



Universidad de las Ciencias  
Informáticas

# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

## FACULTAD 7

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

Componente web para la valoración de enfermería por patrones en el Sistema  
Integral para la Atención Primaria de Salud.

**Autores:** Zaillet Arévalo Moreno  
Darlís Darián Corría Castañeda

**Tutores:** Ing. Lisandra Pérez Albear  
Ing. Rolando Michel Fernández Pérez

**La Habana, junio de 2013**

**“Año 55 de la Revolución”**

## Datos de contacto

**Ing. Lisandra Pérez Albear:** Graduada de Ingeniera en Ciencias Informáticas en el año 2009. Actualmente labora en la Universidad de Ciencias Informáticas (UCI), presenta la categoría docente de profesor Instructor. Se desempeñó como profesora de la Facultad No. 7 donde ha impartido la asignatura de Ingeniería de Software. Vinculada a la producción en el proyecto Sistema Integral para la Atención Primaria de Salud (alasiAPS), es la analista del módulo de Enfermería. Participó y tiene artículos publicados en el evento Informática 2011 y 2013.

*Correo electrónico: [lalbear@uci.cu](mailto:lalbear@uci.cu)*

**Ing. Rolando Michel Fernández Pérez:** Graduado de Ingeniero en Ciencias Informáticas en el año 2011. Actualmente labora en la Universidad de Ciencias Informáticas (UCI), desempeñándose como especialista en el Centro de Informática Médica (CESIM). Forma parte del proyecto Sistema Integral para la Atención Primaria de Salud (alasiAPS) donde se desarrolla como Jefe del módulo Enfermería.

*Correo electrónico: [rmfernandez@uci.cu](mailto:rmfernandez@uci.cu)*

## Agradecimientos

*Muchas gracias:*

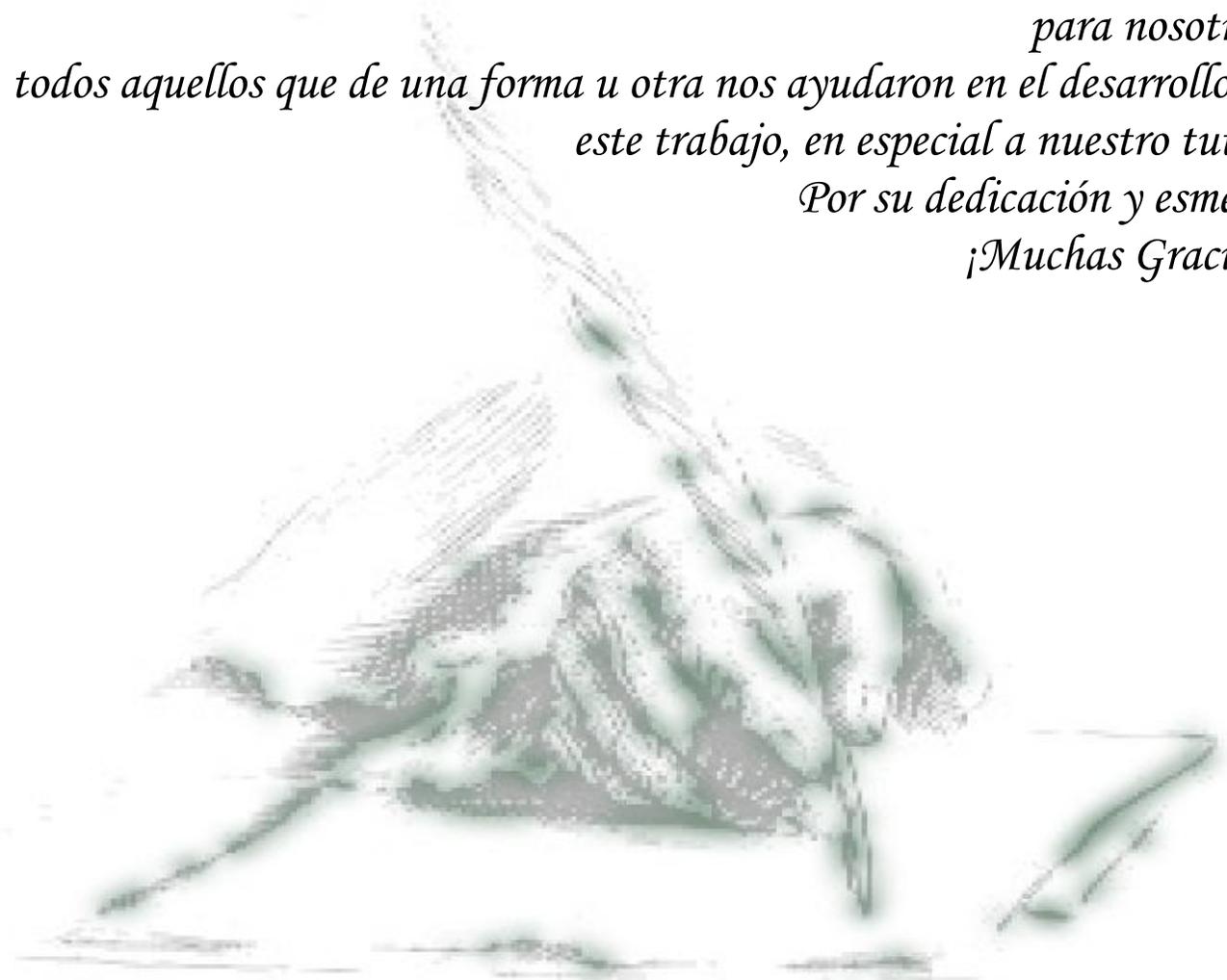
*A nuestros padres por todo el esfuerzo que han hecho para lograr nuestros sueños.*

*A nuestros amigos y compañeros de la vida que siempre estuvieron ahí para nosotros.*

*A todos aquellos que de una forma u otra nos ayudaron en el desarrollo de este trabajo, en especial a nuestro tutor.*

*Por su dedicación y esmero,*

*¡Muchas Gracias!*



## Resumen

Cuando se pretende ofrecer servicios médicos asistenciales de calidad y enfocados al paciente como su principal beneficiado, la realización de una correcta valoración de enfermería se convierte en un elemento de vital importancia. La calidad del diagnóstico, la planificación y ejecución de los planes de cuidados que deben adaptarse a las necesidades del paciente, tanto enfermo como sano, dependen en gran medida de la relevancia y consistencia de la información que se recolecta y organiza en la valoración.

Con estas ideas como punto de partida, se pretende viabilizar la valoración de enfermería basada en patrones. Para ello se propone la creación de un componente web configurable, flexible y escalable que permita la recogida, almacenamiento y utilización de la información procedente de la valoración. Para su desarrollo se emplea el entorno de desarrollo integrado Eclipse, el lenguaje de programación Java, el servidor de aplicaciones JBoss Application Server, el sistema gestor de base de datos PostgreSQL, el motor de persistencia Hibernate y el framework integrador JBoos Seam.

El componente web permitirá recoger la información que se desea, organizándola en patrones; para de esta manera ajustarlos a las necesidades de cada unidad de salud y configurar nuevos en caso de surgir. Esto traerá consigo que la información obtenida esté mejor organizada y sea de mayor utilidad tanto para el personal de enfermería como para los directivos y estadísticos de la salud.

**Palabras claves:** componente web, enfermería, patrones, Proceso de Atención de Enfermería (PAE), valoración.

# Tabla de Contenidos

Introducción.....	6
<b>Capítulo 1. Fundamentación teórica del componente web para la valoración de enfermería por patrones.....</b>	<b>13</b>
1.1 Marco conceptual.....	13
1.2 Enfermería en Cuba.....	14
1.3 La informática en la Enfermería .....	14
1.4 Análisis de soluciones existentes .....	15
1.5 Ambiente de desarrollo .....	18
<b>Capítulo 2. Propuesta del componente web para la valoración de enfermería por patrones.....</b>	<b>25</b>
2.1 Modelo de Dominio .....	25
2.2 Especificación de requerimientos de software.....	26
<b>Capítulo 3. Diseño del componente web para la valoración de enfermería por patrones. ....</b>	<b>31</b>
3.1 Patrones de arquitectura.....	31
3.2 Patrones de diseño.....	32
3.3 Definición de elementos de diseño.....	34
3.4 Diagrama de clases del diseño.....	34
<b>Capítulo 4. Implementación del componente web para la valoración de enfermería por patrones...53</b>	<b>53</b>
4.1 Propuesta de integración con otros módulos.....	53
4.2 Implementación.....	53
4.3 Diagrama de despliegue .....	53
4.4 Estándares de codificación y tratamiento de errores .....	55
<b>Conclusiones .....</b>	<b>58</b>
<b>Recomendaciones .....</b>	<b>59</b>
<b>Referencias Bibliográficas .....</b>	<b>60</b>
<b>Bibliografía.....</b>	<b>62</b>
<b>Glosario de términos .....</b>	<b>66</b>

## Introducción

La atención de la salud está sujeta a cambios y en los últimos años estos han estado muy ligados a la incorporación de nuevas tecnologías y al nuevo rol que juega el paciente como principal beneficiado. La enfermería, por tanto, no escapa a estos cambios y está evolucionando desde una concepción tradicional que identifica a estos profesionales como auxiliares del médico en la consulta y responsables de inyecciones y curas, hacia otra que identifica a la enfermería como una gestora de cuidados en materia de salud, de carácter personalizado, pero también de ámbito comunitario. (1)

“La enfermería abarca la atención autónoma y en colaboración dispensada a personas de todas las edades, familias, grupos y comunidades, enfermos o no, y en todas circunstancias. Comprende la promoción de la salud, la prevención de enfermedades y la atención dispensada a enfermos, discapacitados y personas en situación terminal”. (2)

El personal de enfermería lleva a cabo el Proceso de Atención de Enfermería (PAE), el cual es la “aplicación del método científico en la práctica asistencial y un sistema de planificación en la ejecución de los cuidados de Enfermería, de modo que se pueda ofrecer cuidados sistematizados, lógicos y racionales”. (3)

El PAE va dirigido a un objetivo, utilizando un enfoque organizado para lograrlo y responde a un cambio continuo. Está basado en las relaciones recíprocas que se establecen entre el personal de enfermería y el paciente, su familia y los demás profesionales de la salud. Se puede adaptar al ejercicio de la enfermería en cualquier lugar o área especializada que trate con individuos, grupos o comunidades y posee una base teórica concebida a partir de numerosos conocimientos que incluyen ciencias y humanidades. Además se puede aplicar a cualquier modelo teórico de enfermería. Por todas estas razones se puede concluir que es resuelto, sistemático, dinámico, interactivo y flexible. (4)

Los objetivos generales del PAE son: identificar las necesidades reales y potenciales del paciente, familia y comunidad; establecer planes de cuidados individuales, familiares o comunitarios; actuar para cubrir y resolver los problemas, prevenir o curar la enfermedad. (4) Para lograrlos se estructura en cinco fases, las cuales se llevan a cabo de forma cíclica y están estrechamente relacionadas entre sí.

A continuación se describen brevemente cada una de estas fases:

- **Valoración:** en esta etapa se recolectan y organizan los datos tanto de la persona como la familia y el entorno, con el fin de identificar las respuestas humanas y fisiopatológicas. Es la base para las decisiones y actuaciones posteriores.
- **Diagnóstico:** son analizados los datos recogidos en la valoración y usados como base para emitir un juicio clínico sobre los cuidados que necesita el paciente según su estado de salud.
- **Planificación:** consiste en elaborar estrategias diseñadas para reforzar las respuestas del paciente sano o para evitar, reducir o corregir las respuestas del paciente enfermo, identificadas en el diagnóstico de enfermería.
- **Ejecución:** se ejecuta el plan de enfermería descrito anteriormente, se realizan las intervenciones definidas en el proceso de diagnóstico donde es necesario el contacto directo del personal de enfermería con el paciente.
- **Evaluación:** se evalúa el progreso hacia las metas identificadas en las fases previas. Si la meta ha sido alcanzada, entonces el cuidado puede cesar. Se pueden identificar nuevos problemas en esta etapa, y así el proceso se reiniciará otra vez.

La primera y más importante fase es la valoración la cual es “un proceso planificado, sistemático, continuo y deliberado de recogida e interpretación de información que permite determinar la situación de salud que están viviendo las personas y su respuesta a esta” (5). Debe ser pensado y diseñado de antemano, requiere un método y un orden para su realización. Comienza cuando la persona entra en contacto con el sistema de cuidados de enfermería y continúa durante todo el tiempo que necesita de ellos. Precisa de una actitud reflexiva, voluntariosa, consciente y con un objetivo por parte de quien la ejecuta. (5)

La valoración es el primer paso y la base para determinar el estado de salud del paciente. Debe ser exacta porque los datos recogidos son fundamentales para la fase posterior; completa pues deben recogerse datos relevantes del paciente; orientada a los procesos vitales y recursos del paciente para evitar conclusiones precipitadas que puedan llevar a un error en el diagnóstico.

Para realizar la valoración se tienen en cuenta fases que están interrelacionadas:

**Recogida de información:** esta es una fase crucial, en ella toma partido la experiencia y capacidad de interacción con el paciente del profesional que la lleva a cabo. La utilidad, relevancia y compleción son características indispensables que ha de cumplir la información recolectada. Debe tener en cuenta:

- Tanto las fuentes directas (paciente valorado) como las indirectas (familia, amistades) y la utilidad que estas tienen.
- Los tipos de datos objetivos (temperatura, olor, color) y subjetivos (ideas expresadas por el paciente).
- Las técnicas de las enfermeras para recoger la información (observación, entrevistas y examen físico).

La recogida de la información se realiza en tres momentos:

- Antes: son recogidos los datos que no deben variar y que no brindan información sobre el estado de salud como el nombre, el sexo y la edad.
- Durante: se obtienen los datos mediante el examen físico, la observación y las respuestas del paciente a las preguntas que se le realizan. Deben de hacerse las anotaciones en el momento adecuado para evitar errores y olvidos.
- Después: se aclaran y completan los datos o aspectos concretos que generan dudas.

**Validación o verificación de los datos:** durante la recogida de datos se pueden cometer errores, que van desde la lectura y anotación incorrecta de la talla o el peso hasta la formulación ambigua de una pregunta y la interpretación incorrecta de una respuesta del paciente. Por tanto la información debe ser validada para confirmar que los datos recogidos son los exactos y completos. Para validar se suelen usar técnicas como: dobles comprobaciones, preguntas y comprobación según la observación.

**Organización de los datos:** los datos son agrupados en categorías o conjuntos de información que guardan cierta relación. Los criterios utilizados para organizarlos influirán en la determinación de un tipo de problema u otro.

**Comprobación de las primeras impresiones:** se decide lo que es relevante y lo que no. Puede reorientar y focalizar la valoración a obtener más datos o datos de otro tipo para determinar concretamente el estado de salud del paciente y que las primeras impresiones sean revalidadas o descartadas.

**Registro y comunicación de los datos:** se crea una constancia por escrito de toda la información que se obtuvo. Se hacen anotaciones concisas y concretas de forma legible. Esta constancia debe ser lo más cercana a la situación real del paciente.

El proceso de valoración de enfermería ha sido analizado y estudiado por investigadores de la salud como Virginia Henderson y Marjorie Gordon, además de la Asociación Norteamericana de Diagnósticos de Enfermería (NANDA por sus siglas en inglés), los cuales han aportado resultados significativos. A continuación se exponen aspectos generales de algunos de ellos:

- Necesidades humanas de Virginia Henderson. Estas “parten de la teoría de las necesidades humanas para la vida y la salud como núcleo para la acción de enfermería” (6). Este modelo que se compone de catorce necesidades puede utilizarse en todos los campos de acción, contempla la persona como un todo y reconoce su potencial de cambio. Tiene en cuenta aspectos objetivos y subjetivos del paciente relacionados con su fisiología, seguridad y autoestima.
- Patrones funcionales de Marjorie Gordon. Estos “son una configuración de comportamientos, generalmente comunes a todas las personas, que contribuyen a su salud, calidad de vida y al logro de su potencial humano” (7). De la valoración de los once patrones funcionales se obtienen datos físicos, psíquicos, sociales y del entorno del paciente, que facilitan su análisis. Son una división artificial y estructurada del funcionamiento humano integrado; la interrelación que se da entre ellos es lógica. Algunos patrones comparten información y ciertos datos pueden estar presentes en más de un patrón, no siendo necesario registrarlos de forma repetida.
- Patrones de respuestas humanas: Estos se centran en “las actuaciones que realiza una persona ante la presencia de procesos vitales y problemas de salud que le afectan directamente. Cuando estas respuestas no son efectivas, y alteran o pueden alterar el estado funcional de la persona, pueden ser abordadas por la enfermera a través de los diagnósticos de enfermería”. (8)

Dada la efectividad de estas investigaciones, varias son las personalidades que han aunado esfuerzos para promover su uso. Rosalinda Alfaro-LeFevre (9) describe la aplicación del PAE por etapas y recomienda en la fase de valoración la utilización de los patrones mencionados anteriormente. Además María José Aguilar Cordero (10) trata los aspectos sobre los cuidados del niño y su familia basados en los patrones funcionales.

También existen instituciones que actualmente utilizan estos patrones como guía para realizar la valoración. En el Hospital General Universitario de Valencia se utiliza la valoración de enfermería basada en patrones funcionales. El Instituto Municipal de Asistencia Sanitaria de Barcelona (IMAS) y el Departamento de Calidad de Enfermería diseñaron trayectorias clínicas para la gestión de los cuidados en un Hospital de IMAS donde utilizan registros basados en los patrones funcionales y las necesidades humanas.

El proceso de valoración de enfermería en Cuba se realiza recogiendo los datos del paciente según apreciaciones del personal de enfermería, mediante interrogatorios y exámenes físicos. El personal no cuenta con una guía que les oriente sobre la información que deben recoger y cómo organizarla, lo que trae implícito que los datos recogidos puedan sufrir de irrelevancia, inconsistencia y redundancia. Igualmente, no existe un lenguaje estándar que permita una correcta comunicación entre los profesionales de la salud, lo que trae consigo que en ocasiones se interprete erróneamente la información, se lleguen a conclusiones y se tomen decisiones basadas en datos incorrectos o alterados; que además se le confeccione un plan de cuidados al paciente que no responda a sus necesidades, por lo que se puede afectar su salud.

Por lo antes expuesto se define como **problema a resolver**: ¿cómo viabilizar el proceso de valoración de enfermería por patrones?

El problema se enmarca en el **objeto de estudio**: el proceso de valoración de enfermería.

Como **objetivo general** se define: desarrollar un componente web que permita al personal de enfermería realizar la valoración por patrones en el Sistema Integral para la Atención Primaria de Salud.

Centrándose en el **campo de acción**: valoración de enfermería por patrones.

Para dar cumplimiento al objetivo del trabajo se plantean las siguientes **tareas de investigación**:

- Análisis de bibliografía actualizada relacionada con la valoración de enfermería, determinando características relevantes de la misma, los procedimientos y mecanismos usados para llevarla a cabo.
- Análisis de las tendencias actuales de los sistemas de información de salud relacionados con la valoración de enfermería, estableciendo criterios sobre la factibilidad de estos con respecto al problema planteado.

- Determinación de la flexibilidad, escalabilidad y configuración que debe poseer el componente web para dar solución al problema existente.
- Aplicación de las herramientas y tecnologías propuestas por el Departamento de Atención Primaria de Salud (APS), para el desarrollo de aplicaciones web.
- Desarrollo de un componente web capaz de realizar la valoración de enfermería guiada por patrones en el Sistema Integral para la Atención Primaria de Salud y que cuente con las características determinadas.

Para el desarrollo de la investigación se utilizan los siguientes métodos teóricos:

**Histórico-Lógico:** fue utilizado para elaborar la fundamentación teórica del proceso de valoración de enfermería, permitiendo conocer lo más relevante de la enfermería, la informática aplicada a la salud, la valoración de enfermería, las herramientas y tecnologías que se utilizan para el desarrollo del trabajo. Además se utilizó para realizar un estudio de los sistemas existentes nacional e internacionalmente que sirven de apoyo al proceso de valoración de enfermería.

**Analítico-Sintético:** este método se utilizó para descomponer el proceso de valoración de enfermería en varias partes que posibilitaron una mejor comprensión del mismo.

El documento tiene una estructura de cuatro capítulos, que incluyen todo lo relacionado con el trabajo investigativo, así como el diseño e implementación del sistema.

En el **Capítulo 1. Fundamentación teórica del componente web para la valoración de enfermería por patrones:** se hace referencia a la Enfermería en Cuba, teniendo en cuenta la influencia de la informática en esta área. Se analizan los sistemas informáticos existentes tanto a nivel internacional como nacional. Además se realiza un análisis de las herramientas y tecnologías necesarias en el desarrollo del componente web.

En el **Capítulo 2. Propuesta del componente web para la valoración de enfermería por patrones:** se describe el componente web para la valoración de enfermería, se aborda lo referente al funcionamiento del dominio y se describen los requerimientos funcionales y no funcionales.

En el **Capítulo 3. Diseño del componente web para la valoración de enfermería por patrones:** se realiza el diseño del componente web para la valoración de enfermería por patrones, se describen los

patrones del diseño utilizados así como el patrón arquitectónico. Se definen los elementos del diseño y se describen los diagramas de clases del mismo.

En el **Capítulo 4. Implementación del componente web para la valoración de enfermería por patrones**: trata los aspectos relacionados con la construcción del componente web para la valoración de enfermería por patrones, se modela el diagrama de despliegue, se aborda la descripción de los estándares de codificación y el tratamiento de errores para el componente web.

# Capítulo 1. Fundamentación teórica del componente web para la valoración de enfermería por patrones

En este capítulo se ofrecen los conceptos fundamentales relacionados con el tema a desarrollar. Se realiza un análisis del estado del arte y se describe la guía de desarrollo, las tecnologías y las herramientas a utilizar para el desarrollo del componente web.

## 1.1 Marco conceptual

**Valoración:** método sistemático de recogida de datos que consiste en la apreciación del sujeto con la finalidad de identificar las respuestas que este manifiesta en relación con las diversas circunstancias que le afectan.

**Necesidades humanas:** las necesidades son la expresión de lo que un ser vivo requiere indispensablemente para su conservación y desarrollo. Es la sensación de carencia unida al deseo de satisfacerla. Los recursos para producir bienes y servicios son escasos y las necesidades humanas son ilimitadas, de ahí que el problema básico que se presenta en todas las sociedades sea la escasez.

**Respuestas humanas:** fenómenos que preocupan a los enfermeros, que se centran en dos tipos de respuestas humanas:

- Reacciones del individuo y grupos ante problemas reales de salud: restablecimiento de la salud tales como el impacto de una enfermedad, efectos sobre el individuo y familia y las necesidades de autocuidado.
- Preocupación de individuos o grupos por posibles problemas de salud: respuestas de apoyo a la salud, como el control y la información en poblaciones o comunidades de riesgo, con necesidades educativas de información, técnicas, actividades orientadas a la salud y cambios de comportamiento.

**Patrones funcionales:** “los patrones funcionales de salud describen una serie de once áreas relativas a la salud. Tienen un enfoque funcional, de desarrollo y cultural que permite su aplicación a todos los ámbitos, especialidades y grupos de edad. Los patrones son configuraciones de comportamientos contruidos desde las descripciones del paciente y las observaciones de la enfermera. Entender un patrón

puede ser difícil hasta que no se hayan valorado todos los patrones debido a la naturaleza interdependiente de los patrones”. (11)

Se desarrollan a partir de la interacción del paciente y su ambiente. Están influidos por factores biológicos, evolutivos, culturales, sociales y espirituales.

Los patrones funcionales se adaptan al ámbito de la Atención Primaria de Salud (APS), permitiendo estructurar y ordenar la valoración, tanto a nivel individual como familiar y comunitario, a lo largo del ciclo vital de la persona, en situaciones de salud o enfermedad; y facilitan claramente la siguiente fase de diagnóstico enfermero al estar en el origen de la clasificación por dominios de la NANDA.

## **1.2 Enfermería en Cuba**

Los servicios de enfermería en Cuba se integran a la atención médica y social del Sistema Nacional de Salud (SNS), constituyen una red de servicios diferenciados por niveles de atención que están presentes desde la atención a la comunidad en la atención primaria hasta la atención terciaria. Estos niveles de atención garantizan los servicios de enfermería en el equipo de salud, en una amplia gama de actividades que van desde las actividades de prevención, apoyo asistencial y de seguimiento a pacientes; hasta la rehabilitación de su salud individual y familiar. El personal de enfermería está presente en las campañas y acciones de vacunación, visitas de terreno, en consultorios e incluso en el domicilio de los pacientes cuando es necesario.

En la actualidad, Cuba, a través de su Sociedad Nacional de Enfermería es miembro de la Asociación de Profesionales de Enfermería de Centroamérica y el Caribe, de la cual forma parte desde la década del 90 y trabaja en el fortalecimiento de las redes de enfermería en la región, a fin de mantener los resultados alcanzados en las metas del milenio hasta el año 2015.

La enfermería en Cuba juega un papel activo en las transformaciones que se realizan en la actualidad en el SNS para dar respuesta a los tres grandes retos del sistema de salud, elevar el grado de satisfacción de la población, hacer el sistema sostenible y elevar los indicadores de salud de la población.

## **1.3 La informática en la Enfermería**

Las funciones asistenciales, docentes, administrativas e investigativas de los profesionales de la enfermería; la implementación de nuevas tecnologías, el necesario manejo de amplios volúmenes de datos de los pacientes, hacen imprescindible una disposición, manejo y tratamiento de la información.

Como la informática responde adecuadamente a estas nuevas necesidades, la misma ha ido en desarrollo en el ámbito global de la enfermería.

En la actualidad se cuenta con varios sistemas para el soporte de la actividad de enfermería. En el ámbito docente educativo como en el asistencial se encuentran en fase de perfeccionamiento para su posterior aplicación en las diferentes áreas de la salud, específicamente en la actividad de enfermería.

La informática aplicada a la enfermería permite la gestión de la información en el procesamiento y la tecnología de la comunicación. Es esencial para la adopción de decisiones y para la práctica de enfermería de calidad. La información y los conocimientos adquiridos gracias a la aplicación de la informática en la enfermería pueden ayudar a conocer y a comprender mejor los asuntos relacionados con la enfermería y la atención de salud. Tiene por objeto conseguir que se mantenga una perspectiva clínica y se fomente la investigación que sirva directamente para mejorar los cuidados dispensados a los pacientes.

## **1.4 Análisis de soluciones existentes**

### **Ámbito Internacional**

#### **Gestión de Cuidados de Enfermería Línea Abierta**

La aplicación informática GACELA (Gestión de Cuidados de Enfermería Línea Abierta) es un sistema diseñado fundamentalmente para el trabajo de enfermería enfocado en los registros y en la elaboración de planes de cuidados (12). Permite una valoración inicial, continuada y focalizada, para el registro de la evolución de los pacientes, planificando los cuidados necesarios y garantizando continuidad en el proceso asistencial.

Cuenta con una versión para *Personal Data Assistant* (PDA) llamada *GACELA Pocket*. El Gacela tanto en su versión para computadoras convencionales como en la versión para PDA fue realizado para ejecutarse única y exclusivamente sobre el sistema operativo Windows. Realiza un estudio descriptivo de los pacientes mediante un plan de cuidados que tiene en cuenta la valoración por patrones funcionales de Marjorie Gordon.

Los usuarios de GACELA han identificado problemas tales como:

- Interfaz con poca usabilidad: es poco atractiva, poco evidente y, a veces, tediosa.
- Poca agilidad del sistema a la hora de realizar un plan de cuidados.

## **Oficina Médica Informatizada-Atención Primaria**

Existen dos aplicaciones que son utilizadas en la Comunidad de Madrid en Atención Primaria (AP), el OMI (Oficina Médica Informatizada) que presenta dos variantes: de forma generalizada en Atención Primaria es OMI-AP y en Atención Especializada es OMI-AE.

OMI-AP es un programa informático para la atención sanitaria integral de la población que incorpora un desarrollo importante de los planes de cuidados. Tiene una amplia implantación y utilización en numerosas comunidades autónomas, lo que le convierte en una herramienta informática que puede aportar grandes beneficios a la profesión enfermera tales como: unificar el sistema de trabajo de la enfermera, favorecer la aplicación sistemática de cuidados a la población, permitir la evaluación del trabajo de la enfermera y conseguir una base de datos.

El OMI-AP utiliza una estructura cliente-servidor para lo cual requiere:

- Un servidor con sistema operativo Windows 2003 Server, antivirus e Informix como gestor de base de datos.
- En los clientes un sistema operativo Windows XP, antivirus, paquete ofimática y cliente de Informix.

## **CARESUITE**

Es un sistema comercializado por un proveedor global de innovadoras soluciones de información en las áreas de cuidados críticos llamado PICIS. Ha supuesto un impacto muy importante en el sistema de trabajo de las enfermeras. Es un sistema web usa un Microsoft Internet Information Server (IIS) que se ejecuta sobre un sistema operativo Windows. Esta aplicación cuenta con datos referentes a la valoración de enfermería que se han introducido de dos formas, por respuestas fisiopatológicas y por patrones funcionales con la posibilidad de aplicar diagnósticos de enfermería.

## **LogicPAE**

Es un software inteligente centrado en:

- La gestión de pacientes.
- La generación de los diagnósticos de enfermería.
- La automatización de todos los procesos: valoraciones, constantes, escalas, diagnósticos, manifestaciones de dependencia, factores relacionados, resultados NOC (Clasificación de los Resultados de Enfermería), intervenciones NIC (Clasificación de las Intervenciones de Enfermería) y actividades de enfermería.
- La gestión de dietas y servicios.
- El control de la medicación y patologías.
- El registro de curas.

Puede ejecutarse sobre equipos con sistemas operativos Windows XP, Windows VISTA o Windows 7, de manera general su licencia es privativa y posee una versión limitada que es gratuita.

## **Ámbito Nacional**

### **Sistema Informático en Enfermería**

El CECAM (Centro de Cibernética Aplicada a la Medicina) ha desarrollado algunas investigaciones referentes a la Enfermería. Una de estas es el sistema para la gestión del cuidado o Sistema Informático en Enfermería (SICENF) desarrollado por el Msc. Gustavo Sánchez Fernández y la Msc. Yoadis Cuesta García, el cual fue aplicado en la Unidad de Cuidados Intermedios del Hospital de Guantánamo. Este brinda mejores opciones de consistencia, posibilidad y facilidad de acceder y reutilizar la información frente al método tradicional. El sistema contiene una base de datos que permite el almacenamiento, recuperación y actualización sistemática de todo el proceso de información de cuidado de enfermería que se administra.

No obstante, pese todas las ventajas vistas del soporte digitalizado del plan de cuidados, la necesidad de dar continuidad al PAE en servicios ubicados en locales que no poseen terminales de computadoras, obliga a generar el informe impreso del mismo para archivar en la Historia Clínica, una vez que el paciente sea traslado a otra sala o servicio.

## **Sistema Integral para la Atención Primaria de Salud**

El Sistema Integral para la Atención Primaria de Salud (alasiAPS), del Centro de Informática Médica (CESIM), subordinado a la Universidad de las Ciencias Informáticas (UCI) cuenta con funcionalidades que permiten realizar el proceso de valoración de enfermería. Sin embargo se basa solo en la recogida y organización de los datos basada en patrones funcionales e ignora las valoraciones de enfermería basadas en patrones de respuestas y necesidades humanas, los cuales aportan datos significativos y específicos del paciente, que no se tienen en cuenta en los patrones funcionales. Dichas funcionalidades no permiten ajustar estos patrones a las necesidades de cada unidad de salud.

Los sistemas anteriormente analizados no cumplen las políticas de migración a software libre del gobierno cubano. Todos son mono plataforma que pueden ser ejecutados sobre sistemas operativos Windows XP o superiores. Estos sistemas, con excepción del SICENF, son distribuidos bajo licencias privativas, lo que representa una cuantiosa inversión. Para comprender la magnitud del capital necesario, se puede analizar el software LogicPAE; el cual cuenta con un módulo que realiza valoración de enfermería por patrones. Cada licencia cuesta 278.81 euros por unidad (13), lo que requiere 3 334 009.98 euros anualmente para llevar dicho módulo a los 11 506 consultorios médicos y 452 policlínicos (14) con los que cuenta el SNS. Esta cifra no incluye los cursos de adiestramiento que requiere su uso.

Estos sistemas no permiten en su mayoría realizar la valoración de enfermería por patrones de forma general y solo se enfocan principalmente en la valoración por patrones funcionales. Dado que son aplicaciones de escritorio, excepto el CARESUITE y el alasiAPS, su mantenimiento y actualización son engorrosos debido a que es necesario actualizar cada una de las estaciones de trabajo en las que están desplegados. Por los elementos antes expuestos, se concluye que el software existente no brinda una solución total al problema planteado.

## **1.5 Ambiente de desarrollo**

### **1.5.1 Sistemas distribuidos. Modelo Cliente-Servidor**

“Sistema que se apoya en terminales (clientes) conectadas a una computadora que los provee de un recurso (servidor). De esta manera los clientes son los elementos que necesitan servicios del recurso y el servidor es la entidad que lo posee. Los clientes, sin embargo, no dependen totalmente del servidor debido a que pueden realizar los procesamientos para desplegar la información (por ejemplo en forma gráfica). El servidor los provee únicamente de la información sin hacerse cargo de otros procesos de

forma que el tráfico en la red se ve aligerado y las comunicaciones entre las computadoras se realizan más rápido”. (15)

## 1.5.2 Framework, librerías y componentes

Los frameworks son un conjunto de clases cooperativas que construyen un diseño reutilizable para un tipo específico de software. Un framework proporciona la arquitectura del diseño en clases abstractas y define sus responsabilidades y colaboraciones. Un desarrollador realiza una aplicación, haciendo subclases y creando instancias a partir de las clases definidas por el framework.

### Java Server Faces 1.2

La tecnología JSF (*Java Server Faces*) constituye un framework de interfaces de usuario del lado del servidor para aplicaciones web basadas en tecnología Java.

Toda aplicación JSF debe incluir una librería de etiquetas personalizadas que define las etiquetas que representan componentes de Interfaz de Usuario (UI por sus siglas en inglés), así como una librería de etiquetas para controlar otras acciones importantes, como validadores y manejadores de eventos. La librería de etiquetas de componentes elimina la necesidad de codificar componentes UI en HTML (*Hyper Text Markup Language*) u otro lenguaje de marcas, lo que se traduce en el empleo de componentes completamente reutilizables.

### RichFaces 3.3.1

RichFaces es una librería de componentes visuales para JSF, que permite la integración de funcionalidades Ajax (*Asynchronous JavaScript And XML*) en dichos componentes, mediante el soporte de la librería Ajax4JSF. Se integra perfectamente en el ciclo de vida de JSF y contiene un conjunto de componentes visuales con un número bastante amplio que cubren casi todas las necesidades.

### Ajax4JSF

Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código JS (*Java Script*). Ajax4jsf presenta mejoras sobre los propios beneficios del framework JSF, incluye en 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.

## JavaScript

JS es un lenguaje orientado a objetos ligero, más conocido como el lenguaje de *script* para páginas web, pero también usado en muchos entornos sin navegador, que soporta la construcción de objetos basado en prototipos. La sintaxis básica es similar a Java y C++ con la intención de reducir el número de nuevos conceptos necesarios para aprender el lenguaje.

## Facelets 1.1

Facelets es un motor de plantillas y 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 2.1.1

JBoss Seam es un framework que integra y unifica los distintos estándares de la plataforma Java EE 5.0, pudiendo trabajar con todos ellos siguiendo el mismo modelo de programación. Ha sido diseñado intentado simplificar al máximo el desarrollo de aplicaciones, basando el diseño en *Plain Old Java Objects* (POJOs) con anotaciones. Estos componentes se usan desde la capa de persistencia hasta la de presentación, poniendo todas las capas en comunicación directa. Integra la capa de presentación (JSF) con la capa de negocios y persistencia EJB (*Enterprise Java Beans*), ya que su núcleo principal está formado por las especificaciones EJB y JSF.

## Hibernate 3.3

Hibernate parte de una filosofía de mapear objetos Java, también conocidos como POJOs. Una de las principales características de Hibernate es su flexibilidad, envolviéndolo todo bajo un framework común. Es una capa de persistencia objeto/relacional que permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. Es un software libre y está bajo licencia LGPL.

## Sistema Gestor de Base de Datos PostgreSQL 8.4

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.

PostgreSQL utiliza un modelo cliente/servidor que usa multiprocesos y no multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

### 1.5.3 Lenguaje de programación

Se utiliza Java en su versión 1.6 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 cada día más por la 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, utiliza XML como tipo de conexión de red entre cualquier sistema.

### 1.5.4 Guía de Desarrollo de Software

La creciente dimensión de los proyectos relacionados con los sistemas de información hace que sea necesario acometerlos con la mayor solidez metodológica, por esto la UCI certificó el nivel 2 de CMMI (*Capability Maturity Model Integration*).

#### Modelo para la mejora y evaluación de procesos

CMMI es un modelo para la mejora y evaluación de procesos en el desarrollo, mantenimiento y operación de sistemas de software. Las mejores prácticas de CMMI se publican en documentos llamados modelos, los cuales contienen las prácticas 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.

La UCI se encuentra en el Nivel 2 de CMMI, es la primera institución cubana certificada con este modelo. Los beneficios con este nivel alcanzado son entre muchos: la mejora del ciclo de vida dentro del desarrollo de software, mayor productividad, calidad de los productos y servicios que ofrece la universidad a sus clientes y por ende la satisfacción de los mismos.

“Es utilizado el documento IPP- 3510:2009 (Libro de Proceso para la Administración de Requisitos) confeccionado 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 y se establece por cada fase la relación con los subprocesos descritos en el libro de procesos específico del área Administración de Requisitos”. (16)

“La descripción del IPP-3510:2009 incluye la definición de roles, sus responsabilidades y las habilidades en la ejecución de las actividades de los distintos procesos y los productos típicos de trabajos que se obtienen como resultado de la ejecución de dichas actividades. La información contemplada en cada producto típico de trabajo, está registrada basándose en la experiencia de utilización de los expertos de la metodología Proceso Unificado de Desarrollo (RUP por sus siglas en inglés).” (16)

## **1.5.6 Lenguaje Unificado de Modelado**

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es un modelo de procesos de negocio. Este es el empleado en el presente negocio.

Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que han existido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los analistas sólo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Ofrece nueve diagramas en los cuales modelar sistemas. Es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos.

## **1.5.7 Herramientas**

Toda metodología de desarrollo de sistemas debe sustentarse sobre la base de una herramienta de Ingeniería de Software Asistida por Computadora (CASE por sus siglas en inglés) que garantice un mejor análisis del sistema y la calidad del resultado final. Las herramientas utilizadas para el desarrollo del componente web fueron:

## **Visual Paradigm 6.4**

Es una herramienta que utiliza a UML como lenguaje de modelado. Apoya los estándares más altos de las notaciones de Java y de UML. Genera productos de calidad, soporta aplicaciones web y es fácil de instalar y actualizar. En sentido general reduce significativamente los esfuerzos en todas las etapas del ciclo de vida de desarrollo de software.

## **Eclipse SDK 3.5.2**

Para el desarrollo del software se utilizará el Eclipse, que es un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) para aplicaciones con java. Está soportado por IBM (*International Business Machines Corporation*), es un proyecto de código abierto multiplataforma para desarrollar lo que el proyecto llama Aplicaciones de Cliente Enriquecido. Eclipse no es tan sólo un IDE, se trata de un framework modular ampliable mediante complementos (plugins). De hecho, existen complementos que permiten usar Eclipse para programar en PHP, Perl, Python, C/C++.

## **PgAdmin 1.10.5**

Es una herramienta de código abierto para la administración de bases de datos PostgreSQL. Incluye:

- Interfaz administrativa gráfica.
- Herramienta de consulta SQL (con un EXPLAIN gráfico).
- Editor de código procedural.
- Agente de planificación SQL/shell/batch.
- Administración de Slony-I.

PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas.

La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, entre ellos Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris.

Los servicios de enfermería en Cuba son una parte importante dentro de la atención médica y social del SNS. Los sistemas informáticos relacionados con la valoración de enfermería que fueron analizados no satisfacen las necesidades del proceso de valoración de enfermería en el SNS, teniendo en cuenta las características políticas y económicas del país. La guía de desarrollo utilizada permite la calidad y organización del componente web, así como las herramientas y tecnologías brindaron un ambiente robusto y seguro.

---

## Capítulo 2. Propuesta del componente web para la valoración de enfermería por patrones

En este capítulo se explican los conceptos fundamentales abordados en el desarrollo del Modelo de Dominio facilitando la comprensión de estos. Se especifican los requisitos funcionales que deben ser automatizados y se describen las características del sistema a través de los requisitos no funcionales.

### 2.1 Modelo de Dominio

Un modelo de dominio captura los tipos de objetos más importantes en el contexto del sistema. Los objetos del dominio representan los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio.

El objetivo del modelado de dominio es comprender y describir las clases más importantes dentro del contexto del sistema. Para una mayor comprensión del contexto en que se desarrolla el componente web, se definen los principales conceptos relacionados con el entorno.

#### 2.1.1 Conceptos fundamentales del dominio

Para brindar una mejor comprensión del Diagrama del Modelo de Dominio a continuación se realiza una breve descripción de los conceptos encontrados en el proceso de valoración de enfermería.

**Valoración:** método para el proceso de recogida y organización de datos.

**Enfermera:** integrante del personal de enfermería que realiza la valoración.

**Paciente:** persona que es objeto de valoración.

**Patrón:** forma de organización de la información que ayudará a la identificación de problemas y diagnósticos enfermeros.

**Tipo Patrón:** clasificación del patrón.

**Variable:** abstracción de la información recogida que puede tomar diferentes valores en cada valoración y cada paciente.

**Posible valor:** valor que puede tomar la variable.

**Tipo dato:** tipo de dato de la variable.

**Grupo:** clasificación que agrupa a un conjunto de variables relacionadas entre sí.

## 2.1.2 Diagrama del Modelo de Dominio

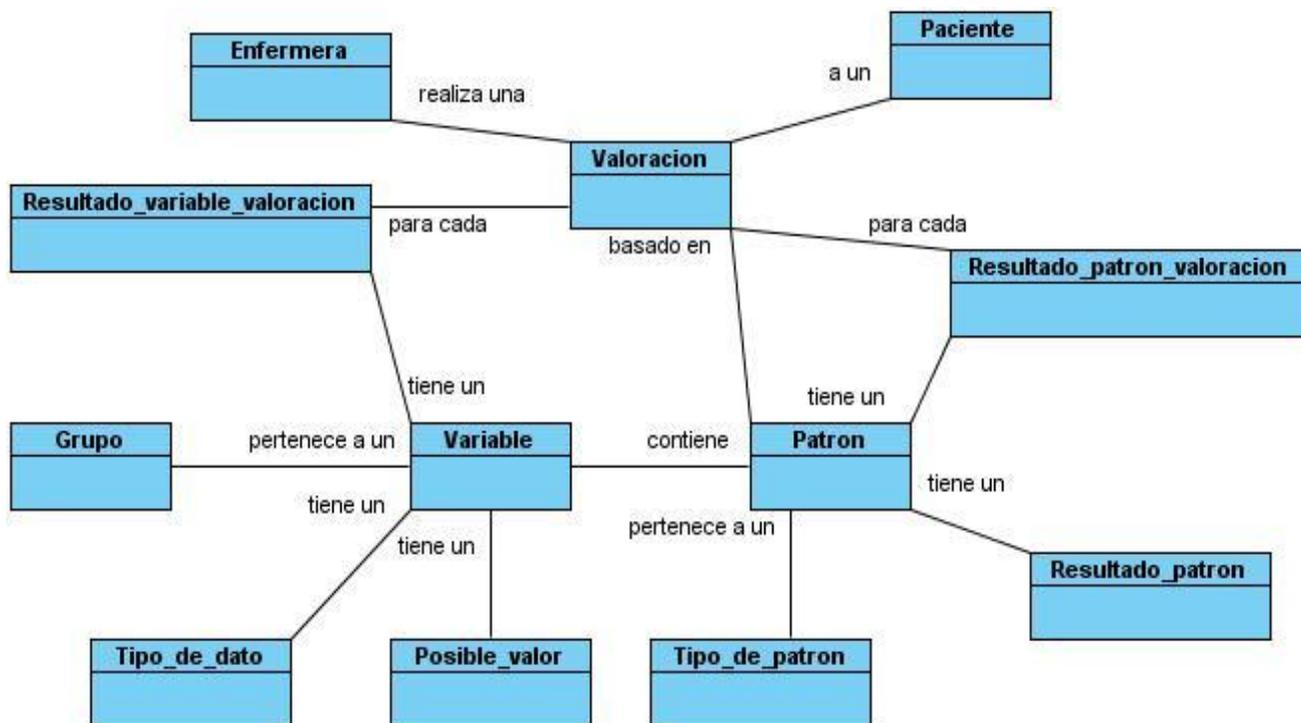


Figura 1. Diagrama del Modelo de Dominio.

## 2.2 Especificación de requerimientos de software

### 2.2.1 Requerimientos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema. Especifican la manera en que éste debe reaccionar a determinadas entradas. Especifican cómo debe comportarse el sistema en situaciones particulares. Pueden declarar explícitamente lo que el sistema no debe hacer.

### Requerimientos funcionales del componente web

- **RF 1 Realizar valoración:** crea la valoración con los datos fundamentales de los pacientes según los tipos de patrones existentes.
- **RF 2 Ver detalles de valoración:** muestra toda la información recogida de la valoración.
- **RF 3 Gestionar configuración de patrón:** permite realizar un conjunto de acciones (agregar, listar, modificar y eliminar) a los patrones.
- **RF 3.1 Agregar patrón:** agrega nuevos patrones con sus respectivas configuraciones.
- **RF 3.2 Listado patrón:** muestra todos los patrones configurados y dados criterios de búsqueda, realiza una búsqueda de la información.
- **RF 3.3 Modificar patrón:** al seleccionar un patrón ya configurado se pueden modificar sus datos.
- **RF 3.4 Eliminar patrón:** se selecciona el(los) patrones que van a eliminarse (cambian del estado Falso a Verdadero y viceversa, pues no se eliminan físicamente de la base de datos) y primeramente hay que eliminar las variables y grupos que lo conforman.
- **RF 3.5 Ver detalles de patrón:** muestra los datos de los patrones.
- **RF 3.6 Gestionar variable.**
- **RF 3.6.1 Agregar variable:** agrega variables independientes o que pueden pertenecer a un grupo.
- **RF 3.6.2 Modificar variable:** modifica variables que previamente son seleccionadas.
- **RF 3.6.3 Eliminar variable:** se elimina la variable seleccionada (cambian del estado Falso a Verdadero y viceversa, pues no se elimina físicamente de la base de datos).
- **RF 3.7 Gestionar grupo.**
- **RF 3.7.1 Agregar grupo:** agrega nuevos grupos que tendrán variables.
- **RF 3.7.2 Eliminar grupo:** se elimina el grupo seleccionado (cambia de estado pues no se elimina físicamente de la base de datos) pero deben eliminarse previamente las variables que contiene.

### 2.2.2 Requerimientos no funcionales

Describen las propiedades o características que el producto debe tener. Cada requerimiento funcional tiene asociados requerimientos no funcionales. Algunos pueden afectar a nivel de eventos, otros a todo el producto. Responden a las características en cuanto a funcionabilidad, usabilidad, confiabilidad, compatibilidad con hardware y software.

Son restricciones de los servicios o funciones ofrecidas por el sistema (fiabilidad, tiempo de respuestas, capacidad de almacenamiento). Surgen de las necesidades del usuario (restricciones de presupuesto, políticas de la organización, necesidad de interoperabilidad). Define restricciones adicionales con las que el sistema deberá contar.

#### Requerimientos no funcionales del componente web

##### Confiabilidad

**RNF 1:** Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan.

**RNF 2:** Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.

**RNF 3:** Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la base de datos, independientemente de que para el sistema, este elemento ya no exista.

**RNF 4:** El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

**RNF 5:** Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo.

##### Eficiencia

**RNF 6:** El sistema minimizará el volumen de datos en las peticiones y optimizará el uso de recursos críticos como la memoria.

##### Restricciones de diseño

**RNF 7:** La capa de presentación contendrá todas las vistas y la lógica de la presentación.

---

**RNF 8:** La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden ser ejecutados por cada usuario.

**RNF 9:** La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas.

**RNF 10:** El acceso a datos está basado en el estándar JPA (*Java Persistence API*) y en Hibernate que es la implementación del mismo.

**RNF 11:** Las interfaces se realizarán por el estándar definido en el departamento Atención Primaria de Salud (APS).

### **Interfaz**

**RNF 12:** Las ventanas del sistema contendrán los datos claros y bien estructurados para facilitar la interpretación correcta de la información.

**RNF 13:** La entrada de datos incorrecta será detectada claramente e informada al usuario.

**RNF 14:** Todos los textos y mensajes en pantalla aparecerán en el idioma inherente al país o región donde se despliegue el sistema.

### **Requerimientos de hardware**

**RNF 15:** Las estaciones de trabajo debe tener como mínimo 256 Mb de memoria RAM y un microprocesador de 2.0 Hz.

**RNF 16:** Los servidores de base de datos deberán tener: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.

**RNF 17:** Los servidores de aplicaciones deberán tener: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.

**RNF 18:** Los servidores de Intercambio deberán tener: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

### **Requerimientos de software**

**RNF 19:** El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA, el servidor de aplicaciones JBoss Application Server y el sistema gestor de base de datos PostgreSQL.



**RNF 20:** El sistema dispondrá de un navegador web Firefox 2 o superior.

El modelo de dominio es como un diccionario o glosario de los términos fundamentales que se utilizan para el desarrollo del componente web. Además los requerimientos funcionales y no funcionales describen las capacidades y cualidades que debe poseer el componente web, obteniéndose una perspectiva de su desarrollo.

---

## Capítulo 3. Diseño del componente web para la valoración de enfermería por patrones.

El propósito de este capítulo está encaminado a adquirir una comprensión de los aspectos relacionados con los requerimientos, lenguajes de programación, componentes reutilizables, tecnologías de distribución y concurrencia. Entre los artefactos que serán mostrados en el presente capítulo se encuentran: el Modelo de Diseño, especificándose la estructura y la definición de los elementos que este posee; Diagramas de Clases del Diseño y descripción de las mismas.

### 3.1 Patrones de arquitectura.

Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos.

Las vistas son las clases interfaz con las que interactúa el usuario y están construidas mayormente con componentes JSF. Usa la librería de componentes RichFaces que se integra con el framework JBoss Seam y permite generar vistas no necesariamente basadas en HTML. Incluye conversión y validación de campos establecimiento de reglas de navegación declarativas, la internacionalización y accesibilidad de la interfaz de usuario. Un modelo orientado a eventos y combinado con Facelets, elimina la necesidad de dos motores de renderización mejorando el rendimiento.

El controlador abarca la lógica del negocio, está representado por las clases controladoras que son las encargadas de responder a las solicitudes del usuario. Usa como framework de integración JBoss Seam. Además integra la capa de presentación (JSF) con la de negocios y persistencia (EJB) facilitando la comunicación entre estas dos capas.

El modelo se encarga del almacenamiento y acceso a los datos para lo cual utiliza Hibernate 3.3 para la implementación de JPA, lo cual posibilita minimizar las configuraciones en XML, la eliminación de una gran parte del código infraestructural en cuanto a transacciones y la transmisión del contexto de persistencia. 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:** sirve para evitar tener en memoria dos representaciones distintas del mismo objeto en una transacción de negocio; funciona como una caché de objetos de negocio, lo cual también contribuye a reducir el número de accesos a la base de datos.
  - **Identity field:** consiste en generar un número y administrarlo; puede ser, por ejemplo, un contador en la base de datos, o un atributo de una clase genérica en memoria.
  - **Foreign key mapping:** sirve para mapear relaciones de uno a muchos; una relación de este tipo, que en memoria consiste en un objeto A con una colección de objetos B dentro, se mapea con una clave foránea del lado de la entidad contenida; la relación se invierte en la base de datos.
  - **Association table mapping:** sirve para mapear relaciones de muchos a muchos; en una base de datos esta relación se resuelve con la incorporación de una tercera tabla que contenga las claves primarias de ambas entidades involucradas en la relación.
  - **Lazy load:** resuelve problemas de carga desmesurada y dependencias circulares; el ejemplo clásico es una relación uno a muchos, donde un objeto de negocio posee una colección de otros objetos; cuando el objeto A es solicitado, se trae a memoria el objeto A y sólo el objeto A, con la colección de objetos B vacía; luego, los objetos B serán leídos cuando realmente se necesiten.

### 3.2 Patrones de diseño.

Cada patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema, de forma que puede utilizarse un millón de veces sin tener que hacer dos veces lo mismo.

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se debe tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios).

---

En el desarrollo del componente web se utilizan múltiples patrones de diseño, entre ellos se encuentran los Patrones de Software para la Asignación General de Responsabilidades (GRASP por sus siglas en inglés). Constituyen un apoyo para entender el diseño del 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.
- **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 acciones 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; así potenciar la reutilización y disminución de 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). 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 modelo de diseño y con el objetivo de mejorar la calidad de los diagramas correspondientes a esta disciplina fueron empleados los patrones mencionados anteriormente.

### 3.3 Definición de elementos de diseño.

El modelo de diseño está constituido por los diagramas de clases de diseño. En el mismo existen elementos que por su vital importancia, son imprescindibles de conocer para su posterior entendimiento. Ellos son:

- **Páginas Clientes (*Client Page*):** Son las páginas web, las cuales están del lado del cliente y proveen la interacción directa con el usuario, estas se ejecutan sobre un navegador.
- **Páginas Servidoras (*Server Page*):** Se ejecutan del lado del servidor. Su objetivo es proveer una respuesta a las peticiones realizadas desde la vista. Gestionan una acción específica.
- **Formularios (*Form*):** Contienen un conjunto de elementos de entrada que están contenidos en las páginas clientes. Su función es enviar directamente la información a las páginas servidoras.

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:** Diagramas de Clases del Diseño. <Nombre de la funcionalidad>.

Para representar las clases se usó la siguiente nomenclatura, frmsiaps, <Nombre de la vista>.siaps, C<Nombre de la Opción>.java (Realizar), CD<Nombre de la Opción>.java (Detalles), CC<Nombre de la Opción>.java (Agregar), CL<Nombre de la Opción>.java (Listar), CM<Nombre de la Opción>.java (Modificar) y <Tr o Tn o Tb><Nombre de la Opción>.java, para los formularios, páginas clientes, clases servidoras y entidades respectivamente ubicadas en los diferentes niveles de la aplicación.

### 3.4 Diagrama de clases del diseño.

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema, muestra sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

3.4.1 Diagrama de clases del diseño. Realizar valoración

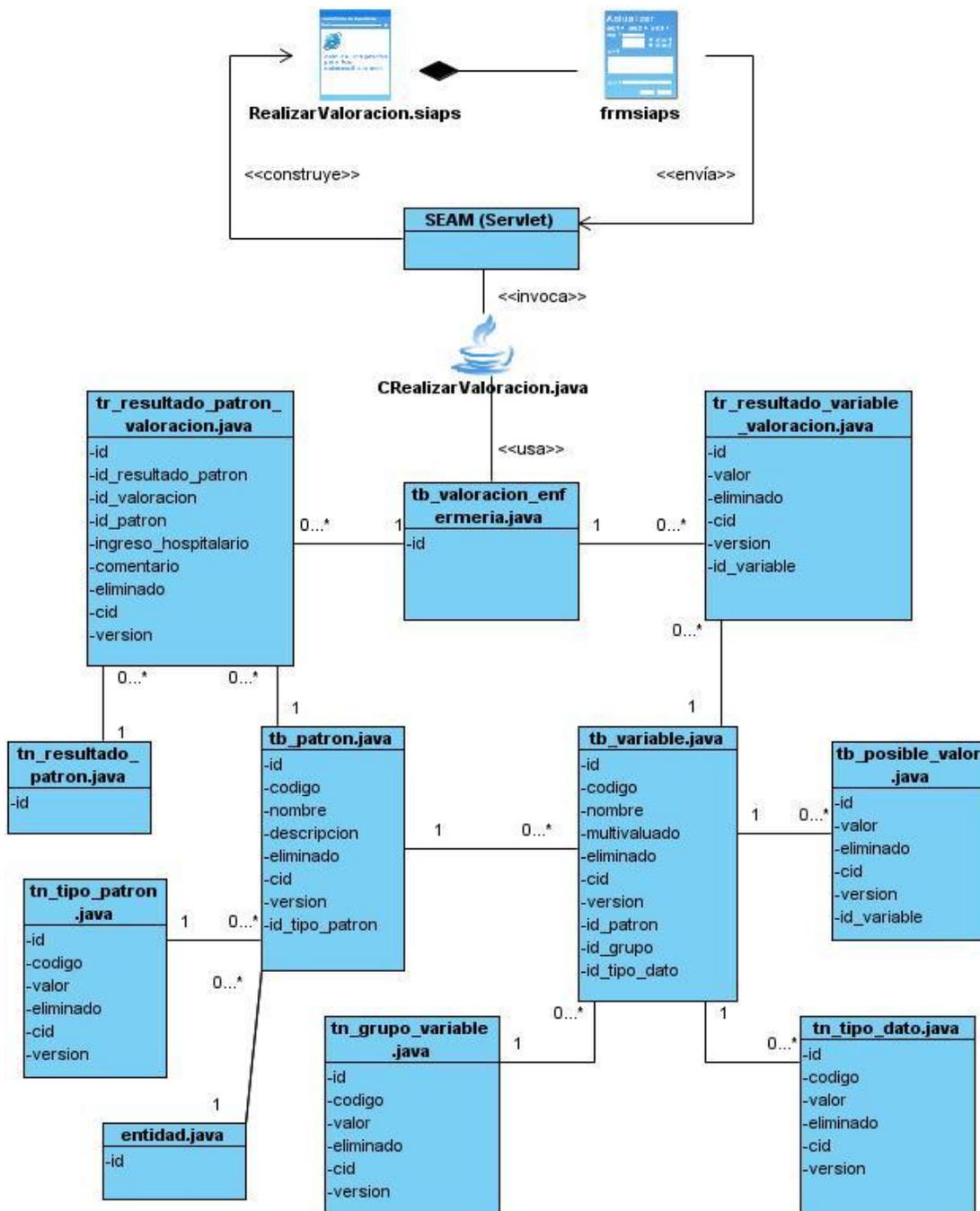
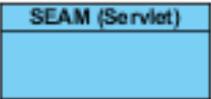
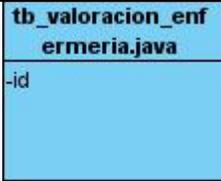
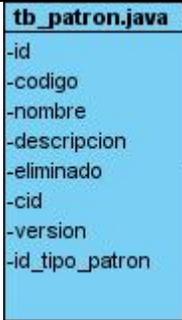


Figura 2. Diagrama de Clases del Diseño. Realizar valoración.

3.4.2 Descripción textual.

Capa de Presentación	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 3.</b> RealizarValoracion.siaps</p>	Proveer la interacción con el usuario.
<b>Descripción:</b>	
La clase RealizarValoracion.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite a los usuarios realizar las valoraciones.	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 4.</b> frmsiaps</p>	Enviar los datos a las páginas servidoras.
<b>Descripción:</b>	
La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en las páginas clientes para insertar todos los datos referentes a las valoraciones. Sus atributos son los elementos de entrada del formulario ( <i>inputtext</i> , <i>textarea</i> , <i>checkbox</i> , entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 5.</b> SEAM Servlet</p>	Proveer la interacción de la capa de presentación con la de negocio.
<b>Descripción:</b>	
Es el controlador de SEAM que capta las peticiones derivadas de la interacción del usuario después de interactuar con el Servlet de JSF. Enruta las peticiones hacia los controladores que posibilitarán darle respuesta a la petición solicitada. Interviene en la integración de las capas de presentación y negocio.	

Capa de Negocio	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 6.</b> CRealizarValoracion.java</p>	Proveer una respuesta a las peticiones realizadas en la vista.
<b>Descripción:</b>	
La clase CRealizarValoracion.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 las acciones de crear las instancias de una entidad, solicitadas sobre la página cliente correspondiente.	
Capa de Datos	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 7.</b> tb_valoracion_enfermeria.java</p>	Proveer el mapeo con la base de datos.
<b>Descripción:</b>	
La clase tb_valoracion_enfermeria.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de la valoración. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 8.</b> tb_patron.java</p>	Proveer el mapeo con la base de datos.

<b>Descripción:</b>	
<p>La clase <code>tb_patron.java</code> es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos del patrón. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p style="text-align: center; margin: 0;"><b>tb_posible_valor</b> <b>.java</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-valor</li> <li>-eliminado</li> <li>-cid</li> <li>-version</li> <li>-id_variable</li> </ul> </div> <p style="text-align: center; margin-top: 10px;"><b>Figura 9.</b> <code>tb_posible_valor.java</code></p>	<p>Proveer el mapeo con la base de datos.</p>
<b>Descripción:</b>	
<p>La clase <code>tb_posible_valor.java</code> es una clase que se ejecuta del lado del servidor. En ella se almacena los posibles valores que puede tener una variable. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p style="text-align: center; margin: 0;"><b>tb_variable.java</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-codigo</li> <li>-nombre</li> <li>-multivaluado</li> <li>-eliminado</li> <li>-cid</li> <li>-version</li> <li>-id_patron</li> <li>-id_grupo</li> <li>-id_tipo_dato</li> </ul> </div> <p style="text-align: center; margin-top: 10px;"><b>Figura 10.</b> <code>tb_variable.java</code></p>	<p>Proveer el mapeo con la base de datos.</p>
<b>Descripción:</b>	
<p>La clase <code>tb_variable.java</code> es una clase que se ejecuta del lado del servidor. En ella se almacenan datos importantes de una variable. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.</p>	
<b>Nombre:</b>	<b>Propósito:</b>

<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <b>tn_grupo_variable</b>  <b>.java</b>          -id          -codigo          -valor          -eliminado          -cid          -version       </div> <p><b>Figura 11.</b> tn_grupo_variable.java</p>	Proveer el mapeo con la base de datos.
--	--

**Descripción:**

La clase tn\_grupo\_variable.java es una clase que se ejecuta del lado del servidor. En ella se almacenan datos importantes de un grupo. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

<p><b>Nombre:</b></p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <b>tn_resultado_patron.java</b>          -id       </div> <p><b>Figura 12.</b> tn_resultado_patron.java</p>	<p><b>Propósito:</b></p> Proveer el mapeo con la base de datos.
--	---

**Descripción:**

La clase tn\_resultado\_patron.java es una clase que se ejecuta del lado del servidor. En ella se almacena los resultados de los patrones. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

<p><b>Nombre:</b></p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <b>tn_tipo_dato.java</b>          -id          -codigo          -valor          -eliminado          -cid          -version       </div> <p><b>Figura 13.</b> tn_tipo_dato.java</p>	<p><b>Propósito:</b></p> Proveer el mapeo con la base de datos.
---	---

**Descripción:**

La clase tn\_tipo\_dato.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los tipos de datos que puede tener una variable. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

Nombre:	Propósito:
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="margin: 0;"><b>tn_tipo_patron</b> <b>.java</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-codigo</li> <li>-valor</li> <li>-eliminado</li> <li>-cid</li> <li>-version</li> </ul> </div> <p style="text-align: center; margin-top: 10px;"><b>Figura 14.</b> tn_tipo_patron.java</p>	<p>Proveer el mapeo con la base de datos.</p>

**Descripción:**  
 La clase tn\_tipo\_patron.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los tipos de patrones que existen. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

Nombre:	Propósito:
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="margin: 0;"><b>tr_resultado_patron</b> <b>valoracion.java</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-id_resultado_patron</li> <li>-id_valoracion</li> <li>-id_patron</li> <li>-ingreso_hospitalario</li> <li>-comentario</li> <li>-eliminado</li> <li>-cid</li> <li>-version</li> </ul> </div> <p style="text-align: center; margin-top: 10px;"><b>Figura 15.</b> tr_resultado_patron_valoracion.java</p>	<p>Proveer el mapeo con la base de datos.</p>

**Descripción:**  
 La clase tr\_resultado\_patron\_valoracion.java es una clase que se ejecuta del lado del servidor. En ella se almacena datos importantes y la relación de las tablas tb\_valoracion\_enfermeria y tb\_patron. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

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



<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p><b>tr_resultado_variable_valoracion.java</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-valor</li> <li>-eliminado</li> <li>-cid</li> <li>-version</li> <li>-id_variable</li> </ul> </div> <p style="text-align: center;"><b>Figura 16.</b></p> <p style="text-align: center;">tr_resultado_variable_valoracion.java</p>	<p>Proveer el mapeo con la base de datos.</p>
--	---

**Descripción:**

La clase tr\_resultado\_variable\_valoracion.java es una clase que se ejecuta del lado del servidor. En ella se almacena los resultados de la relación entre las tablas tb\_valoracion\_enfermeria y tb\_variable. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

<b>Nombre:</b>	<b>Propósito:</b>
----------------	-------------------

<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p><b>entidad.java</b></p> <ul style="list-style-type: none"> <li>-id</li> </ul> </div> <p style="text-align: center;"><b>Figura 17.</b> entidad.java</p>	<p>Proveer el mapeo con la base de datos.</p>
---	---

**Descripción:**

La clase entidad.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de las entidades en las que se puede implementar el componente web. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

3.4.3 Diagrama de clases del diseño. Gestionar configuración de patrón.

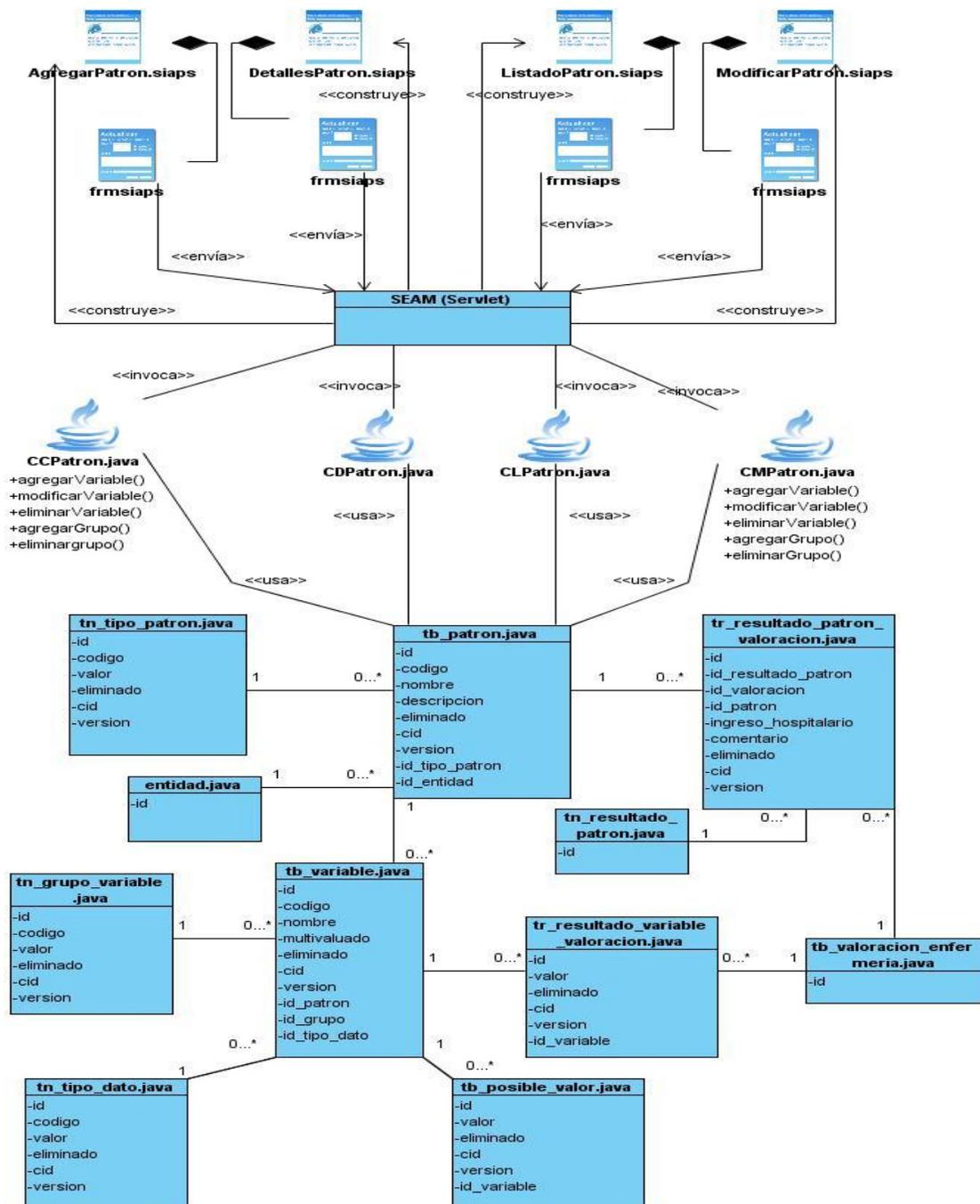


Figura 18. Diagrama de clases del diseño. Gestionar configuración de patrón.

3.4.4 Descripción textual.

Capa de Presentación	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 19.</b> AgregarPatron.shaps</p>	Proveer la interacción con el usuario.
<b>Descripción:</b>	
<p>La clase AgregarPatron.shaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite insertar todos los datos necesarios referentes a los patrones. 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>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 20.</b> frmsiaps</p>	Enviar los datos a las páginas servidoras.
<b>Descripción:</b>	
<p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para insertar todos los datos referentes a los patrones. Sus atributos son los elementos de entrada del formulario (<i>inputtext</i>, <i>textarea</i>, <i>checkbox</i>, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p><b>Figura 21.</b> DetallesPatron.shaps</p>	Proveer la interacción con el usuario.
<b>Descripción:</b>	
<p>La clase DetallesPatron.shaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite mostrar todos los datos referentes a los patrones. Posee un conjunto de validaciones en</p>	

JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.

<b>Nombre:</b>	<b>Propósito:</b>
 <p data-bbox="310 583 545 615"><b>Figura 22.</b> frmsiaps</p>	<p data-bbox="699 474 1187 506">Enviar los datos a las páginas servidoras.</p>
<b>Descripción:</b>	
<p data-bbox="180 680 1435 800">La clase frmsiaps contiene una colección de elementos de salida que están contenidos en la página cliente para mostrar todos los datos referentes a los patrones. Sus atributos son los elementos de salida del formulario (<i>outputtext</i>).</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p data-bbox="241 1012 610 1043"><b>Figura 23.</b> ListadoPatron.siaps</p>	<p data-bbox="699 911 1133 942">Proveer la interacción con el usuario.</p>
<b>Descripción:</b>	
<p data-bbox="180 1108 1435 1184">La clase ListadoPatron.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Muestra todos los datos necesarios referentes a los patrones.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p data-bbox="310 1449 545 1480"><b>Figura 24.</b> frmsiaps</p>	<p data-bbox="699 1339 1187 1371">Enviar los datos a las páginas servidoras.</p>
<b>Descripción:</b>	
<p data-bbox="180 1549 1435 1711">La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para listar todos los datos referentes a los patrones. Sus atributos son los elementos de entrada del formulario (<i>inputtext</i>, <i>textarea</i>, <i>checkbox</i>, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>	
<b>Nombre:</b>	<b>Propósito:</b>



Figura 25. ModificarPatron.siaps

Proveer la interacción con el usuario.

**Descripción:**

La clase ModificarPatron.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite modificar todos los datos necesarios referentes a los patrones. 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:**



Figura 26. frmsiaps

Enviar los datos a las páginas servidoras.

**Descripción:**

La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para modificar todos los datos referentes a los patrones. Sus atributos son los elementos de entrada del formulario (*inputtext*, *textarea*, *checkbox*, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$\_POST.

**Nombre: Nombre:**

**Propósito:**



Figura 27. EliminarPatron.siaps

Proveer la interacción con el usuario.

**Descripción:**

La clase EliminarPatron.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite eliminar todos los datos necesarios referentes a los patrones.

**Nombre:**

**Propósito:**



**Figura 28.** frmsiaps

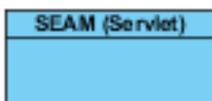
Enviar los datos a las páginas servidoras.

**Descripción:**

La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para eliminar todos los datos referentes a los patrones.

**Nombre:**

**Propósito:**



**Figura 29.** SEAM Servlet

Proveer la interacción de la capa de presentación con la de 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. Enruta las peticiones hacia los controladores que posibilitarán darle respuesta a la petición solicitada. Interviene en la integración de las capas de presentación y negocio.

**Capa de Negocio**

**Nombre:**

**Propósito:**



**Figura 30.** CCPatron.java

Proveer una respuesta a las peticiones realizadas en la vista.

**Descripción:**

La clase CCPatron.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, modificar y eliminar instancias de una entidad solicitadas sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

**Nombre:**

**Propósito:**



**Figura 31.** CDPatron.java

Proveer una respuesta a las peticiones realizadas en la vista.

**Descripción:**

La clase CDPatron.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 ver detalles solicitada sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

**Nombre:**

**Propósito:**



**Figura 32.** CLPatron.java

Proveer una respuesta a las peticiones realizadas en la vista.

**Descripción:**

La clase CLPatron.java es una clase que se ejecuta del lado del servidor. Se encarga de gestionar la acción de listar instancias de una entidad solicitadas sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

**Nombre:**

**Propósito:**



**Figura 33.** CMPatron.java

Proveer una respuesta a las peticiones realizadas en la vista.

**Descripción:**

La clase CMPatron.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 contiene. Se encarga de gestionar la acción de modificar, agregar y eliminar instancias de una entidad solicitada sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

**Capa de Datos**

Nombre:	Propósito:
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <b>tb_patron.java</b>                      -id                      -codigo                      -nombre                      -descripcion                      -eliminado                      -cid                      -version                      -id_tipo_patron                 </div> <p style="text-align: center;"><b>Figura 34.</b> tb_patron.java</p>	<p>Proveer el mapeo con la base de datos.</p>

**Descripción:**  
 La clase tb\_patron.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos del patrón. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

Nombre:	Propósito:
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <b>tn_tipo_patron.java</b>                      -id                      -codigo                      -valor                      -eliminado                      -cid                      -version                 </div> <p style="text-align: center;"><b>Figura 35.</b> tn_tipo_patron.java</p>	<p>Proveer el mapeo con la base de datos.</p>

**Descripción:**  
 La clase tn\_tipo\_patron.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los tipos de patrones que existen. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

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



<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <b>tr_resultado_patron_valoracion.java</b> </div> <div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px;">                 -id                  -id_resultado_patron                  -id_valoracion                  -id_patron                  -ingreso_hospitalario                  -comentario                  -eliminado                  -cid                  -version             </div> <p style="text-align: center;"><b>Figura 36.</b></p> <p style="text-align: center;">tr_resultado_patron_valoracion.java</p>	Proveer el mapeo con la base de datos.
--	--

**Descripción:**

La clase tr\_resultado\_patron\_valoracion.java es una clase que se ejecuta del lado del servidor. En ella se almacena datos importantes y la relación de las tablas tb\_valoracion\_enfermeria y tb\_patron. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <b>tn_resultado_patron.java</b> </div> <div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px;">                 -id             </div> <p style="text-align: center;"><b>Figura 37.</b> tn_resultado_patron.java</p>	Proveer el mapeo con la base de datos.

**Descripción:**

La clase tn\_resultado\_patron.java es una clase que se ejecuta del lado del servidor. En ella se almacena los resultados de los patrones. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

<b>Nombre:</b>	<b>Propósito:</b>
<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <b>tb_valoracion_enfermeria.java</b> </div> <div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px;">                 -id             </div> <p style="text-align: center;"><b>Figura 38.</b> tb_valoracion_enfermeria.java</p>	Proveer el mapeo con la base de datos.

**Descripción:**

La clase `tb_valoracion_enfermeria.java` es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de la valoración. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

Nombre:	Propósito:
<div style="border: 1px solid black; padding: 5px; background-color: #e6f2ff;"> <p><b>tb_variable.java</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-codigo</li> <li>-nombre</li> <li>-multivaluado</li> <li>-eliminado</li> <li>-cid</li> <li>-version</li> <li>-id_patron</li> <li>-id_grupo</li> <li>-id_tipo_dato</li> </ul> </div> <p style="text-align: center;"><b>Figura 39.</b> <code>tb_variable.java</code></p>	<p>Proveer el mapeo con la base de datos.</p>

**Descripción:**  
 La clase `tb_variable.java` es una clase que se ejecuta del lado del servidor. En ella se almacenan datos importantes de una variable. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

Nombre:	Propósito:
<div style="border: 1px solid black; padding: 5px; background-color: #e6f2ff;"> <p><b>tn_grupo_variable.java</b></p> <ul style="list-style-type: none"> <li>-id</li> <li>-codigo</li> <li>-valor</li> <li>-eliminado</li> <li>-cid</li> <li>-version</li> </ul> </div> <p style="text-align: center;"><b>Figura 40.</b> <code>tn_grupo_variable.java</code></p>	<p>Proveer el mapeo con la base de datos.</p>

**Descripción:**  
 La clase `tn_grupo_variable.java` es una clase que se ejecuta del lado del servidor. En ella se almacenan datos importantes de un grupo. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

Nombre:	Propósito:

tn_tipo_dato.java
-id
-codigo
-valor
-eliminado
-cid
-version

Figura 41. tn\_tipo\_dato.java

Proveer el mapeo con la base de datos.

**Descripción:**

La clase tn\_tipo\_dato.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los tipos de datos que puede tener una variable. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

**Nombre:**

**Propósito:**

tb_posible_valor .java
-id
-valor
-eliminado
-cid
-version
-id_variable

Figura 42. tb\_posible\_valor.java

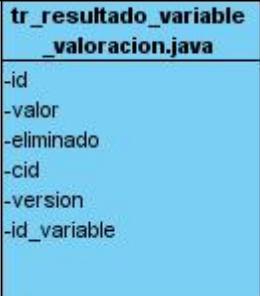
Proveer el mapeo con la base de datos.

**Descripción:**

La clase tb\_posible\_valor.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los posibles valores que puede tener una variable. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.

**Nombre:**

**Propósito:**

 <p><b>Figura 43.</b> tr_resultado_variable_valoracion.java</p>	<p>Proveer el mapeo con la base de datos.</p>
--	---

<p><b>Descripción:</b></p>	
<p>La clase tr_resultado_variable_valoracion.java es una clase que se ejecuta del lado del servidor. En ella se almacena los resultados de la relación entre la tabla tb_valoración_enfermeria y tb_variable. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.</p>	
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
 <p><b>Figura 44.</b> entidad.java</p>	<p>Proveer el mapeo con la base de datos.</p>
<p><b>Descripción:</b></p>	
<p>La clase entidad.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de las entidades en las que se puede implementar el componente web. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.</p>	

El patrón arquitectónico MVC permite una distribución óptima de funcionalidades para el desarrollo del componente web. Los patrones de diseños GRASP permiten que las clases involucradas en el desarrollo de este componente web realicen sus funciones específicas. Los diagramas de clases del diseño permiten describir la estructura del componente web, sus clases y atributos, definiendo el diseño conceptual de la información y los componentes que se encargarán del funcionamiento.

---

# Capítulo 4. Implementación del componente web para la valoración de enfermería por patrones.

El presente capítulo constituye la secuencia lógica del Diseño. A través del mismo se implementarán las clases encontradas durante este en términos de componentes, que indicará cómo la solución implementada estará distribuida físicamente. Además se definen los estándares de codificación y el tratamiento de errores.

### 4.1 Propuesta de integración con otros módulos

El Componente web para la valoración de enfermería por patrones del módulo Enfermería perteneciente al sistema alasSIAPS, se relaciona con:

#### Módulo Configuración

El módulo Configuración le permite al módulo de Enfermería obtener la información relacionada con los departamentos y centros laborales, además de proporcionar los datos del personal de salud y las entidades a las que pertenece.

### 4.2 Implementación

La implementación comienza con los resultados de la fase de diseño y se procede a implementar las clases y componentes (ficheros de código fuente) encontrados durante el diseño. Su propósito consiste en desarrollar la arquitectura y el sistema en su conjunto.

### 4.3 Diagrama de despliegue

“Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo: dispositivos, procesadores y memoria”. (17)

El diagrama de despliegue del componente web para la valoración de enfermería por patrones quedaría de la siguiente manera:

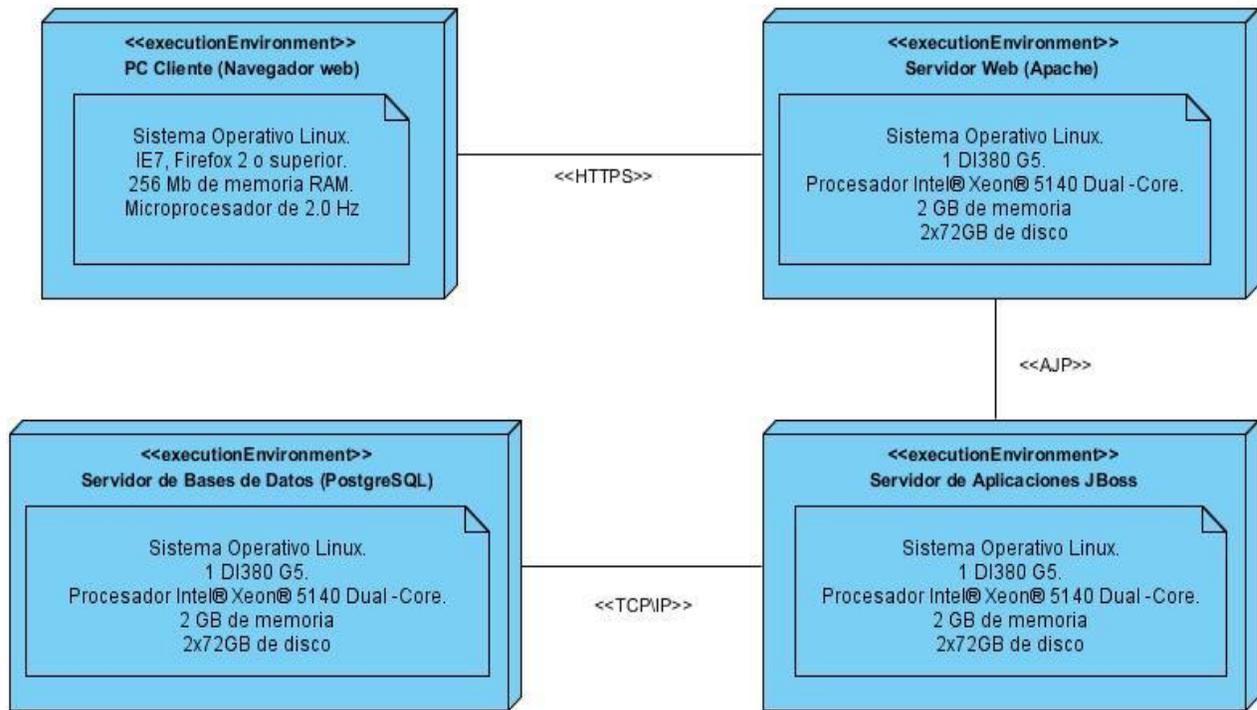


Figura 44. Diagrama de despliegue.

### 4.3.1 Descripción de los elementos del Diagrama de despliegue

**Estaciones clientes:** serán las estaciones de los usuarios, las cuales servirán para acceder al sistema web mediante un navegador.

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

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

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

### 4.4 Estándares de codificación y tratamiento de errores

Con el objetivo de lograr uniformidad en todo el alasSIAPS, se definen una serie de estándares para la codificación. Además se dará tratamiento a las excepciones lanzadas en tiempo de ejecución y se notificará al usuario sobre estas a través de mensajes, de manera uniforme y en correspondencia al resto del sistema.

#### 4.4.1 Estándares de codificación

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

- ✓ Casi ningún software lo mantiene toda su vida el autor original.
- ✓ El 80% del coste del código de un programa va a su mantenimiento.
- ✓ Las convenciones de código mejoran la lectura del software y permiten entender código nuevo más rápido y más a fondo.
- ✓ Si se distribuye el código fuente como un producto, necesita asegurarse 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 el derecho de autor.

Las líneas en blanco mejoran la facilidad de lectura, se separan 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 ocho espacios.

Evitar las líneas de más de ochenta 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 guión bajo “\_” 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 debe ser un mnemónico, designado para indicar a un observador casual su función.

Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas y separadas las palabras con un guión bajo (“\_”). (Las constantes ANSI (Instituto Nacional Estadounidense de Estándares) 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.

### 4.4.2 Tratamiento de excepciones

En el Componente web para la valoración de enfermería por patrones 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 ser tratadas sin interrumpir el sistema. También se emplean un conjunto de tipos de excepciones predefinidas por los frameworks 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, estos permiten establecer tipos de datos y formatos que controlan el envío de los activos al servidor.

Además, el framework JBoss 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.

El Diagrama de despliegue describe los nodos que serán usados para la implantación del componente web y los estándares de codificación permiten lograr un componente web legible y uniforme. Las posibles excepciones que el componente web puede presentar son tratadas para que no interrumpan su función. Al concluir la implementación se obtuvo un sistema que responde a las funcionalidades previstas para la primera versión y se garantiza su correcto funcionamiento.

## Conclusiones

El desarrollo del Componente web para la valoración de enfermería por patrones del Sistema Integral para la Atención Primaria contempla las tareas previstas para la investigación y aporta elementos que permiten arribar a las siguientes conclusiones:

- La valoración es una fase vital en el Proceso de Atención de Enfermería y la calidad de este depende en gran medida de la información que esta aporta.
- El análisis de los sistemas informáticos relacionados con la valoración de enfermería no arrojó la existencia de uno que satisfaga las necesidades de la misma en la actualidad, la cual muestra tendencias hacia el uso de los patrones como modelo para guiar esta importante fase del PAE.
- Las herramientas y tecnologías definidas por el Departamento de Atención Primaria de Salud para el desarrollo de aplicaciones web brindan un ambiente robusto y seguro para el desarrollo de este tipo de aplicaciones, aportándole escalabilidad y usabilidad.
- El componente web desarrollado brinda al alasSIAPS funcionalidades que permiten realizar la valoración por patrones de una manera flexible, escalable y configurable; así como acceder a la misma con el objetivo de ser consultada por los profesionales autorizados.

## Recomendaciones

La solución propuesta puede ser enriquecida con algunas funcionalidades que pueden aportar eficiencia.

Por esta razón se recomienda:

- Aplicar técnicas de inteligencia artificial a la información recogida y organizada en la fase de valoración con el uso del componente web desarrollado; con el objetivo de brindar posibles resultados de los patrones valorados y valiosa información que apoye el proceso de toma de decisiones y el estudio estadístico.
- Implementar una funcionalidad que permita la creación en formato PDF de una plantilla, basada en un subconjunto (determinado por el usuario) de los patrones configurados; con el objetivo de ser impreso y permita recoger los datos en los lugares distantes de las estaciones de trabajo en las que se encuentre desplegado el sistema y no se cuente con los recursos necesarios que permitan llevar el sistema a través de equipos móviles hasta donde se encuentre el paciente.

## Referencias Bibliográficas

1. **Legido, O. Ruiz.** Sociedad Andaluza de Medicina Familiar y Comunitaria. *El nuevo papel de la enfermera*. [En línea] 02 de 2008. [Citado el: 22 de 03 de 2013.] <http://www.samfyc.es/Revista/PDF/v8n2/02.pdf>.
2. **OMS.** Organización Mundial de la Salud. *Enfermería*. [En línea] 2013. [Citado el: 20 de 01 de 2013.] <http://www.who.int/topics/nursing/es/>.
3. **SESCAM.** Servicio de Salud de Castilla-La Mancha. *P.A.E: Proceso de Atención de Enfermería*. [En línea] 2009. [Citado el: 05 de 02 de 2013.] [http://sescam.jccm.es/web1/gaptalavera/prof\\_enfermeria/PAE.pdf](http://sescam.jccm.es/web1/gaptalavera/prof_enfermeria/PAE.pdf).
4. **Vergel, M.Sc. Lázaro Luis Hernández y Velarde, M.Sc. Lloyne Concepción.** Clínica Central Cira García. *Proceso de enfermería*. [En línea] [Citado el: 20 de 01 de 2013.] <http://www.enfermeria.sld.cu/enfermeriacirag/proceso.html>.
5. **Ponce, Dra. Catalina Rodríguez.** Portal del Hospital Regional Universitario Carlos Haya de Málaga. *LA VALORACIÓN ENFERMERA. GUÍA PARA LA VALORACIÓN DEL PACIENTE CON TRASTORNOS NEFROLÓGICOS*. [En línea] [Citado el: 25 de 11 de 2012.] <http://www.carloshaya.net/biblioteca/contenidos/docs/nefrologia/predialisis/catalinarodriguez.PDF>.
6. **Reyes, Isidora.** Gerencia, Innovación y Salud. *Virginia Henderson: Definición de Enfermería*. [En línea] 18 de 11 de 2012. [Citado el: 23 de 01 de 2013.] <http://quemundo.bligoo.com/ve/virginia-henderson-definicion-de-enfermeria>.
7. **Fundación para el Desarrollo de la Enfermería.** Observatorio de Metodología de Enfermera. *Patrones funcionales de Marjory Gordon*. [En línea] [Citado el: 12 de 01 de 2013.] [http://www.ome.es/04\\_01\\_desa.cfm?id=391](http://www.ome.es/04_01_desa.cfm?id=391).
8. **OME.** Observatorio Metodología Enfermera. *CONCEPTO DE "RESPUESTAS HUMANAS" EN RELACIÓN A LA SALUD*. [En línea] [Citado el: 03 de 04 de 2013.] [http://www.ome.es/02\\_02\\_01\\_desa.cfm?id=101](http://www.ome.es/02_02_01_desa.cfm?id=101).
9. **Alfaro-LeFevre, Rosalinda.** *Applying Nursing Process. A step-by-step*. Florida, EE.UU : Lippincott-Raven Publishers, Filadelfia, 1998. 84-07-00197-X.

10. **Cordero, María José Aguilar.** *Tratado de Enfermería Infantil: cuidados pediátricos.* España : Elsevier, España, S.A, 2003. 84-8174-558-8.
11. **Sáez, Ángeles y Paredes, Antonio.** Scribd. *VALORACION PATRONES FUNCIONALES M.GORDON.* [En línea] [Citado el: 06 de 04 de 2013.] <http://es.scribd.com/doc/49174696/VALORACION-PATRONES-FUNCIONALES-M-GORDON>.
12. **Jiménez, A.García, Albá, C.Botana y Diéguez, I. Alvarez.** *Medicina Interna de Galicia. VALORACION DEL SISTEMA INFORMATICO APLICATIVO "GACELA" EN UNA UNIDAD ESPECIAL HOSPITALARIA.* [En línea] 05 de 04 de 2003. [Citado el: 01 de 03 de 2013.] <http://www.meiga.info/meiga.asp?cap=1&mat0=8&mat1=385&mat=387&id=792>.
13. **LogicPAE Software para sanidad.** *Valoración Integral y Diagnósticos.* [En línea] 2009-2011. [Citado el: 10 de 04 de 2013.] [http://www.logicpae.com/web/index.php?option=com\\_hikashop&ctrl=product&task=show&cid=4&name=valoracion-integral-y-diagnosticos&Itemid=28&category\\_pathway=0](http://www.logicpae.com/web/index.php?option=com_hikashop&ctrl=product&task=show&cid=4&name=valoracion-integral-y-diagnosticos&Itemid=28&category_pathway=0)
14. **MINISTERIO DE SALUD PÚBLICA DIRECCIÓN NACIONAL DE REGISTROS MÉDICOS Y ESTADÍSTICAS DE SALUD.** *Anuario Estadístico de Salud 2012.* La Habana, : Comité Editorial, abril 2013. 1561-4425
15. **Glosario de Informática e Internet.** *Modelo Cliente-Servidor.* [En línea] [Citado el: 10 de 03 de 2013.] <http://www.internetglosario.com/370/ModeloClienteServidor.html>.
16. **Vinent, Yaneisi Betancourt y Acosta, Yasniel Pérez.** *Componente web para Programas de enfermedades transmisibles del Sistema Integral para la Atención Primaria de Salud.* La Habana : s.n., 2012.
17. **Huallpara, Hugo Michael Marca y Limachi, Nancy Susana Quisbert.** *ANALISIS Y DISEÑO DE SISTEMAS II. Diagrama de Despliegue.* [En línea] [Citado el: 28 de 03 de 2013.] [http://www.google.com/cu/url?sa=t&rct=j&q=diagrama+de+despliegue&source=web&cd=19&ved=0CGkQFjAS&url=http%3A%2F%2Fvirtual.usalesiana.edu.bo%2Fweb%2Fpractica%2Farchiv%2Fdespliegue.doc&ei=JwiRUZitBeSE0QHWk4HQBg&usq=AFQjCNGu7s\\_\\_JDdYCEwEmbkYnxa2XI2Q2Q&cad=rja](http://www.google.com/cu/url?sa=t&rct=j&q=diagrama+de+despliegue&source=web&cd=19&ved=0CGkQFjAS&url=http%3A%2F%2Fvirtual.usalesiana.edu.bo%2Fweb%2Fpractica%2Farchiv%2Fdespliegue.doc&ei=JwiRUZitBeSE0QHWk4HQBg&usq=AFQjCNGu7s__JDdYCEwEmbkYnxa2XI2Q2Q&cad=rja)

## Bibliografía

**Alfaro-LeFevre, Rosalinda.** *Applying Nursing Process. A step-by-step.* Florida, EE.UU: Lippincott-Raven Publishers, Filadelfia, 1998. 84-07-00197-X

**ArPUG (Grupo de Usuarios PostgreSQL de Argentina).** *pgAdmin 3.* [En línea] [Citado el: 25 de 02 de 2013.] <http://www.arpug.com.ar/trac/wiki/PgAdmin>

**Autores, Colectivo de.** Biblioteca Virtual en Salud. *La Enfermería y el Sistema de Salud cubano.* [En línea] [Citado el: 23 de Noviembre de 2012.] <http://bvs.sld.cu/revistas/infnd/n1412/infnd011412.htm>

**Camiño, MSc. Reinaldo Rodríguez, Álvarez, MSc. María del Carmen Hinojosa y Larín, Lic. Silvia Serra.** Revista de Ciencias Médicas. *LAS BIBLIOTECAS VIRTUALES DE SALUD EN LA INFORMATIZACIÓN DE LA SOCIEDAD CUBANA.* [En línea] 2007. [Citado el: 22 de Noviembre de 2012.] [http://www.cpicmha.sld.cu/hab/vol13\\_2\\_07/hab17207.htm](http://www.cpicmha.sld.cu/hab/vol13_2_07/hab17207.htm)

**Colectivo de autores.** *Diplomado en Enfermería/ATS del Consorcio Hospital General Universitario de Valencia (CHGUV).* España: Mad, S.L, 2004. 84-665-4066-0.

**Colectivo de autores.** *Diplomado en Enfermería/ATS del Consorcio Hospital General Universitario de Valencia (CHGUV).* España: Mad, S.L, 2004. 84-665-4066-0.

**Cordero, María José Aguilar.** *Tratado de Enfermería Infantil: cuidados pediátricos.* España: Elsevier, España, S.A, 2003. 84-8174-558-8.

**Cornejo, José Enrique González.** DocIRs. *¿Qué es UML?* [En línea] [Citado el: 21 de 02 de 2013.] <http://www.docirs.cl/uml.htm>

**FUDEN (Fundación para el Desarrollo de la Enfermería).** Observatorio de Metodología de Enfermera. *Valoración de enfermería.* [En línea] [Citado el: 11 de 12 de 2012.] [http://www.ome.es/media/docs/G1-Doc.%20Valoraci%C3%B3n\\_enfermera.pdf](http://www.ome.es/media/docs/G1-Doc.%20Valoraci%C3%B3n_enfermera.pdf)

**Fundamentos Metodológicos de Enfermería.** *ANEXO II - MODELO-GUIA PARA LA VALORACIÓN POR PATRONES FUNCIONALES DE SALUD (M. Gordon).* [En línea] 2012. [https://www5.uva.es/guia\\_docente/uploads/2012/475/46175/1/Documento6.pdf](https://www5.uva.es/guia_docente/uploads/2012/475/46175/1/Documento6.pdf)

**Gobierno de Aragon.** Departamento de Sanidad, Bienestar Social y Familia. [En línea] [www.aragon.es/DepartamentosOrganismosPublicos/Departamentos/SanidadBienestarSocialFamilia/Areas Tematicas/SanidadProfesionales/SaludPublica/ci.01\\_OMI\\_AP\\_mixto.detalleDepartamento?channelSelecte d=7ce514d66d9cb210VgnVCM100000450a15acRCRD](http://www.aragon.es/DepartamentosOrganismosPublicos/Departamentos/SanidadBienestarSocialFamilia/AreasTematicas/SanidadProfesionales/SaludPublica/ci.01_OMI_AP_mixto.detalleDepartamento?channelSelecte d=7ce514d66d9cb210VgnVCM100000450a15acRCRD)

**Haro, José María Galindo y Riba, Josep María Camps.** Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE. *Frameworks*. [En línea] 2008. [Citado el: 10 de 03 de 2013.] <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765tfc.pdf>

**Izuel, Directora Chita López, Camps, Coordinadora Marta Riu y Bscheid, Coordinadora Maite Forner.** *Cuidados Enfermeros*. Barcelona, España: MASSON, S.A, 2005. 84-458-1414-1.

**Izuel, Directora Chita López, Camps, Coordinadora Marta Riu y Bscheid, Coordinadora Maite Forner.** *Cuidados Enfermeros*. Barcelona, España: MASSON, S.A, 2005. 84-458-1414-1.

**Junta de Andalucía.** Marco de Desarrollo de la Junta de Andalucía. *JavaServer Faces (JSF)*. [En línea] [Citado el: 23 de 02 de 2013.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/101>

**Junta de Andalucía.** Marco de Desarrollo de la Junta de Andalucía. *RichFaces*. [En línea] [Citado el: 23 de 02 de 2013.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/134>

**Junta de Andalucía.** Marco de Desarrollo de la Junta de Andalucía. *Hibernate*. [En línea] [Citado el: 23 de 02 de 2013.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/97>

**Legido, O. Ruiz.** Sociedad Andaluza de Medicina Familiar y Comunitaria. *El nuevo papel de la enfermera*. [En línea] 02 de 2008. [Citado el: 22 de 03 de 2013.] <http://www.samfyc.es/Revista/PDF/v8n2/02.pdf>.

**LogicPAE Software para sanidad.** *Valoración Integral y Diagnósticos*. [En línea] 2009-2011. [Citado el: 10 de 04 de 2013.] [http://www.logicpae.com/web/index.php?option=com\\_hikashop&ctrl=product&task=show&cid=4&name=valoracion-integral-y-diagnosticos&Itemid=28&category\\_pathway=0](http://www.logicpae.com/web/index.php?option=com_hikashop&ctrl=product&task=show&cid=4&name=valoracion-integral-y-diagnosticos&Itemid=28&category_pathway=0)

**MINISTERIO DE SALUD PÚBLICA DIRECCIÓN NACIONAL DE REGISTROS MÉDICOS Y ESTADÍSTICAS DE SALUD.** *Anuario Estadístico de Salud 2012*. La Habana, : Comité Editorial, abril 2013. 1561-4425

**Montoya, Verónica.** Implementación de Software. *Fases De Implementación Del Software*. [En línea] 12 de 11 de 2012. [Citado el: 28 de 03 de 2013.] <http://veronica1235.blogspot.com/2012/11/fases-de-implementacion-del-software.html>

**Observatorio de Metodología de Enfermera.** *Patrones funcionales de Marjory Gordon*. [En línea] [Citado el: 12 de 1 de 2013.] [http://www.ome.es/04\\_01\\_desa.cfm?id=391](http://www.ome.es/04_01_desa.cfm?id=391)

**OMS.** Organización Mundial de la Salud. *Enfermería*. [En línea] 2013. [Citado el: 20 de 01 de 2013.] <http://www.who.int/topics/nursing/es/>.

**Palenzuela, Lic. Anabel Baños Benítez y Lic. María Emilia Victores.** Informática en Salud 2007. *UN ESPACIO VIRTUAL PARA ELEVAR LA SALUD MENTAL Y LA CALIDAD DE VIDA DE NUESTROS USUARIOS*. [En línea] 2007. [Citado el: 12 de 1 de 2013.] <http://www.informatica2007.sld.cu/Members/Anabel/un-espacio-virtual-para-elevar-la-salud-mental-y-la-calidad-de-vida-de-nuestros-usuarios/2006-10-26.8429211852>

**Ponce, Dra. Catalina Rodríguez.** Portal del Hospital Regional Universitario Carlos Haya de Málaga. *LA VALORACIÓN ENFERMERA. GUÍA PARA LA VALORACIÓN DEL PACIENTE CON TRASTORNOS NEFROLÓGICOS*. [En línea] [Citado el: 25 de 11 de 2012.]

**Pública, Ministerio de Salud.** DIRECCIÓN NACIONAL DE REGISTROS MÉDICOS Y ESTADÍSTICAS DE SALUD. *Anuario Estadístico de Salud 2012*. [En línea] 4 de 2013. [Citado el: 29 de 4 de 2013.] [http://files.sld.cu/dne/files/2013/04/anuario\\_2012.pdf](http://files.sld.cu/dne/files/2013/04/anuario_2012.pdf)

**Pública, Ministerio de Salud.** DIRECCIÓN NACIONAL DE REGISTROS MÉDICOS Y ESTADÍSTICAS DE SALUD. *Anuario Estadístico de Salud 2012*. [En línea] 4 de 2013. [Citado el: 29 de 4 de 2013.] [http://files.sld.cu/dne/files/2013/04/anuario\\_2012.pdf](http://files.sld.cu/dne/files/2013/04/anuario_2012.pdf)

**rafaelma.** PostgreSQL-es (Portal en español sobre PostgreSQL). *PostgreSQL*. [En línea] 02 de 10 de 2010. [Citado el: 21 de 02 de 2013.] [www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)

**Ramos, Juan Alonso.** Adictos al Trabajo. Introducción a Ajax4jsf. [En línea] [Citado el: 20 de 02 de 2013.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>

**Scribd.** *JBoss*. [En línea] 24 de 08 de 2009. [Citado el: 21 de 02 de 2013.] <http://es.scribd.com/doc/19026497/JBOSS>

**SESCAM (Servicio de Salud de Castilla-La Mancha).** *P.A.E: Proceso de Atención de Enfermería.* [En línea] 2009. [Citado el: 05 de 02 de 2013.] [http://sescam.jccm.es/web1/gaptalavera/prof\\_enfermeria/PAE.pdf](http://sescam.jccm.es/web1/gaptalavera/prof_enfermeria/PAE.pdf)

**SESCAM.** Servicio de Salud de Castilla-La Mancha. *P.A.E: Proceso de Atención de Enfermería.* [En línea] 2009. [Citado el: 05 de 02 de 2013.] [http://sescam.jccm.es/web1/gaptalavera/prof\\_enfermeria/PAE.pdf](http://sescam.jccm.es/web1/gaptalavera/prof_enfermeria/PAE.pdf).

**The Eclipse Foundation.** *About the Eclipse Foundation.* [En línea] [Citado el: 25 de 02 de 2013.] <http://www.eclipse.org/org/>

**Theo, Otto.** Scribd. *Facelets y JSF.* [En línea] 17 de 12 de 2008. [Citado el: 21 de 02 de 2013.] <http://es.scribd.com/doc/44720276/Facelets-y-JSF>

**Vergel, MSc. Lázaro Luis Hernández y Velarde, MSc. Lloyne Concepción.** Clínica Central Cira García. *Proceso de enfermería.* [En línea] [Citado el: 20 de 1 de 2013.] <http://www.enfermeria.sld.cu/enfermeriacirag/proceso.html>

## Glosario de términos

**API (Application Programming Interface):** es considerada como un conjunto de funciones y procedimientos que permiten la comunicación entre componentes de software.

**Interoperabilidad:** condición necesaria para que los usuarios (humanos o mecánicos) tengan un acceso completo a la información disponible.

**JPA (Java Persistence API):** es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones donde se usa la plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE).

**Juicio clínico:** determinación de un diagnóstico según la experiencia de profesionales aplicado a la información que brinda el paciente en la observación de su estado de salud.

**Lenguaje estándar:** lenguaje que posibilita la comunicación de forma organizada y común entre dos sujetos.

**Modelo enfermero:** es un modelo válido para diferentes ámbitos clínicos de trabajo.

**Respuestas fisiopatológicas:** son respuestas de las personas ante enfermedades, procesos físicos y químicos.

**Software inteligente:** es una aplicación creada para prevenir y actuar por si solo en determinadas situaciones.