idDosis : int</p> <p style="margin: 0;">-nombreDosis : int</p> </div> <p style="text-align: center; margin-top: 10px;">Figura 47. Clase TnDosis.java</p> | <p>Proveer el mapeo con la base de datos.</p> |
| Descripción: | |
| <p>La clase tn_dosis.java es una clase que se ejecuta del lado del servidor. En ella se las descripciones o nombres referentes a las dosis de una vacuna. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p> | |
| Nombre: | Propósito: |
| <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p style="text-align: center; margin: 0;">TnVacuna.java</p> <p style="margin: 0;">-idVacuna : int</p> <p style="margin: 0;">-nombreVacuna : varchar</p> <p style="margin: 0;">-descripcion : int</p> <p style="margin: 0;">-periodoReactivacion : int</p> <p style="margin: 0;">-idVia : int</p> <p style="margin: 0;">-idRegion : int</p> </div> <p style="text-align: center; margin-top: 10px;">Figura 48. Clase TnVacuna.java</p> | <p>Proveer el mapeo con la base de datos.</p> |
| Descripción: | |
| <p>La clase TnVacuna.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres referentes a las vacunas y los identificadores de sus dosis correspondientes. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p> | |
| Nombre: | Propósito: |
| <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p style="text-align: center; margin: 0;">TrRegistroEvento.java</p> <p style="margin: 0;">-idRegistroVacunas : int</p> <p style="margin: 0;">-idEventoAdverso : int</p> <p style="margin: 0;">-idRegistroEvento : int</p> </div> <p style="text-align: center; margin-top: 10px;">Figura 49. Clase TrRegistroEvento.java</p> | <p>Proveer el mapeo con la base de datos.</p> |

CAPÍTULO 3. DISEÑO DEL SISTEMA

| | |
|---|--|
| Descripción: | |
| <p>La clase TrRegistroEvento.java es una clase que se ejecuta del lado del servidor. En ella se almacenan la relación entre los registros de vacunación de cada paciente vacunado y el evento adverso ocurrido. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p> | |
| Nombre: | Propósito: |
|  <p>Figura 50. Clase TnEventoAdverso.java</p> | Proveer el mapeo con la base de datos. |
| Descripción: | |
| <p>La clase TnEventoAdverso.java es una clase que se ejecuta del lado del servidor. En ella se almacenan la relación entre los registros de vacunación de cada paciente vacunado y el evento adverso ocurrido. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p> | |

Para obtener más información de las clases del diseño de las áreas de Vacunación y Esterilización, remitirse al Expediente de Proyecto. [22]

En el capítulo se describió la propuesta de solución al problema planteado. Se especificaron y justificaron los patrones de diseño utilizados. Fue detallada la estructura del diseño, la cual complementa los requerimientos funcionales y no funcionales. Todo esto constituyó un paso fundamental para el desarrollo del flujo de Trabajo de Implementación.

Capítulo 4. Implementación

Este capítulo constituye la continuidad del Modelo de Diseño. Fundamenta la necesidad de integración del Módulo de Enfermería Web con otros sistemas. Se presenta el Diagrama de Despliegue de la solución propuesta y se definen también elementos clave para la seguridad, el tratamiento de errores y los estilos de código a utilizar en la aplicación.

4.1 Propuesta de integración entre módulos

En la actualidad casi ninguna aplicación para la salud se concibe como un sistema aislado, y el Módulo Enfermería del Subsistema Web no es la excepción. Se contempla su integración con el Sistema Integral para la Atención Primaria de Salud alas SIAPS como plataforma única para la gestión, procesamiento y transmisión de la información clínica en el SNS. La comunicación se realizará a nivel de bases de datos.

4.1.1 Módulo Medicina Familiar del Subsistema Web

El Módulo Enfermería del Subsistema Web del SIAPS se relaciona con el Módulo de Medicina Familiar del cual toma los datos de los pacientes registrados a la hora de gestionar la Vacunación de los mismos.

4.1.2 Módulo Configuración del Subsistema Web

El Módulo Configuración del Subsistema Web le permite al Módulo de Enfermería obtener la información relacionada con los nomencladores Vía de Administración, Departamentos, Clasificador Internacional de Enfermedades (CIE) y Centros Laborales, para la gestión de la Vacunación y la Esterilización, además de proporcionar los datos del Personal de Salud y las Entidades a las que pertenece.

4.2 Propuesta de seguridad del módulo

Debido a la importancia de los procesos que se llevan a cabo en los departamentos de enfermería, surge la necesidad de que estos sean realizados con la calidad requerida y que el acceso a la información de los pacientes atendidos solo sea posible para las personas que se encuentren acreditadas, todo esto trae consigo que se tengan en cuenta una serie de requisitos de seguridad de modo que no se pueda afectar el funcionamiento del sistema.

Para lograr un sistema seguro se llevará a cabo un control entre los usuarios y sus contraseñas, permitiendo el acceso por tipo de usuario logrando así la visibilidad sólo a las áreas establecidas de

acorde a la función que realizan. Las contraseñas solo podrán ser cambiadas por el usuario o por el administrador del sistema.

El sistema además permitirá llevar una traza de todas las operaciones llevadas a cabo por cada usuario mediante un registro de actividades por usuario en todo momento.

Para lograr la fidelidad de los datos, todo el intercambio entre el sistema y otros sistemas que soliciten información desde cualquier entidad, se realizará de forma cifrada eliminando posibilidades de acceso o modificación de la misma.

4.3 Implementación

Para comenzar a desarrollar el flujo de trabajo de Implementación se comienza con el resultado del diseño, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación. Además de implementar las clases y subsistemas encontrados durante el diseño. Las clases se implementan como componentes de ficheros que contienen código fuente. Uno de los principales propósitos de este flujo de trabajo consiste en desarrollar la arquitectura y el sistema en su conjunto.

4.4 Diagrama de Despliegue

Un diagrama de Despliegue muestra cómo y dónde se desplegará el sistema, describe la distribución física del sistema partiendo de cómo se distribuyen las funcionalidades entre los nodos. Los cuales se representan como máquinas físicas y procesadores, y la construcción interna puede ser representada por nodos o artefactos embebidos. Los nodos representan recursos de cómputos entre los cuales existen relaciones que representan el medio de comunicación.

De manera general el diagrama de despliegue del Módulo de Enfermería Web quedaría de la siguiente manera:

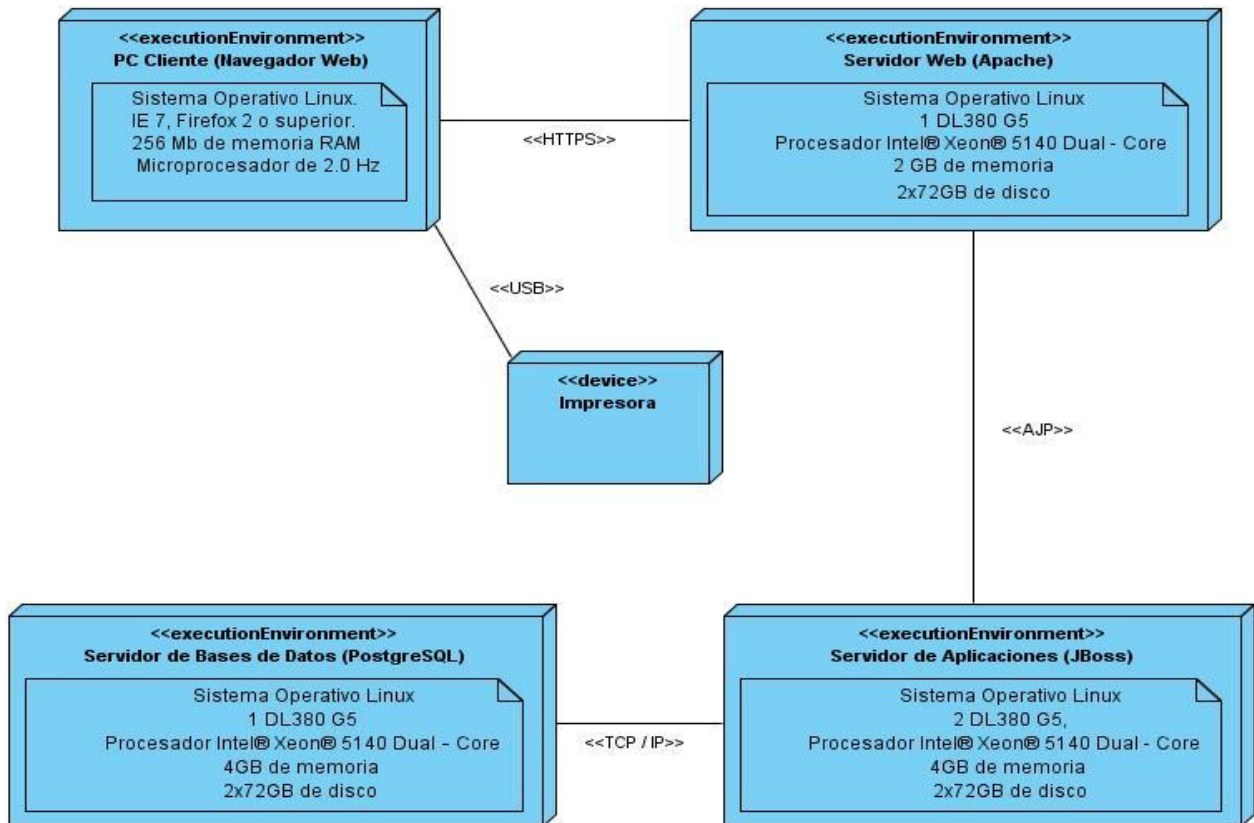


Figura 51. Diagrama de Despliegue.

Para obtener información acerca del Diagrama de Despliegue, remitirse al Expediente de Proyecto. [23]

4.5 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 a las SIAPS se han definido una serie de estándares tanto para el diseño como para la codificación, además como se le dará el tratamiento de errores al módulo para que los códigos fuentes de las aplicaciones y los mensajes que se emitan mantengan una uniformidad.

4.5.1 Estándares de codificación

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

CAPÍTULO 4. IMPLEMENTACIÓN

- ✓ El 80% del coste del código de un programa va a su mantenimiento.
- ✓ Casi ningún software lo mantiene toda su vida el autor original.
- ✓ Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo.
- ✓ Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.

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

Comentarios, líneas y espacios en blanco

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

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

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

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

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

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

Se deben usar espacios en blanco en las siguientes circunstancias:

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

Indentación y longitud de la línea

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

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

Variables, constantes, clases y métodos

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

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

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

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

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

4.5.2 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. [24]

CAPÍTULO 4. IMPLEMENTACIÓN

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. También se emplean un conjunto de tipos de excepciones predefinidas por los marcos de trabajos 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 que el envío de los activos al servidor sean los esperados.

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

En este capítulo se mostró el diagrama de despliegue, donde se ilustran los nodos que serán usados para la implantación de la aplicación y para cada uno de éstos, su protocolo de comunicación. También se presentaron los estándares de codificación y de diseño usados para lograr una aplicación legible y con uniformidad. Además, se trataron las posibles excepciones que el sistema pudiera presentar, para poder lograr la implementación de un sistema seguro y confiable.

Al concluir la implementación se obtuvo un sistema que responde a las funcionalidades previstas para la primera versión, garantizando su correcto funcionamiento.

Conclusiones

Con la elaboración del presente trabajo y en virtud de dar cumplimiento al objetivo y tareas trazadas, se arribó a las siguientes conclusiones:

- ✓ Los procesos de negocio llevados a cabo en los departamentos de enfermería en la APS demostraron la necesidad de desarrollar un sistema informático para la gestión de la información. A través de estos procesos se identificaron las funcionalidades previas a tener en cuenta para el desarrollo del sistema propuesto.
- ✓ Los sistemas informáticos analizados que automatizan los procesos que se desarrollan en los departamentos de enfermería en la APS, no responden totalmente a las funcionalidades identificadas.
- ✓ Se informatizaron los procesos de negocio de las áreas de Vacunación y Esterilización.
- ✓ Se implementó una aplicación web que facilita la gestión de la información de los servicios de Enfermería en la Atención Primaria de Salud.

Recomendaciones

Se recomienda:

- ✓ Incluir las funcionalidades para la gestión de los procesos en las áreas de Citología, Apoyo Vital, Consulta de Enfermería, Hospitalización y Puericultura.
- ✓ Definir e implementar reportes estadísticos que contribuyan a obtener información más precisa y detallada.

Referencias Bibliográficas

1. **Colectivo Autores.** cpicm HABANA. *Informática en la Salud Pública Cubana*. [En línea] 2007. [Citado el: 12 de octubre de 2009.] http://www.cpicmha.sld.cu/hab/vol13_2_07/hab17207.htm.
2. **Ramos Delgado, Ariel y Ledo Vidal, María.** Scielo Cuba. *Estrategias de Informatización del Sistema Nacional de Salud*. [En línea] 2006. [Citado el: 18 de octubre de 2009.] http://scielo.sld.cu/scielo.php?pid=S0864-34662006000300015&script=sci_arttext.
3. **Colectivo Autores.** Informática en Salud 2009. *Estrategias de Informatización del Sistema Nacional de Salud*. [En línea] 2007. [Citado el: 23 de octubre de 2009.] <http://www.informatica2009.sld.cu>.
4. Ídem a Referencia 2.
5. —. Colegio Oficial de Diplomados en Enfermería de Madrid CODEM. *Grado, Postgrado y Doctorado en Enfermería, el gran cambio de la profesión*. [En línea] 2008. [Citado el: 28 de octubre de 2009.] http://www.codem.es/Tribuna/Hemeroteca/tribuna_214.pdf.
6. **Colectivo Autores.** Consejo Internacional de Enfermería. *Servir a la comunidad y garantizar la calidad: Las enfermeras al frente de la atención primaria de salud*. [En línea] 2008. [Citado el: 28 de octubre de 2009.] <http://www.seapaonline.org/UserFiles/File/Actividades%20de%20Seapa/2008/DIE%202008.pdf>
7. —. *Curso Universitario Sistema de Información en los Sistemas de Salud (10 x 10 de la AMIA)*. Argentina : s.n., 2009.
8. —. Organización Mundial de la Salud. *Sistemas de Salud*. [En línea] 2007. [Citado el: 8 de noviembre de 2009.] http://www.who.int/topics/health_systems/es.

REFERENCIAS BIBLIOGRÁFICAS

9. **Noguera Ortiz, Norma Yaneth.** Encolombia. *Sistemas de Proceso de Atención de Enfermería. Una herramienta para la garantía del cuidado.* [En línea] 2008. [Citado el: 12 de noviembre de 2009.]
<http://www.encolombia.com/medicina/enfermeria/Enfermvol110408/Procesosdeatenciondeenfermeria.htm>.
10. —. Ídem a Referencia 7
11. **Rojo, Oscar J.** Asociación de Usuarios de GNU/Linux de Castilla y León AUGCyL. *Introducción a los Sistemas Distribuidos.* [En línea] 2003. [Citado el: 30 de noviembre de 2009.]
<http://www.augcyl.org/?q=glol-intro-sistemas-distribuidos>.
12. **Colectivo Autores.** Universidad de las Americas Puebla UDLAP. *Arquitectura del Software.* [En línea] 2007. [Citado el: 2 de diciembre de 2009.]
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf.
13. **Junta de Andalucía.** *RichFaces.* [En línea] 2008. [Citado el: 14 de diciembre de 2009.]
<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces>.
14. **Alcantar Hernández, Fernando.** GestioPolis. *Mapeo entre clases persistentes utilizando el software Hibernate My Eclipse para conectar a una base de datos con XML.* [En línea] 2008. [Citado el: 19 de diciembre de 2009.] <http://www.gestiopolis.com/administracion-estrategia/hibernate-para-bases-de-datos-con-xml.htm>.
15. **Código abierto como alternativa.** *Jboss Application Server 4.2.* [En línea] 2009. [Citado el: 6 de enero de 2010.] <http://www.osalt.com/es/jboss>.
16. **Código abierto como alternativa.** *Sobre PostgreSQL.* [En línea] 2009. [Citado el: 9 de enero de 2010.] <http://www.osalt.com/es/jboss>.

REFERENCIAS BIBLIOGRÁFICAS

17. Ídem a Referencia 16.
18. **Colectivo Autores.** Departamento de Ingeniería Informática y Ciencia de la Computación. *Análisis de Modelos de Procesos de Negocio en relación a la dimensión informática.* [En línea] 2008. [Citado el: 14 de enero de 2010.] <http://www.inf.udec.cl/~revista/ediciones/edicion9/cjimenez.pdf>.
19. **Pérez Albear, Lisandra.** Repositorio de documentos para el Departamento de Atención Primaria de Salud del Centro de Informática Médica. [En línea] 2010. <https://repositorio.cesim.prod.uci.cu/svn/aps/enfermeria/WEB/EXPEDIENTE%DE%PROYECTO>
20. **Sánchez Fornaris, Maite, Alcantara Rabí, Elvia Dayanis y Hernández Luque, Eylin.** Vinculado. *Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica.* [En línea] 2010. [Citado el: 17 de enero de 2010.] http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html.
21. **Monografías .** *Diseño de un Software.* [En línea] 2006. [Citado el: 18 de enero de 2010.] <http://www.monografias.com/trabajos42/propuesta-software/propuesta-software.shtml>.
22. Ídem a Referencia 19.
23. Ídem a Referencia 19.
24. **Cursos de Informática.** *Tratamiento de Excepciones.* [En línea] 2007. [Citado el: 22 de enero de 2010.] <http://elvex.ugr.es/decsai/builder/intro/6.html>.

Bibliografía

Alcantar Hernández, Fernando. 2008. GestioPolis. *Mapeo entre clases persistentes utilizando el software Hibernate My Eclypse para conectar a una base de datos con XML.* [En línea] 2008. [Citado el: 19 de diciembre de 2009.] <http://www.gestiopolis.com/administracion-estrategia/hibernate-para-bases-de-datos-con-xml.htm>.

2006. Biohealthmatics. *Biohealthmatics.* [En línea] 10 de agosto de 2006. [Citado el: 17 de marzo de 2010.] <http://www.biohealthmatics.com/technologies/his/nis.aspx>.

2009. Código abierto como alternativa. *Jboss Application Server 4.2.* [En línea] 2009. [Citado el: 6 de enero de 2010.] <http://www.osalt.com/es/jboss>.

2007. Cursos de Informática. *Tratamiento de Excepciones.* [En línea] 2007. [Citado el: 22 de enero de 2010.] <http://elvex.ugr.es/decsai/builder/intro/6.html>.

2008. Junta de Andalucía. *RichFaces.* [En línea] 2008. [Citado el: 14 de diciembre de 2009.] <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces>.

2006. Monografías. *Diseño de un Software.* [En línea] 2006. [Citado el: 18 de enero de 2010.] <http://www.monografias.com/trabajos42/propuesta-software/propuesta-software.shtml>.

1997. Nursing Center. *Nursing Center.* [En línea] julio/agosto de 1997. [Citado el: 18 de marzo de 2010.] http://www.nursingcenter.com/library/JournalArticle.asp?Article_ID=102101.

Colectivo Autores. 2003. ADL DATA SYSTEMS. *ADL DATA SYSTEMS.* [En línea] 2003. [Citado el: 13 de abril de 2010.] <http://www.adldata.com/Software/Clinical/index.html>.

—. **2000.** *Aprenda Java como si estuviera en primero.* Universidad de Navarra. España : s.n., 2000.

BIBLIOGRAFÍA

- . **2008.** Colegio Oficial de Diplomados en Enfermería de Madrid CODEM. *Grado, Postgrado y Doctorado en Enfermería, el gran cambio de la profesión.* [En línea] 2008. [Citado el: 28 de octubre de 2009.] http://www.codem.es/Tribuna/Hemeroteca/tribuna_214.pdf.
- . **2007.** COMPUTATA Health Corporation. *COMPUTATA Health Corporation.* [En línea] 2007. [Citado el: 21 de abril de 2010.] <http://www.compudatahealth.com/software/nursing-home-software.asp>.
- . **2007.** cpicm HABANA. *Informática en la Salud Pública Cubana.* [En línea] 2007. [Citado el: 12 de octubre de 2009.] http://www.cpicmha.sld.cu/hab/vol13_2_07/hab17207.htm.
- . **2009.** *Curso Universitario Sistema de Información en los Sistemas de Salud (10 x 10 de la AMIA).* Argentina : s.n., 2009.
- . **2008.** Departamento de Ingeniería Informática y Ciencia de la Computación. *Análisis de Modelos de Procesos de Negocio en relación a la dimensión informática.* [En línea] 2008. [Citado el: 14 de enero de 2010.] <http://www.inf.udec.cl/~revista/ediciones/edicion9/cjimenez.pdf>.
- . **2010.** Forbes. *Forbes.* [En línea] 2010. [Citado el: 21 de abril de 2010.] <http://software.forbes.com/assisted-living-software>.
- . **2000.** Infomed. *Actuación de Enfermería en APS.* [En línea] 2000. [Citado el: 25 de octubre de 2009.] http://www.sld.cu/sistema_de_salud/metodologica/enfermeria.html.
- . **2007.** Informática en Salud 2009. *Estrategias de Informatización del Sistema Nacional de Salud.* [En línea] 2007. [Citado el: 23 de octubre de 2009.] <http://www.informatica2009.sld.cu>.
- . **2007.** Organización Mundial de la Salud. *Sistemas de Salud.* [En línea] 2007. [Citado el: 8 de noviembre de 2009.] http://www.who.int/topics/health_systems/es.
- . **2006.** Portal Nacional de Enfermería. *Libros de enfermería.* [En línea] 2006. [Citado el: 24 de octubre de 2009.] <http://www.enfermeria.sld.cu/libros.html>.

BIBLIOGRAFÍA

- . 2007. Universidad de las Americas Puebla UDLAP. *Arquitectura del Software*. [En línea] 2007. [Citado el: 2 de diciembre de 2009.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf.
- Bryant, Rosemary. 2010.** Consejo Internacional de Enfermeras. *Consejo Internacional de Enfermeras*. [En línea] 12 de abril de 2010. [Citado el: 3 de mayo de 2010.] <http://www.icn.ch/index.html>
- Dowling, Alan F. 2005.** SpringerLink. *SpringerLink*. [En línea] 17 de enero de 2005. [Citado el: 16 de marzo de 2010.] <http://www.springerlink.com/content/qx8k454270831550/>.
- Martinez Évora, Yolanda Dora. 2007.** Portal de la Universidad Federal de Goiás. *La Enfermería en la Edad de la Informática*. [En línea] 2007. [Citado el: 20 de noviembre de 2009.] http://www.fen.ufg.br/revista/v9/n1/pdf/es_v9n1a01.pdf.
- Ramos Delgado, Ariel y Ledo Vidal, María. 2006.** Scielo Cuba. *Estrategias de Informatización del Sistema Nacional de Salud*. [En línea] 2006. [Citado el: 18 de octubre de 2009.] http://scielo.sld.cu/scielo.php?pid=S0864-34662006000300015&script=sci_arttext.
- Rojo, Oscar J. 2003.** Asociación de Usuarios de GNU/Linux de Castilla y León AUGCyL. *Introducción a los Sistemas Distribuidos*. [En línea] 2003. [Citado el: 30 de noviembre de 2009.] <http://www.augcyl.org/?q=glol-intro-sistemas-distribuidos>.
- Sánchez Fornaris, Maite, Alcantara Rabí, Elvia Dayanis y Hernández Luque, Eyllin. 2010.** Vinculado. *Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica*. [En línea] 2010. [Citado el: 17 de enero de 2010.] http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html.
- Suárez González, Héctor. 2003.** javaHispano. *Manual Hibernate*. [En línea] 21 de marzo de 2003. [Citado el: 10 de diciembre de 2009.] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.

Anexos.

| MODELO 18-30-01 MINISTERIO DE SALUD PUBLICA Estadísticas | | REPORTE DE VACUNACION | | | | | FECHA: | | | | |
|---|--------------------|------------------------------|-------|------|------------|--------|-----------------|-----------------|-----------------|--------|------|
| UNIDAD: | | | | | | | _____ DIA | _____ MES | _____ AÑO | | |
| Núm. Orden | No. Cons. o Sector | HISTORIA CLINICA | EDAD* | SEXO | Embarazada | VACUNA | DOSIS | | | | LOTE |
| | | | | | | | 1 ^{RA} | 2 ^{DA} | 3 ^{RA} | React. | |
| 1 | | | | | | | | | | | |
| 2 | | | | | | | | | | | |
| 3 | | | | | | | | | | | |
| 4 | | | | | | | | | | | |
| 5 | | | | | | | | | | | |
| 6 | | | | | | | | | | | |
| 7 | | | | | | | | | | | |
| 8 | | | | | | | | | | | |

* Especifique: Años cumplidos, meses o días si es menor de un año.

CONFECCIONADO POR: _____
 NOMBRE Y APELLIDOS

Anexo 1. Imagen del Registro de Vacunación (Modelo 18-30-01).

| Tarjeta de Vacunación | | | | | | |
|---|-----------------|-----------------|-----------------|--------------------|------|------------------------|
| Nombre y Apellidos: _____ | | | | | | Fecha |
| Nacimiento: _____ | | | | | | |
| Dirección Particular: _____ | | | | | | EBS # _____ GBT: _____ |
| Hijo de Madre Ag Superficie positivo (+): Si: _____ No: _____ | | | | | | |
| Vacuna | Fecha 1ra Dosis | Fecha 2da Dosis | Fecha 3ra Dosis | Fecha Reactivación | Lote | Vía de Administración |
| BCG | | | | | | |
| HVB* | | | | | | |
| HVB** | | | | | | |
| DPT + HB Tetravalente (**) | | | | | | |
| DPT | | | | | | |
| HIB | | | | | | |
| AM-BC | | | | | | |
| PRS | | | | | | |
| DT | | | | | | |
| AT | | | | | | |
| TT | | | | | | |
| Polio | | | | | | |

Anexo 2. Imagen de la Tarjeta de Vacunación (Modelo 84-20-02).

