

***Universidad de las Ciencias Informáticas***

***Facultad 7***



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

***Título: Desarrollo del Módulo de Nutrición del Sistema de  
Evaluación del Neurodesarrollo en Niños***

***Autores: Odelkys Guzmán Moreno***

*Marcel Alí Alegre Fabre*

***Tutores: Malena Guzmán Pérez***

*Solainy Fajardo Araujo*

***Asesores: Dr. Pedro Mestre Villavicencio***

*Dr. Lázaro Alfonso Novo*

*Téc. Mabel Vera Díaz*

*Ciudad de La Habana, Junio 2010*

*“Año 52 de la Revolución”*

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Odelkys Guzmán Moreno**

Firma de la Autora

---

**Marcel Alí Alegre Fabre**

Firma del Autor

---

**Ing. Malena Guzmán Pérez**

Firma de la Tutora

---

**Ing. Solainy Fajardo Araujo**

Firma de la Tutora

### **DATOS DE CONTACTO**

**Ing. Malena Guzmán Pérez.** Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2008. Instructora recién graduada en adiestramiento. Durante su trabajo como profesora ha impartido la asignatura de Historia de la informática y cursos pertenecientes a práctica profesional. Se ha desempeñado como jefa de la asignatura Historia de la Informática. Actualmente imparte la asignatura Historia de la Informática, de la cual continúa siendo Jefa de asignatura.

En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Medicina del Centro Especializado en Soluciones de Informática Médica y específicamente trabaja en el desarrollo del proyecto alas SENDN donde se desempeña como Líder de Proyecto. Además se ha desempeñado como responsable de PP 2–5 y actualmente es responsable de las tesis del mismo departamento.

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

**Ing. Solainy Fajardo Araujo.** Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2009. Instructora recién graduada en adiestramiento. Durante su trabajo como profesora ha impartido la asignatura de Matemática Discreta I.

En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Medicina del Centro Especializado en Soluciones de Informática Médica y específicamente trabaja en el desarrollo del proyecto alas SENDN donde se desempeña como Analista principal.

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

## RESUMEN

En Cuba, el Hospital Pediátrico Universitario William Soler lleva a cabo un Programa de Atención y Evaluación del Neurodesarrollo en Niños con el objetivo de proporcionarle al paciente enfermo una calidad de vida óptima. En el presente trabajo se desarrolla una aplicación web para facilitar la gestión de la información generada en los procesos de atención y evaluación del Neurodesarrollo en la especialidad de Nutrición, del grupo de Atención Temprana y Evaluación del Neurodesarrollo del centro mencionado.

Para el desarrollo de la aplicación se emplean como herramientas: PostgreSQL como Sistema Gestor de Base de Datos, el Lenguaje de Programación Java, la Metodología de Desarrollo de Software: el Proceso Unificado de Desarrollo (RUP). Se hace uso del Lenguaje Unificado de Modelado (UML) para especificar, construir y documentar el sistema y Visual Paradigm for UML 2 para la creación de los artefactos que se generan durante el ciclo de vida del software. BPMN es la notación para el modelado de procesos del negocio y el Framework JBoss Seam que utiliza el patrón Modelo-Vista-Controlador (MVC).

El desarrollo del módulo de Nutrición del Sistema de Evaluación del Neurodesarrollo en Niños, permitirá a los profesionales de esta especialidad, atender una mayor cantidad de pacientes en el día con la calidad requerida, demostrándose así la eficiencia de la aplicación en cuestión.

**Palabras Claves:** Neurodesarrollo, Framework, Software.

**Tabla de Contenido**

**RESUMEN**.....III

**INTRODUCCIÓN**.....1

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**.....6

    1.1. Conceptos Básicos .....6

    1.2. Antecedentes de la Atención Temprana .....6

    1.3. Sistemas Automatizados Existentes .....7

    1.4. Metodología de Desarrollo .....9

    1.5. Tecnologías .....11

        1.5.1 Patrones de arquitectura y diseño.....11

        1.5.2. Tecnologías horizontales .....17

        1.5.3 Lenguajes .....18

    1.6. Herramientas a utilizar .....21

**CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**.....35

    2.1 Flujo actual de los procesos involucrados en el campo de acción .....35

    2.2. Descripción de los Procesos de Negocio.....35

    2.3. Análisis crítico de ejecución de los procesos actuales .....37

    2.4. Objeto de Automatización.....37

    2.5. Modelo del negocio .....37

        2.5.1. Identificación de roles del entorno del negocio .....38

    2.6. Especificación de los Requisitos de Software .....45

        2.6.1 Requerimientos Funcionales .....45

        2.6.2 Requisitos No Funcionales .....47

2.7. Modelo de casos de uso del sistema .....	52
2.7.1 Definición de actores del sistema .....	53
2.7.2. Diagrama de Casos de Uso del Sistema .....	53
2.7.3 Listado de Casos de Uso.....	54
2.7.4 Organización de los Elementos: Casos de Uso en Paquetes. ....	59
<b>CAPÍTULO 3: DISEÑO DEL SISTEMA .....</b>	<b>64</b>
3.1 Arquitectura .....	64
3.2 Modelo de Diseño.....	64
3.3 Diagrama de Interacción (Secuencia) .....	65
<b>CAPÍTULO 4: IMPLEMENTACIÓN. ....</b>	<b>67</b>
4.1 Modelo de datos .....	67
4.2 Modelo de implementación .....	67
4.2.1 Diagrama de componentes.....	68
4.2.2 Diagrama de despliegue.....	70
4.3 Seguridad .....	71
4.4 Estándares utilizados.....	72
4.4.1 Estándares de Diseño. ....	72
4.4.2 Estándares de codificación.....	77
<b>CONCLUSIONES .....</b>	<b>80</b>
<b>RECOMENDACIONES.....</b>	<b>81</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>82</b>
<b>BIBLIOGRAFÍA.....</b>	<b>84</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>86</b>

## **Introducción.**

En Cuba, antes del Triunfo de la Revolución, la atención médica y hospitalaria se caracterizaba por el predominio de servicios de carácter privado y mutualista. Esta modalidad en los servicios de salud impedía el acceso a las personas de más bajos ingresos, las que contaban como única opción con las Casas de Socorro que atendían principalmente los casos de urgencia. A partir del triunfo revolucionario se comenzó a trabajar en este sector, con el objetivo de llevar la acción del trabajador de la salud a los lugares más apartados del país.

Con el principio de que la salud es derecho del pueblo y responsabilidad del Estado, se garantiza la accesibilidad de la población a todos los servicios de esta rama. La atención médica se hizo gratuita y cubrió a toda la población, lo que trajo consigo una mejora notable de la situación del estado de salud del pueblo y un fortalecimiento del sistema de salud, su nivel primario de atención y la medicina familiar.

Todo este proyecto social que ha garantizado la equidad, accesibilidad y seguridad médica de todos los ciudadanos desde el Triunfo de la Revolución ha sido posible gracias a la creación del Sistema Nacional de Salud (SNS). El cual es regido por el Ministerio de Salud Pública (MINSAP), organismo encargado de dirigir, ejecutar y controlar la aplicación de la política de Estado y del Gobierno en cuanto a Salud Pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica.

Con el objetivo de incorporar la informática y los avances en las nuevas Tecnologías de la Información y las Comunicaciones (TIC) a los procesos vinculados a la salud, el Ministerio de Salud Pública comenzó con el proceso de la informatización de la misma ya hace algunos años.

Como apoyo a la informatización de la salud, la Universidad de las Ciencias Informáticas (UCI) y dentro de ella la Facultad 7 ha puesto en marcha una serie de proyectos productivos apoyando el desarrollo de software, centrándose en productos para este sector.

Como parte de los productos que se desarrollan se encuentra el Sistema de Evaluación del Neurodesarrollo en Niños (SENDN) vinculado al Hospital Pediátrico Universitario "William Soler", donde se desarrolla un programa médico de atención temprana y evaluación del neurodesarrollo. El objetivo fundamental del programa es brindar una atención especializada al paciente grave, siendo éstos niños de

cero a cinco años de edad egresados de las Unidades de Cuidados Intensivos Polivalente y Neonatal (UCIP/UCIN).

Estas evaluaciones las realiza un equipo interdisciplinario, compuesto por los especialistas en: neurología, psicología, fisiatría, neurofisiología, logopedia, y nutrición. Pues cada una de estas especialidades, se debe tener en cuenta para la atención y evaluación del neurodesarrollo en niños.

Los especialistas en Nutrición realizan una serie de evaluaciones, que les permite determinar el estado nutricional de los niños, éste resulta un parámetro imprescindible para su valoración integral y constituye además un indicador de la calidad de vida en cualquier etapa del ciclo vital.

El estado nutricional se puede explorar mediante evaluaciones antropométricas. Se realizan una serie de mediciones técnicas sistematizadas que expresan, cuantitativamente, las dimensiones del cuerpo humano. Además, se hacen evaluaciones clínicas que permiten saber el comportamiento del paciente en ese momento y evaluaciones bioquímicas para saber las condiciones de su organismo para adaptarse a una dieta.

Luego de realizadas cada una de estas evaluaciones, se elabora un resumen que es recogido en el expediente del paciente donde se reflejan los resultados encontrados, plasmando la impresión diagnóstica de los médicos, la cual se discute entre todos los especialistas para poder determinar las afectaciones en el neurodesarrollo del paciente y prescribir un tratamiento. Cada una de estas evaluaciones requiere de la recogida de un gran volumen de información, siendo necesario en algunas de ellas consultar tablas matemáticas demasiado grandes y complejas por lo que puede cometerse algún error humano al registrar la información.

Además se dificulta la comparación de diagnósticos emitidos en diferentes consultas debido a que en el expediente del paciente se van archivando los resultados de las consultas de cada una de las especialidades, lo cual a medida que van aumentando las consultas constituye un gran cúmulo de información. Esto provoca un lento y engorroso procesamiento de la información, existiendo además la posibilidad de cometer errores, los cuales pueden afectar el diagnóstico del paciente y por ende su calidad de vida.



Por lo antes planteado se identifica como **problema a resolver**: ¿Cómo viabilizar el proceso de gestión de la información relacionada con la atención y evaluación del neurodesarrollo de los niños en la especialidad de Nutrición?

Este problema se enmarca en el **objeto de estudio**: El proceso de gestión de la información relacionada con la atención y evaluación del neurodesarrollo de los niños egresados de terapia intensiva. El objeto delimita el **campo de acción**: El proceso de gestión de la información relacionada con la atención y evaluación del neurodesarrollo de los niños egresados de terapia intensiva en la especialidad de Nutrición del Hospital Pediátrico William Soler de Cuba.

Para la solución del problema se plantea como **objetivo general**: Desarrollar un sistema informático que permita viabilizar la gestión de la información que se genera en los procesos relacionados con la atención y evaluación del neurodesarrollo en la especialidad de nutrición del grupo de Atención Temprana y Evaluación del Neurodesarrollo del Hospital Pediátrico Universitario William Soler de Cuba.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas de la investigación**:

- ✓ Analizar los procesos de negocio asociados a la atención y evaluación del neurodesarrollo en la especialidad de nutrición del grupo de Atención Temprana y Evaluación del Neurodesarrollo del Hospital Pediátrico Universitario William Soler de Cuba.
- ✓ Realizar un análisis acerca de los sistemas informáticos existentes a nivel nacional e internacional referentes a las consultas especializadas en nutrición, específicamente en consultas donde se evalúe y de seguimiento al neurodesarrollo en niños de cero a cinco años de edad.
- ✓ Asimilar la arquitectura definida por el Departamento Gestión Hospitalaria para el desarrollo de sus aplicaciones.
- ✓ Modelar el proceso actual de atención y evaluación del neurodesarrollo de los niños egresados de terapia intensiva al ser atendidos por el especialista en Nutrición.
- ✓ Analizar las necesidades de funcionamiento de la aplicación describiendo la especificación de Requisitos de Software.

- ✓ Realizar el modelado de casos de uso del sistema.
- ✓ Diseñar el sistema informático utilizando la arquitectura definida por el Departamento Gestión Hospitalaria.
- ✓ Realizar el diseño de la base de datos.
- ✓ Implementar el sistema informático aplicando las pautas de diseño y siguiendo las necesidades de funcionamiento establecidas en la Especificación de Requisitos de Software.

El desarrollo del módulo de Nutrición del Sistema de Evaluación del Neurodesarrollo en Niños proporcionará un grupo de beneficios. Entre ellos se encuentran:

- ✓ Mayor rapidez en el diagnóstico médico, logrando posteriormente niños con mejor calidad de vida y menos discapacidades que limiten su vida útil y complejicen el entorno familiar.
- ✓ Recibir una atención de salud que garantice una mayor seguridad y confiabilidad de su información médica.
- ✓ Disponer de un sistema que tenga como centro de referencia al paciente dando respuesta a sus necesidades de salud reales y de asistencia médica.
- ✓ Facilitará que el programa de evaluación y atención temprana se pueda comenzar a realizar en todos los hospitales pediátricos de Cuba.
- ✓ El país contará con una aplicación gratuita y de libre distribución.
- ✓ Se podrán atender mayor cantidad de pacientes en el día.

Para lograr una mayor organización, el documento se estructuró en cuatro capítulos, los que se describen a continuación:

**Capítulo I: Fundamentación Teórica.** Se analizan conceptos relacionados con la Atención Temprana y Evaluación del Neurodesarrollo. Se hace referencia de manera crítica y valorativa a los diferentes sistemas existentes tanto en el ámbito nacional como internacional y que están vinculados a funcionalidades que se llevan a cabo en el programa de atención y evaluación del neurodesarrollo del

Hospital Pediátrico Universitario “William Soler”. Además se describe la metodología, tecnologías y herramientas escogidas para la solución del problema.

**Capítulo 2: Características del Sistema.** Se expone una definición del objeto de estudio del problema donde se plantean los objetivos estratégicos de la organización y los procesos de negocio que los soportan. Se realiza una descripción de los procesos que serán objeto de automatización así como una descripción general de la propuesta de sistema. Se proporciona una descripción de las características del sistema que se desea construir, detallándose las funcionalidades que permitirán a los usuarios finales contar con una herramienta útil para llevar a cabo el proceso de transferencia del conocimiento. Se especifican además las cualidades, capacidades o propiedades que el sistema cumplirá, en términos de requerimientos y casos de uso.

**Capítulo 3: Análisis y Diseño del Sistema.** Se centra en la modelación detallada y la construcción de la estructura de la aplicación, sentando las bases para la futura implementación. Se realizan las definiciones del Modelo de Análisis y de Diseño. Comprende además los diferentes diagramas de diseño, así como los diagramas de interacción y las descripciones de las diferentes clases utilizadas en el mismo.

**Capítulo 4: Implementación.** Se hace una descripción de las tablas del Modelo de Datos. Se explica cómo está estructurado físicamente el sistema implementado mediante los modelos de despliegue y de componentes. Se describen las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de la evaluación del neurodesarrollo en niños; además de especificar detalles del tratamiento de errores, la seguridad y las estrategias de codificación.

## Capítulo 1: Fundamentación Teórica

El presente capítulo hace referencia al estado del arte para determinar los antecedentes de la Atención Temprana. Se brinda la base teórica y conceptual para el desarrollo del Módulo de Nutrición del Sistema de Evaluación de Neurodesarrollo en Niños. Se describen las tecnologías, metodologías y herramientas propuestas para darle solución al problema planteado.

### 1.1. **Conceptos Básicos**

Neurodesarrollo: Adquisición de funciones, dependientes del sistema nervioso, que implican un incremento de estructuras orgánicas y funcionales a través de un proceso de maduración.

Estimulación Temprana: Conjunto de acciones tendientes a proporcionar al niño las experiencias que éste necesita desde su nacimiento, para desarrollar al máximo su potencial psicológico.

Atención Temprana: Conjunto de acciones dirigidas no solo al niño sino también a la familia y la comunidad, para el desarrollo armónico de los niños con su entorno.

Nutrición: Proceso biológico en el que los organismos asimilan los alimentos y los líquidos necesarios para el funcionamiento, el crecimiento y el mantenimiento de sus funciones vitales. También es la ciencia que estudia la relación que existe entre los alimentos y la salud, especialmente en la determinación de una dieta.

Estado Nutricional: Estado de salud de una persona en relación con los nutrientes de su régimen de alimentación. Se determina mediante la valoración del crecimiento en los niños y los cambios en la masa corporal de los adultos, refleja diversos grados de bienestar, que en sí mismos son consecuencia de una compleja interacción entre la dieta, factores relacionados con la salud y el entorno físico, social y económico.

### 1.2. **Antecedentes de la Atención Temprana**

La Atención Temprana (AT), desde los principios científicos que se fundamenta, tiene como finalidad ofrecer a los niños con déficit o con riesgo de padecerlos, un conjunto de acciones optimizadoras y

compensadoras, que faciliten su adecuada maduración en todos los ámbitos y que les permita alcanzar el máximo nivel de desarrollo personal y de integración social.

Se podría decir que existen una serie de razones que avalan la importancia decisiva de la Atención Temprana en estas primeras edades: (1)

1º Las bases del desarrollo de las capacidades fundamentales tienen lugar en los primeros años de vida.

2º Desde el nacimiento, los procesos de maduración y desarrollo son posibles gracias a la interrelación e interacción del niño con el medio; por ello, cobran especial relevancia desde las primeras edades y singular significado en aquellos niños que presentan algún tipo de necesidad especial, ya sea como consecuencia de alteraciones en su desarrollo o por deficiencias en el nacimiento.

3º Las primeras edades constituyen un período vital, caracterizado por un potente y cambiante ritmo evolutivo y de desarrollo, donde la plasticidad y flexibilidad de las estructuras fisiológicas y psicológicas del niño adquieren un máximo exponente decisivo para su desarrollo posterior. Desde este punto de vista cualquier anomalía de las capacidades debe ser compensada lo más pronto posible, a fin de evitar limitaciones en su posterior evolución.

La AT persigue objetivos que permiten precisar y caracterizar su aplicación y desarrollo en todas las áreas de intervención. Específicamente en el área de Nutrición ejerce un papel muy importante pues permite al paciente enfermo alcanzar niveles nutritivos correctos, lo que le facilita al mismo un pleno desarrollo de sus funciones vitales.

### **1.3. Sistemas Automatizados Existentes**

#### Sistema de Vigilancia Alimentaria y Nutricional (SISVAN)

Los Sistemas de Vigilancia Alimentaria y Nutricional (SISVAN), son una estrategia acertada para combatir los problemas alimentario-nutricionales.

Cuba posee un Sistema de Vigilancia desde 1977. Estos sistemas son diseñados para la planificación de políticas, programas y ofrecen información sobre la disponibilidad, accesibilidad, consumo de alimentos y

# Capítulo 1: Fundamentación Teórica.

---

el estado nutricional. Por su capacidad y prestigio técnico contribuye a crear conciencia acerca de las implicaciones socioeconómicas de los problemas nutricionales.

Los Sistemas de Vigilancia Alimentaria y Nutricional son de mucha ayuda en el país, pero no son aplicables a las necesidades existentes en el Hospital Pediátrico "William Soler", además de no cumplir con las pautas de desarrollo empleadas en el Área Temática: Sistemas Especializados.

## Planificador Nutricional (Planut): software para mejorar la alimentación

Es un nuevo software diseñado por la Organización Panamericana de la Salud (OPS) y el Instituto Panamericano de Protección de Alimentos y Zoonosis (INPPAZ), está destinado a los profesionales de la nutrición y de la salud. Su propósito es facilitar la elaboración de recomendaciones sobre nutrición para diferentes poblaciones, que permitan lograr el máximo beneficio nutricional a menos costo. Permite que cada usuario establezca sus propias metas de alimentación. (2)

A pesar de ser un software muy moderno, no ofrece interés para la investigación porque no cumple los requisitos necesarios para satisfacer al cliente. Además tampoco cumple con la arquitectura definida para el proyecto SENDN.

## El Buscador de Complementos Nutricionales

Este software puede ayudarle a buscar complementos de salud y nutricionales, escogiendo entre 10 marcas. Tras elegir la marca, le devolverá una lista de productos recomendados.

Como lo indica su nombre, adicionan el nutriente que le hace falta al ser humano: proteínas, vitaminas o minerales. Pero es importante tener en cuenta que su función es complementar la alimentación, no reemplazarla.

Estos complementos tienen contenido de ácido fólico, hierro, calcio y zinc, que, en el caso de las mujeres embarazadas que los consumen, pueden ayudar a la formación del bebé. En cuanto a los niños, sólo en caso de deficiencia nutricional, inapetencia regular o enfermedad crónica, es conveniente el uso de estos productos, previa valoración del nutricionista. (3)

El Buscador de Complementos Nutricionales es un programa informático con un alto poder adquisitivo, de difícil alcance por numerosos países del mundo. Por otra parte, no se adapta a las características del sistema de salud cubano, ni al programa de atención y evaluación que se lleva a cabo en el Hospital Pediátrico Universitario "William Soler".

## Módulo de Antropometría y Composición Corporal

El Módulo de Antropometría y Composición Corporal de Nutripac permite analizar bases de datos antropométricas y realizar investigaciones analizando los datos de 100, 500 ó 1000 pacientes. El módulo está enfocado a la investigación científica, por lo que no hace seguimiento de pacientes, pero sí puede analizar pacientes individuales.

Este módulo no es específico para la atención de niños, sino para personas de cualquier edad. Además, no permite darles seguimiento a los pacientes, por lo que no es aplicable en el proceso de investigación que se lleva a cabo en el Hospital Pediátrico "William Soler".

## Equilibra - Medisys S.A

EquiLibra Profesional (2006) está desarrollado para profesionales de la nutrición que requieren de un asistente para realizar cálculos nutricionales y soporte informático para llevar un registro nutricional de cada uno de sus pacientes.

Además le ayudará a determinar, sobre la base de datos antropométricos y valores provenientes de exámenes específicos de laboratorio, el objetivo nutricional diario de cada usuario, o paciente a evaluar y le proporcionará la información y mecanismos necesarios para generar la dieta balanceada y la actividad física adecuada para cada necesidad.

Existen diversas versiones de este software, cada una de ellas con precios diferentes, llegando hasta un monto de 950 US\$. Es un software propietario y no se ajusta a las condiciones y pautas a seguir en el programa de atención y evaluación que se lleva a cabo en el Hospital Pediátrico Universitario "William Soler".

### **1.4. Metodología de Desarrollo**

Una Metodología de Desarrollo de Software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un software.

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. Indica cómo hay que obtener los distintos productos parciales y finales.

## Proceso Unificado de Desarrollo RUP

Es un proceso de desarrollo de software que constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico proceso unificado. Sus principales características son:

- ✓ Iterativo e incremental
- ✓ Centrado en la arquitectura
- ✓ Guiado por los casos de uso

## Lenguaje Unificado de Modelado (UML).

Lenguaje utilizado para especificar, visualizar y documentar los componentes de un sistema en desarrollo orientado a objetos. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

## Notación para el modelado de procesos del negocio (BPMN).



Es un nuevo estándar para modelar flujos de procesos de negocio y servicios web. Es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo.

El principal objetivo de BPMN es proveer una notación estándar que sea fácilmente leíble y entendible por parte de todos los involucrados e interesados del negocio. Tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.

En síntesis BPMN está planeada para dar soporte únicamente a aquellos procesos que sean aplicables a procesos de negocios. Esto significa que cualquier otro tipo de modelado realizado por una organización con fines distintos de los de negocio no estará dentro de su ámbito.

## **1.5. Tecnologías**

Para modelar un sistema deben tomarse como referencia una serie de herramientas, metodologías y prácticas adaptables a las características de la entidad, respetando sobre todo los requisitos del cliente, argumentando también las herramientas informáticas disponibles para esta tarea.

Las tecnologías a utilizar en el proceso de desarrollo del sistema deben ser libres, multiplataforma y que faciliten u optimicen el mismo. Las mismas aparecerán según su ubicación en las capas de presentación, negocio y acceso a datos.

### **1.5.1 Patrones de arquitectura y diseño**

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos para la creación de un producto de software. Establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema.

Son los patrones los que permitirán hablar posteriormente de la arquitectura, en términos de abarcar también las decisiones más importantes en relación con el sistema a desarrollar. A continuación se mencionan las definiciones de los patrones escogidos, así como sus ventajas.

## Modelo-Vista-Controlador (MVC)

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: el Modelo, las Vistas y los Controladores. (4)

Modelo: Administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado, usualmente formulados desde la vista, respondiendo a instrucciones de cambio para cambiar el estado de estos datos, habitualmente desde el controlador.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Utilizando este tipo de patrones se consigue: más calidad, mejor mantenibilidad, pero una de las cosas más importantes es la normalización y estandarización del desarrollo de Software.

## Patrón en capas

El patrón en capas define como su propio nombre lo indica, tres elementos esenciales: (5)

- ✓ Capa de presentación: Presenta el sistema al usuario, comunica y captura la información del usuario. Esta capa se comunica únicamente con la capa de negocio.
- ✓ Capa de Negocio: La capa de negocio es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio o incluso de lógica del negocio porque es aquí donde se establecen todas las reglas que

deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para almacenar o recuperar los mismos.

- ✓ Capa de Datos: La capa de acceso a datos contiene clases que interactúan con la base de datos, estas clases altamente especializadas permiten, utilizando los procedimientos almacenados (funciones para interactuar con la base de datos) generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.

El patrón en capas y el patrón MVC, pueden relacionarse lógicamente mediante cada uno de sus elementos, donde la Capa de presentación podría corresponderse con la Vista, la Capa de negocio con el Controlador y la Capa de datos con el Modelo.

## *Capa de presentación*

### **Java Server Faces (JSF)**

Java Server Faces (JSF) es un framework Java que permite crear interfaces de usuario (UI) para aplicaciones web, mediante componentes reutilizables. Permite el manejo de estados y eventos, así como la asociación entre los datos de la interfaz y los datos de la aplicación web, la especificación de la navegación del usuario.

### **Richfaces**

JBoss RichFaces es una librería de componentes web enriquecidos, de código abierto y basada en el estándar JSF. Es un framework que permite la integración de AJAX con facilidad en la capacidad de desarrollo de aplicaciones de negocio. Provee facilidades de validación y conversión de los datos proporcionados por el usuario, administración avanzada de recursos como imágenes, código JavaScript y Hojas de Estilo en Cascada (CSS).

RichFaces posee componentes que vienen listos para su uso out-of-the-box, por lo que los desarrolladores pueden ahorrar tiempo de inmediato para aprovechar las características de los componentes para crear aplicaciones Web que proporcionan mejoras en gran medida la experiencia del usuario más fiable y rápidamente. Se integra completamente dentro del ciclo de vida JSF.

## **Ajax4JSF**

Es una biblioteca de código abierto que se integra totalmente en la arquitectura de JSF y hereda las funcionalidades de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript.

Mediante este framework se puede:

- ✓ Variar el ciclo de vida de una petición JSF.
- ✓ Recargar determinados componentes de la página sin necesidad de recargarla por completo.
- ✓ Realizar peticiones automáticas al servidor, control de cualquier evento de usuario, etc.

En definitiva Ajax4JSF permite dotar a la aplicación JSF de mucho contenido profesional con muy poco esfuerzo.

## **Facelets**

Facelets es un framework simplificado de presentación, que permite diseñar de forma libre una página web y luego asociarle los componentes JSF específicos.

Estas son las propiedades más interesantes de Facelets:

- ✓ Trabajo basado en plantillas.
- ✓ Fácil composición de componentes.
- ✓ Creación de etiquetas lógicas a la medida.
- ✓ Funciones para expresiones.
- ✓ Desarrollo amigable para el diseñador gráfico.

- ✓ Creación de librerías de componentes.
- ✓ Es un framework de diseño de páginas muy fácil de usar y configurar.

## **Lenguaje Extensible de Marcado de Hipertexto (XHTML)**

XHTML, acrónimo en inglés de eXtensible Hypertext Markup Language (Lenguaje Extensible de Marcado de Hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.

XHTML es solamente la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Tiene como objetivo, reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML.

Permite una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella. Puede incluir otros lenguajes como MathML, SMIL o SVG, al contrario que HTML.

*Capa de negocio*

## **JBoss Seam**

Es un framework desarrollado por la empresa JBoss. Permite el desarrollo de aplicaciones web en Java. Combina a los 2 frameworks Enterprise JavaBeans EJB3 y JavaServerFaces JSF. Se puede acceder a cualquier componente EJB desde la capa de presentación refiriéndote a él mediante su nombre de componente seam. Realiza nuevos aportes en el campo de la administración de estado y provee una mayor granularidad de contextos de estado.

*Capa de acceso de datos*

## **Hibernate**

Framework que provee herramientas de mapeo objeto/relacional y permite reducir significativamente el tiempo de desarrollo.

Hibernate proporciona además, un lenguaje con el que realizar consultas a la base de datos. Este lenguaje es similar a SQL y es utilizado para obtener objetos de la base de datos según las condiciones especificadas en el HQL (Hibernate Query Language).

El uso de HQL permite usar un lenguaje intermedio que según la base de datos que se usa y el dialecto que se especifique será traducido al SQL dependiente de cada base de datos de forma automática y transparente.

Es open source, lo que supone, entre otras cosas, que no hay que pagar nada por adquirirlo. Se integra en cualquier tipo de aplicación justo por encima del contenedor de datos.

Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

## **Enterprise JavaBeans (EJB3)**

Arquitectura componente del lado del servidor para la plataforma Java. Permite realizar la administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portables y persistencia de datos.

El contenedor EJB es parte del servidor de aplicaciones (usualmente el contenedor EJB y el servidor de aplicaciones no pueden ser separados y son proporcionados por el mismo proveedor). El servidor de aplicaciones provee el ciclo de vida de los componentes, así como servicios de seguridad y manejo de transacciones.

Una de las metas de la arquitectura EJB es la de poder escribir de manera fácil aplicaciones de negocio orientadas a objetos y distribuidas, basadas en el lenguaje de programación JAVA.

## **Java Persistence API (JPA)**

Es la Interfaz de Programación de Aplicaciones (API) estándar para la persistencia y el mapeo objeto/relacional para la plataforma Java EE, y permite utilizar un modelo de dominio Java para administrar bases de datos relacionales.

El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí ocurría con EJB2, y permitir usar objetos regulares.

## **1.5.2. Tecnologías horizontales**

Existen un conjunto de tecnologías que se extienden horizontalmente por todas las capas antes mencionadas y sirven de soporte a las tecnologías que se utilizan en cada una de ellas. Las mismas se describen a continuación.

### **Java Platform Enterprise Edition (JavaEE 5)**

Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con arquitectura de niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

Java EE también configura algunas especificaciones únicas para el manejo de sus componentes. Estas incluyen Enterprise JavaBeans, servlets, JavaServer Pages y varias tecnologías de servicios web.

Permite al desarrollador la creación de aplicaciones portables entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Permite además que el servidor de aplicaciones pueda manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

### **Java Runtime Environment (JRE)**

Es un conjunto de herramientas, provisto por la empresa creadora del lenguaje JAVA, Sun Microsystems, para la ejecución del código.

JVM (Máquina Virtual Java) es una instancia de JRE en tiempo de ejecución, este es el programa que interpreta el código Java, mientras que las librerías de clases estándar son las que implementan las API de Java. Ambas JVM y API deben ser consistentes entre sí, de ahí que sean distribuidas de modo conjunto.

Un usuario sólo necesita el JRE para ejecutar las aplicaciones desarrolladas en lenguaje Java, mientras que para desarrollar nuevas aplicaciones en dicho lenguaje es necesario un entorno de desarrollo, denominado JDK, que además del JRE (mínimo imprescindible) incluye, entre otros, un compilador para Java.

### **1.5.3 Lenguajes**

Los lenguajes de programación son herramientas que permiten crear programas y software. Los lenguajes de programación de una computadora en particular se conocen como código de máquinas o lenguaje de máquinas.

Los mismos facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computador a utilizar. Representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona.

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa: sobre qué datos una computadora debe operar, cómo deben ser estos almacenados y transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. (6)

### **Java**

Lenguaje de programación desarrollado por la compañía Sun Microsystems, que utiliza el paradigma de la Programación Orientada a Objetos (POO). El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel. Dentro de sus principales ventajas se encuentra la de ser multiplataforma.



Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos.

Las características principales que ofrece Java son:

- ✓ Simple: Es un lenguaje rápido y fácil de aprender.
- ✓ Orientado a objetos: Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.
- ✓ Robusto: Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución.

## ***Servidor de aplicaciones***

Dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones.

## **JBoss Server o JBoss AS**

Servidor de aplicaciones Java más utilizado actualmente en el mercado. Cientos de profesionales y desarrolladores de código abierto han contribuido a su creación y desarrollo. Es una plataforma certificada que cumple con las especificaciones de J2EE para el despliegue y desarrollo de aplicaciones empresariales Java, aplicaciones web, y portales.

JBoss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business. Combinando una arquitectura orientada a servicios revolucionaría con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado y distribuido sin restricciones por la licencia.

## **Sistema Gestor de Bases de Datos**

Un sistema de gestión de la base de datos es denominado como: “Conjunto de elementos software con capacidad para definir, mantener y utilizar una base de datos.” (7)

Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad.

Algunas de las características deseables en un SGBD son:

- ✓ Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- ✓ Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- ✓ Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.

## **PostgreSQL**

Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés). PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Es un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Es más completo que MySQL ya que permite métodos

almacenados, restricciones de integridad, vistas, etc. aunque en las últimas versiones de MySQL se han hecho grandes avances en ese sentido.

Permite realizar relaciones de herencia entre tablas, definir sistemas de reglas y eventos. PostgreSQL ejecuta procedimientos almacenados en más de una docena de lenguajes de programación tales como Java, PERL, Python, Ruby, Tcl, C/C++, y su propio lenguaje PL. /pgSQL

## **1.6. Herramientas a utilizar**

Las herramientas de desarrollo son fundamentalmente editores de código que además pueden servir para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación.

### **Herramientas CASE**

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que favorecen y guían a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un software.

### **Visual Paradigm**

Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Incluye una herramienta llamada Visual Architect que permite la generación de código para el manejo de la base de datos.

Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas, generar documentación y ayuda a una más rápida construcción de aplicaciones de calidad a menor costo.

Entre sus características se reflejan las siguientes:

- ✓ Producto de calidad, soporta aplicaciones Web, fácil de instalar y actualizar, compatibilidad entre ediciones y comercial.

Muestra algunas ventajas como:

- ✓ Tiene disponible distintas ediciones: Enterprise, Professional, Standard, Modeler, Personal y Community (que es gratuita).
- ✓ La compañía facilita licencias especiales para fines académicos.
- ✓ Visual Paradigm (VP) para UML apoya un conjunto de idiomas tanto en la generación de código como en la Ingeniería Inversa en Java++, PHP, XML.
- ✓ Apoya la generación de código C#, VB.NET, Delpi, PERL, entre otros.
- ✓ Ambiente de Desarrollo Integrado (IDE)

### **Eclipse**

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas.

Fue desarrollado originalmente por IBM, como el sucesor de su familia de herramientas para VisualAge, pero es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código.

Como conclusiones de este capítulo se puede resaltar que se profundizó en el conocimiento de algunos conceptos necesarios para la comprensión del proceso. Se realizó un estudio sobre los principales sistemas existentes a nivel nacional e internacional y se determinó que estos no brindan una solución estándar para el proceso en desarrollo. Se efectuó un análisis de la elección de las metodologías, tendencias y herramientas propuestas en la política trazada por la dirección del proyecto.

### Capítulo 2: Características del Sistema.

En el presente capítulo se realiza un análisis de la propuesta del sistema. Se abordan las causas que originan la situación problemática y consecuencias, así como los procesos que serán objeto de automatización. El modelo de negocio, especificación de los requisitos del software y la definición de los casos de uso son también otros de los temas tratados.

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

En el Hospital Pediátrico Universitario “William Soler” se atienden a niños menores de 5 años de edad egresados de las Unidades de Cuidados Intensivos Polivalente y Neonatal (UCIP\UCIN), persiguiendo los siguientes objetivos:

- ✓ Brindarle al paciente grave un seguimiento para lograr una calidad de vida óptima.
- ✓ Pesquisar afecciones del neurodesarrollo de los niños en estudio.
- ✓ Evaluar los elementos que influyeron en la aparición de las secuelas.
- ✓ Utilizar la Clasificación de Zuluaga en los pacientes evaluados.
- ✓ Valorar evolución de los casos por grupos de tratamiento y patologías.
- ✓ Estandarizar protocolos de trabajo de acuerdo con los hallazgos encontrados.

Todos serán evaluados por un equipo interdisciplinario, que determinará la presencia o no de afecciones del neurodesarrollo tras su período de gravedad. Este equipo está conformado por los especialistas en: Neurología, Psicología, Fisiatría, Neurofisiología, Logopedia y Nutrición.

#### 2.2. Descripción de los Procesos de Negocio

Los especialistas en Nutrición para evaluar a los niños necesitan realizarle una serie de evaluaciones entre las que se encuentran: Evaluación Clínica, Evaluación Antropométrica y la Evaluación Bioquímica.

## *Capítulo 2: Características del Sistema.*

---

La Evaluación Clínica es realizada por el Especialista en Nutrición y tiene como objetivo conocer el comportamiento del paciente en ese momento. Para tener un buen resultado de esta evaluación es necesario dividirla en dos partes: Evaluación Física General y Evaluación Física por Aparatos.

La Evaluación Física General mide una serie de parámetros, aplicables al paciente, para mejorar de acuerdo con su resultado, las afecciones en su neurodesarrollo. Por su parte la Evaluación Física por Aparatos hace funciones similares.

La Evaluación Antropométrica es llevada a cabo por la Técnica en Antropometría, esta evaluación es una práctica sistematizada que expresa de manera cuantitativa las dimensiones del cuerpo humano. Antes de los 2 años de edad debe tomarse en decúbito supino y se denomina longitud supina, después de esta edad el paciente permanece de pie en posición de firmes.

Mediante esta evaluación se pueden saber disímiles resultados del paciente como es el caso de:

- ✓ Índice de Masa Corporal (IMC)
- ✓ Circunferencia Muscular de Brazo (CMB)
- ✓ Pliegue Tricipital (Pt)
- ✓ Área Total del Brazo (ATB)
- ✓ Área Muscular del Brazo (AMB)
- ✓ Área Grasa del Brazo (AGB)
- ✓ Índice Nutricional de McLaren (IN)

La Evaluación Bioquímica permite saber las condiciones del organismo del paciente, puesto que se realizan una serie de análisis complementarios que ayudan a avalar dichas condiciones. Dentro de los parámetros a medir en este tipo de evaluación se encuentran:

- ✓ Hematometría.

- ✓ Glucosa en Sangre.
- ✓ Química Sanguínea

El Especialista de Nutrición es el encargado de recoger los resultados de esta evaluación para tenerlos en cuenta a la hora de emitir un diagnóstico.

Cada una de estas evaluaciones debe elaborar una hoja de evaluación con el resultado de las mismas, el cual será analizado por el Especialista en Nutrición, quién examinará físicamente al paciente valorando el desarrollo de su estado nutricional, aspecto de esencial importancia en esta investigación y emitirá el resumen final.

### **2.3. Análisis crítico de ejecución de los procesos actuales**

En estos momentos toda la gestión de la información asociada a los procesos de evaluación de la calidad del neurodesarrollo en el Hospital Pediátrico “William Soler” se lleva a cabo de forma manual.

Esta situación dificulta la rapidez de las evaluaciones realizadas por los especialistas en el área de Nutrición ya que son muy extensas y requieren de mucho tiempo, impidiendo atender a gran cantidad de niños el mismo día.

Además, se dificulta la comparación de diagnósticos emitidos en diferentes consultas. Se hace difícil brindar determinada información estadística puesto que no hay forma de llevarla manualmente.

### **2.4. Objeto de Automatización**

Con el objetivo de solucionar los problemas existentes en el Hospital Pediátrico “William Soler”, en el proceso de evaluación de la calidad del neurodesarrollo por los especialistas en nutrición, se propone realizar una aplicación web, que tiene como objeto de automatización inicial, las evaluaciones que se realizan en esta especialidad, posibilitando una mayor confidencialidad, seguridad y control de la información que se manipula, así como la disminución del tiempo de aplicación de dichas evaluaciones.

### **2.5. Modelo del negocio**

## Capítulo 2: Características del Sistema.

Es una representación simplificada de la lógica de negocio. Describe lo que un negocio ofrece a sus clientes, cómo llega a ellos, y cómo se relaciona con ellos. La finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades, roles y las reglas del mismo. Se enfoca en comprender los problemas actuales de la organización e identifica mejoras potenciales.

### 2.5.1. Identificación de roles del entorno del negocio

Una vez identificados los procesos de negocio, es preciso encontrar los involucrados en su realización. Cada uno de éstos, desempeña cierto papel (rol).

Actor del Negocio: Cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. (8)

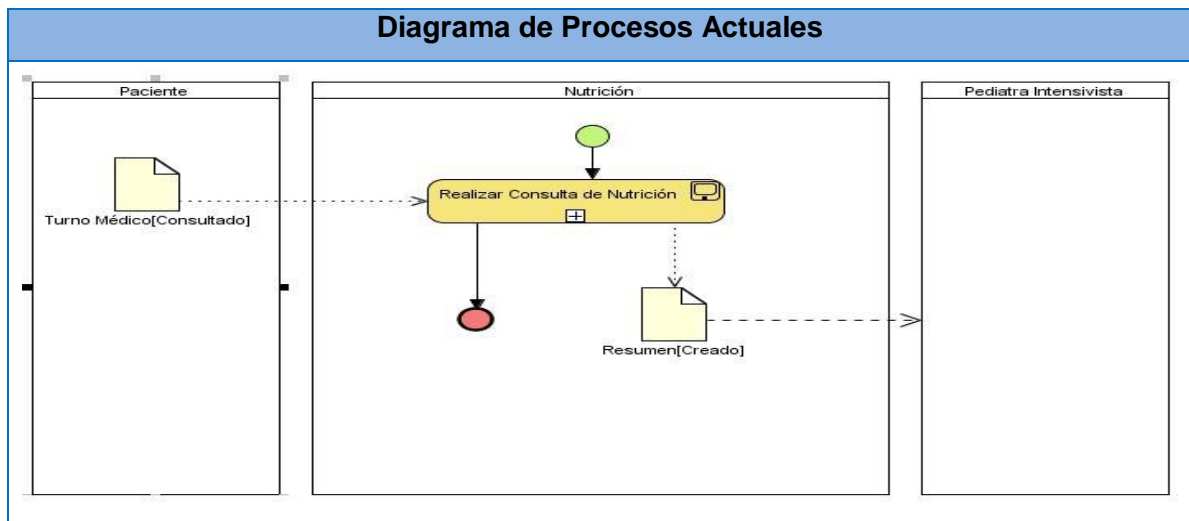
Actores del negocio	Descripción
Paciente	Niños de 0 a 5 años de edad egresados de las Unidades de Cuidados Intensivos Neonatales (UCIN) y Cuidados Intensivos Polivalentes (UCIP).

Trabajador del negocio: Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos. (9)

Trabajador del negocio	Descripción
Especialista en Nutrición	Es el especialista encargado de realizar la evaluación clínica, analizar los resultados de la evaluación bioquímica y emitir el resumen final.
Técnica en Antropometría	Es la persona encargada de realizar las actividades relacionadas con las mediciones del cuerpo del paciente, es decir, la evaluación antropométrica.



### Diagrama de proceso del negocio



### Descripción de los Procesos del Negocio

<b>PROCESO:</b>	Realizar Consulta de Nutrición.
<b>Misión:</b>	Aplicar las evaluaciones necesarias para obtener el estado nutricional de paciente.
<b>Responsable:</b>	Dr. Lázaro Alfonso Novo (MSc. en Nutrición del Hospital Pediátrico Universitario William Soler).
<b>ACTORES INVOLUCRADOS</b>	
<b>Rol</b>	<b>Funciones</b>
Especialista en Nutrición.	Es el/la principal ejecutor(a) de las actividades relacionadas con las evaluaciones que se realizan dentro del proceso Realizar Consulta de Nutrición. Además, analiza el resultado de todas las evaluaciones para valorar el estado nutricional del paciente.
Tec. Antropometría.	Es el/la principal ejecutor(a) de las actividades relacionadas con la evaluación antropométrica.

## *Capítulo 2: Características del Sistema.*

<b>ACTIVIDADES</b>				
<b>Actividad:</b>	Verificar Turno			
<b>Flujo de Información</b>				
El/La paciente llega a la consulta y entrega su turno médico. El/La Especialista en Nutrición verifica que le corresponda el turno ese día. En caso de no corresponderle, el paciente se retira de la consulta, de lo contrario el Especialista en Nutrición procede a realizarle la Evaluación Clínica.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Turno médico	Pediatra intensivista	Especialista en Nutrición	Papel	Cada vez que se planifique una consulta.
<b>Actividad:</b>	Evaluación Clínica			
<b>Flujo de Información</b>				
Para realizar la evaluación del paciente se le realiza el examen Físico General y por Aparatos. Estos resultados quedan registrados en la Hoja de Evaluación Clínica. Luego el Especialista en Nutrición le indica al paciente realizarse los exámenes complementarios.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Hoja de Evaluación Clínica.	Especialista en Nutrición.	Especialista en Nutrición.	Papel	Cada vez que se planifique una consulta.
Orden de exámenes complementarios.	Especialista en Nutrición.	Paciente.	Papel	Cada vez que se planifique una consulta.
<b>Actividad:</b>	Evaluación Antropométrica			
<b>Flujo de Información</b>				
El paciente es atendido por la Técnica en Antropometría, está es la encargada de realizarle la Evaluación Antropométrica. La misma para generar los resultados pertinentes hace uso de las tablas de percentiles, las que le permiten ubicar en que rango percentilar se encuentra el paciente en ese momento. Este resultado es registrado en la Hoja de Evaluación Antropométrica.				

## *Capítulo 2: Características del Sistema.*

Artefacto	Emisor	Receptor	Formato	Frecuencia
Percentil Circunferencia del brazo. (femenino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Circunferencia del brazo. (masculino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Circunferencia Cefálica. (femenino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Circunferencia Cefálica. (masculino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Pliegue Tricipital. (femenino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Pliegue Tricipital. (masculino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Área Muscular. (femenino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Área Muscular. (masculino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Área	Técnica en	Técnica en	Papel	Cada vez que se realiza

## *Capítulo 2: Características del Sistema.*

Grasa. (femenino)	Antropometría.	Antropometría.		una evaluación antropométrica.
Percentil Área Grasa. (masculino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Talla\Edad. (femenino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Talla\Edad. (masculino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Peso\Edad. (femenino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Peso\Edad. (masculino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Peso\Talla. (femenino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Percentil Peso\Talla. (masculino)	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Edad Decimal.	Técnica en Antropometría.	Técnica en Antropometría.	Papel	Cada vez que se realiza una evaluación antropométrica.
Hoja de Evaluación Antropométrica.	Técnica en Antropometría.	Especialista en Nutrición.	Papel	Cada vez que se realiza una evaluación antropométrica.
<b>Actividad:</b>	Evaluación Bioquímica.			

## Capítulo 2: Características del Sistema.

### Flujo de Información

Para poder realizarse esta evaluación es necesario haberse hecho previamente los exámenes complementarios indicados por el Especialista en Nutrición. Esta evaluación permite conocer las condiciones del organismo del paciente en ese momento, pues dichos exámenes complementarios ayudan a avalar estas condiciones. El resultado encontrado en esta evaluación será recogido en la Hoja de Evaluación Bioquímica.

Artefacto	Emisor	Receptor	Formato	Frecuencia
Orden de exámenes complementarios.	Paciente	Especialista en Nutrición.	Papel	Cada vez que se realiza una evaluación bioquímica.
Hoja de Evaluación Bioquímica.	Especialista en Nutrición.	Especialista en Nutrición.	Papel	Cada vez que se realiza una evaluación bioquímica.
<b>Actividad:</b>	Emitir Resumen Final.			

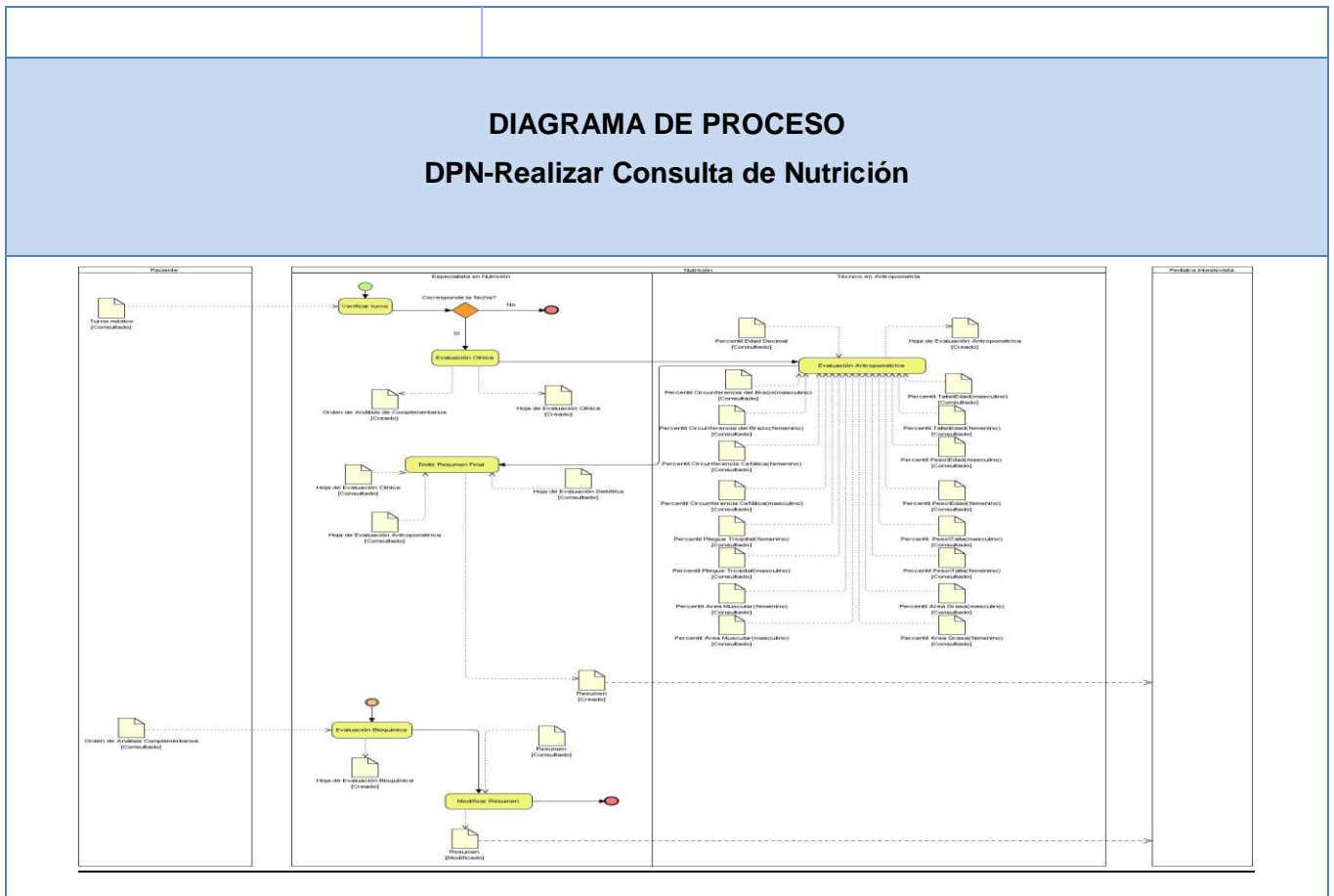
### Flujo de Información

Esta actividad es realizada por el Especialista en Nutrición, quién recibe las Hojas de Evaluación con el resultado de cada una de las evaluaciones que se le han hecho al paciente en el transcurso de la consulta. En este momento el Especialista en Nutrición examinará físicamente al paciente valorando su estado nutricional de manera general, lo que le permitirá emitir el resumen final de la consulta. En este resumen puede hacer todas las observaciones que considere necesarias para el mejoramiento del paciente.

Artefacto	Emisor	Receptor	Formato	Frecuencia
Hoja de Evaluación Clínica.	Especialista en Nutrición.	Especialista en Nutrición.	Papel	Cada vez que se planifique una consulta.
Hoja de Evaluación Antropométrica.	Técnica en Antropometría.	Especialista en Nutrición.	Papel	Cada vez que se planifique una consulta.

## Capítulo 2: Características del Sistema.

Hoja de Evaluación Dietética.	Lic. Nutrición y Alimentos.	Especialista en Nutrición.	Papel	Cada vez que se planifique una consulta.
Resumen	Especialista en Nutrición.	Pediatra intensivista	Papel	Cada vez que se planifique una consulta.
<b>Actividad:</b>	Modificar Resumen.			
<b>Flujo de Información</b>				
Esta actividad es realizada por el Especialista en Nutrición, quién recibe las Hojas de Evaluación con el resultado de cada una de las evaluaciones que se le han hecho al paciente en el transcurso de la consulta, incluyendo los resultados de la Evaluación Bioquímica. En este resumen puede hacer todas las observaciones que considere necesarias para el mejoramiento del paciente.				
<b>Artefacto</b>	<b>Emisor</b>	<b>Receptor</b>	<b>Formato</b>	<b>Frecuencia</b>
Resumen	Especialista en Nutrición.	Pediatra intensivista	Papel	Cada vez que se planifique una consulta.
<b>REGLAS DEL NEGOCIO</b>				
<b>Regla</b>		<b>Actividad</b>		
Para poder realizarse esta evaluación es necesario haberse realizado la Evaluación Clínica.		Evaluación Antropométrica		
Para poder realizarse esta evaluación es necesario haberse realizado previamente los exámenes complementarios.		Evaluación Bioquímica		
<b>REFERENCIAS A PROCEDIMIENTOS</b>				
<b>Procedimiento</b>		<b>Descripción</b>		



## 2.6. Especificación de los Requisitos de Software

Los requerimientos de software son condiciones o capacidades que debe tener el sistema para satisfacer las necesidades de un cliente, estos deben ser especificados por escrito. A continuación se muestra un listado de los requerimientos tanto funcionales como no funcionales de la aplicación. Es necesario hacer énfasis en la precisión con que se debe realizar esta tarea por cumplir un papel primordial en el proceso de producción de software, pues se enfoca en un área fundamental: la definición de lo que se desea producir, mediante una descripción más clara del comportamiento del sistema, minimizando los problemas derivados en el transcurso de su desarrollo.

### 2.6.1 Requerimientos Funcionales

## *Capítulo 2: Características del Sistema.*

---

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. (10)

RF 1- Realizar Evaluación Clínica

RF 2 - Visualizar Detalles Evaluación Clínica

RF 3 - Modificar Evaluación Clínica

RF 4 - Realizar Evaluación Antropométrica

RF 5 - Visualizar Detalles Evaluación Antropométrica

RF 6 - Modificar Evaluación Antropométrica

RF 7 - Realizar Evaluación Bioquímica

RF 8 - Visualizar Detalles Evaluación Bioquímica

RF 9 - Modificar Evaluación Bioquímica

RF 10 - Realizar Resumen Nutrición

RF 11- Visualizar Detalles Resumen Nutrición

RF 12 - Modificar Resumen Nutrición

RF 13 - Buscar Pacientes Atendidos Nutrición

RF 14 - Buscar Resultado Consulta Nutrición

RF 15 - Visualizar Resultado Consulta Nutrición

RF 16 - Buscar Datos Resumen Nutrición



RF 17 - Visualizar Datos Resumen Nutrición

RF 18 - Listar Pacientes a realizar Evaluación Bioquímica

RF 19 - Buscar Pacientes a realizar Evaluación Bioquímica

RF 20 - Listar Pacientes Atendidos Nutrición

RF 21 - Listar Pacientes a atender en Antropométrica

RF 22 - Buscar Pacientes a atender en Antropométrica

### **2.6.2 Requisitos No Funcionales**

Los requerimientos no funcionales son cualidades o propiedades que el producto debe tener, es decir, restricciones en el producto que está siendo desarrollado. Estos no describen lo que el software hará, sino cómo lo hará. Son muy importantes debido a que permiten a los clientes valorar las características no funcionales del producto como: usabilidad, rendimiento y portabilidad, que junto a las funcionalidades esperadas del software ayudarán a marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. Es preciso tener en cuenta que la definición de los requerimientos no funcionales cobra un valor adicional porque los errores que puedan existir en ellos son difíciles y caros de resolver. (11)

#### **Requerimientos de Usabilidad**

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

Usuarios normales: 30 días

Usuarios avanzados: 20 días

#### **Requerimientos de Fiabilidad**

## *Capítulo 2: Características del Sistema.*

---

- ✓ Se mantendrá seguridad y control a nivel de usuario, garantizando su acceso sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema.
- ✓ Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.
- ✓ Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- ✓ Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude. Los mecanismos serán capaces de informar al personal autorizado sobre posibles irregularidades que den indicios sobre la introducción de información falseada.
- ✓ El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.
- ✓ El sistema soportará el uso de firmas digitales para la transferencia de información cuya certificación sea imprescindible para validar el uso de la misma.
- ✓ El sistema implementará un control de cambios a determinados campos de información (seleccionados por su importancia), de forma tal que sea posible determinar cuáles han sido las actualizaciones que se le han realizado.
- ✓ 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.
- ✓ El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

### **Requerimientos de Eficiencia**

- ✓ El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.
- ✓ Se deberá usar siempre que sea posible el patrón Singleton (instancia única), destruir referencias que ya no estén siendo usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

### Requerimientos de soporte

- ✓ Seguridad de acceso y administración de usuarios
- ✓ Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos por direcciones IP.
- ✓ Monitoreo de funcionamiento
- ✓ Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación.
- ✓ Respaldo y recuperación de base de datos
- ✓ Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados.

### Auditoría

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

### Configuración de parámetros

Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

### Réplica

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

### Restricciones de diseño

El sistema estará dividido en las siguientes capas:

#### *Capas físicas*

**Cliente:** Computadora con cualquier tecnología o sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se recomienda IE 6 o superior o Firefox 2.x).

**Servidor de Aplicaciones:** Servidor con cualquier tecnología o sistema operativo que soporte el Java Runtime Environment (JRE) 1.5 o superior y al JBoss AS 4.2 o superior. Estas mismas condiciones se aplican para los servidores de aplicación del Centro de Datos.

**Servidor de Base de Datos:** Servidor con cualquier tecnología o sistema operativo que soporte a PostgreSQL Server 8.2 o superior en los servidores de base de datos de cada hospital, y Oracle 11g o superior para los servidores de base de datos del Centro de Datos.

#### *Capas lógicas*

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

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

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

### Requerimientos para la documentación de usuarios en línea y ayuda del sistema

Se posibilitará el uso de ayudas dinámicas y tutoriales en línea sobre el funcionamiento del sistema.

### Requerimientos de Interfaz

Interfaces de usuario

- ✓ Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.
- ✓ La interfaz contará con teclas de función y menús desplegados que faciliten y aceleren su utilización.
- ✓ La entrada de datos incorrecta será detectada claramente e informada al usuario.
- ✓ Todos los textos y mensajes en pantalla aparecerán en idioma español.
- ✓ Se incorporarán asistentes que faciliten el uso del sistema por los usuarios, en procesos con determinado nivel de complejidad, que lo guíen paso a paso para minimizar la posibilidad de errores.
- ✓ El diseño de la interfaz del sistema responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.
- ✓ Se diseñarán salidas de información en forma de gráficos, estadísticas, pronósticos y análisis comparativos que puedan ser usados como soporte para la toma de decisiones.
- ✓ El sistema incluirá reportes estándares y parametrizables que permitirán al usuario configurar la información de salida y el orden en que aparecen los datos. Las salidas se podrán generar en el formato de fichero PDF.

### Interfaces hardware

- ✓ Los equipos autoanalizadores se podrán programar desde el sistema para realizar análisis y obtener resultados automáticamente.

### Interfaces software

- ✓ Se interactuará con el sistema ALAS-PACS para realizar solicitudes y obtener resultados de estudios radiológicos e imagenológicos.

### Interfaces de comunicación

- ✓ Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven). La utilización de este estándar aporta las siguientes ventajas:
  - Es un estándar independiente de la plataforma y la tecnología.
  - Está ampliamente implementado, por lo que es posible conseguir otros productos compatibles con HL7.
  - El sistema usará el formato estándar WSDL (Lenguaje de descripción de servicios Web) para la descripción de los servicios web.
  - El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos.
  - El sistema utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

### 2.7. Modelo de casos de uso del sistema

El Modelo de Casos de Uso del Sistema es un artefacto de Ingeniería de Software que representa gráficamente a los procesos y su interacción con los actores constituyendo una entrada de gran valor para las siguientes fases de construcción de un software.

Este modelo está formado por actores, casos de usos y las relaciones que se establecen entre estos. Los casos de uso son las funciones que proporciona un sistema para añadir valor a sus usuarios. Los actores representan el entorno del sistema. Son representados por personas aunque también pueden ser sistemas o hardware externo que interactuará con el sistema.

Cada actor asume un conjunto coherente de papeles cuando interactúa con el sistema. Un usuario físico puede actuar como uno o varios actores, desempeñando los papeles de esos actores en su interacción con el sistema. Varios usuarios concretos pueden actuar como diferentes ocurrencias del mismo actor. Se comunican con el sistema mediante el envío y recepción de mensajes hacia y desde el sistema según este lleva a cabo los casos de uso.

### 2.7.1 Definición de actores del sistema

Un actor del sistema es una persona o la abstracción de un software que interactúa de alguna manera con el sistema: puede intercambiar información con él, ser un recipiente pasivo de información, representar el rol que juegan una o varias personas, un equipo o un sistema automatizado. (12)

#### Actores del Sistema

Actor	Descripción
Especialista en Nutrición	Es el/la principal ejecutor(a) de las actividades relacionadas con la aplicación de las evaluaciones que se realizan dentro del proceso Realizar Consulta de Nutrición. Además, analiza el resultado de cada una de las evaluaciones para valorar el estado nutricional del paciente.
Técnica en Antropometría	Es la persona encargada de realizar las actividades relacionadas con las mediciones del cuerpo del paciente, es decir, la evaluación antropométrica.
Administrador	Es el encargado de gestionar los nomencladores.

### 2.7.2. Diagrama de Casos de Uso del Sistema

## Capítulo 2: Características del Sistema.

---

Un diagrama de casos de uso describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores con una asociación entre cada par actor/caso de uso que interactúan.

Los casos de uso son las funciones que proporciona un sistema para añadir valor a sus usuarios. Los mejores casos de uso son aquellos que añaden el mayor valor al negocio que implanta el sistema.

El modelo de casos de uso se utiliza para conseguir un acuerdo con los usuarios y clientes sobre qué debería hacer el sistema para los usuarios. Especifica y delimita de una forma completa todas las formas posibles de utilizar el sistema para sus usuarios. (13)

### 2.7.3 Listado de Casos de Uso

CU - 1	Crear Evaluación Clínica
Actor	Especialista en Nutrición
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear Evaluación Clínica, el sistema brinda la posibilidad de introducir los datos para realizar dicha evaluación, el actor introduce los datos de la evaluación, el sistema crea la Hoja de Evaluación Clínica, el caso de uso termina.
Referencia	RF 2; RF 3

CU - 2	Modificar Evaluación Clínica.
Actor	Especialista en Nutrición
Descripción	El caso de uso inicia cuando se muestran los detalles de la evaluación clínica realizada en el día y el actor accede a la opción Modificar Evaluación Clínica, el sistema muestra los datos y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos de la Hoja de Evaluación Clínica, el caso de uso termina.
Referencia	RF 2



## Capítulo 2: Características del Sistema.

---

CU - 3	Ver Detalles Evaluación Clínica
Actor	Especialista en Nutrición.
Descripción	El caso de uso inicia cuando el actor realiza una Evaluación Clínica, al dar Aceptar se muestran los detalles de dicha evaluación, dando la opción de modificar la misma, el caso de uso termina.
Referencia	RF 1; RF 3

CU - 4	Crear Evaluación Antropométrica
Actor	Técnica en Antropometría
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear Evaluación Antropométrica, el sistema brinda la posibilidad de introducir los datos para realizar dicha evaluación, el actor introduce los datos de la evaluación, el sistema crea la Hoja de Evaluación Antropométrica, el caso de uso termina.
Referencia	RF 5; RF 6

CU - 5	Modificar Evaluación Antropométrica.
Actor	Técnica en Antropometría
Descripción	El caso de uso inicia cuando se muestran los detalles de la evaluación antropométrica realizada en el día y el actor accede a la opción Modificar Evaluación Antropométrica, el sistema muestra los datos y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos de la Hoja de Evaluación Antropométrica, el caso de uso termina.
Referencia	RF 5

CU - 6	Ver Detalles Evaluación Antropométrica.
Actor	Técnica en Antropometría

## Capítulo 2: Características del Sistema.

---

Descripción	El caso de uso inicia cuando el actor realiza una Evaluación Antropométrica, al dar Aceptar se muestran los detalles de dicha evaluación, dando la opción de modificar la misma, el caso de uso termina.
Referencia	RF 6
CU - 14	Buscar Resultado Consulta Nutrición.
Actor	Especialista en Nutrición Técnica en Antropometría Licenciada en Nutrición y Alimentos
Descripción	El caso de uso inicia cuando el actor accede a la opción Buscar Resultado Consulta Nutrición, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el resultado de la consulta, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra el resultado general de la consulta realizada al paciente que cumple con los criterios de búsqueda, el caso de uso termina.
Referencia	RF 1; RF 4; RF 7

CU - 15	Ver Resultado Consulta Nutrición.
Actor	Especialista en Nutrición Técnica en Antropometría Licenciada en Nutrición y Alimentos
Descripción	El caso de uso inicia cuando el actor selecciona un resumen y accede a la opción de Ver Resultado Consulta Nutrición, el sistema muestra el resultado de una consulta, el caso de uso termina.
Referencia	RF 14

CU - 16	Buscar Datos Resumen Nutrición
Actor	Especialista en Nutrición.

## Capítulo 2: Características del Sistema.

---

	Técnica en Antropometría. Licenciada en Nutrición y Alimentos.
Descripción	El caso de uso inicia cuando el actor accede a la opción Buscar Resumen Nutrición, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el resumen, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los resúmenes que cumplen con los criterios de búsqueda, el caso de uso termina.
Referencia	RF 17

CU - 17	Ver Datos del Resumen Nutrición.
Actor	Especialista en Nutrición. Técnica en Antropometría. Licenciada en Nutrición y Alimentos.
Descripción	El caso de uso inicia cuando el actor selecciona un resumen y accede a la opción de Ver Datos Resumen Nutrición, el sistema muestra los datos del resumen, el caso de uso termina.
Referencia	RF 16

CU - 18	Listar Pacientes a realizar Evaluación Bioquímica
Actor	Especialista en Nutrición.
Descripción	El caso de uso inicia cuando el actor accede a la opción Realizar Evaluación Bioquímica, el sistema muestra el listado de los pacientes atendidos que se les puede realizar la evaluación, el caso de uso termina.
Referencia	RF 9; RF 13

CU - 19	Buscar Pacientes a realizar Evaluación Bioquímica
Actor	Especialista en Nutrición.

## Capítulo 2: Características del Sistema.

---

Descripción	El caso de uso inicia cuando el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar un paciente que se pueda realizar la evaluación bioquímica, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra el listado que cumple con los criterios de búsqueda, el caso de uso termina.
Referencia	RF 18

CU - 20	Listar Pacientes Atendidos Nutrición
Actor	Especialista en Nutrición.
Descripción	El caso de uso inicia cuando el actor accede a la opción Realizar Crear Resumen, el sistema muestra el listado de los pacientes atendidos que se les puede realizar el resumen, el caso de uso termina.
Referencia	RF 10; RF 13

CU - 21	Listar Pacientes a atender Antropométrica
Actor	Técnica en Antropometría.
Descripción	El caso de uso inicia cuando el actor accede a la opción Realizar Evaluación Antropométrica, el sistema muestra el listado de los pacientes que se les puede realizar la Evaluación Antropométrica, el caso de uso termina.
Referencia	RF 4; RF 22

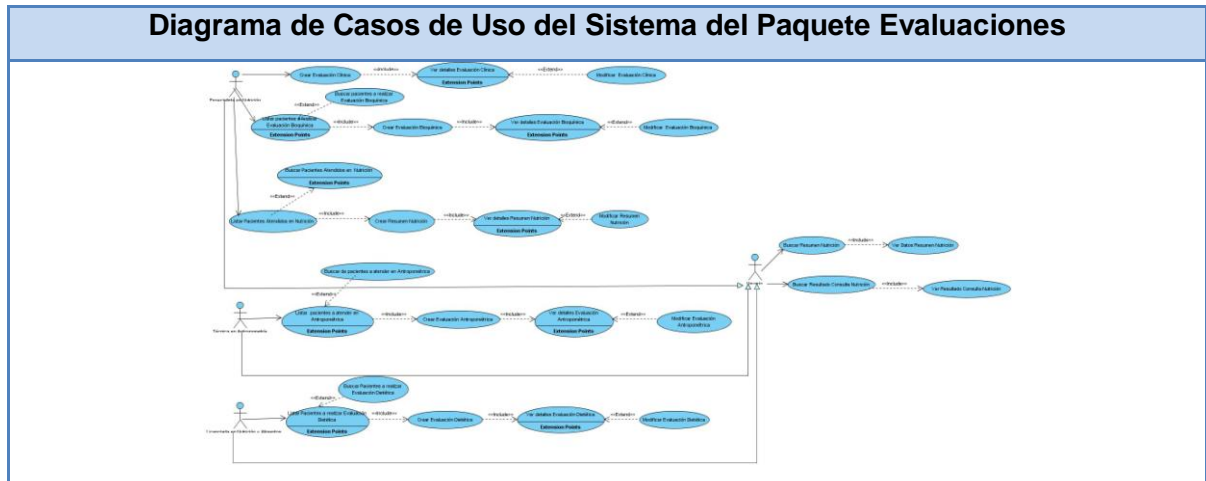
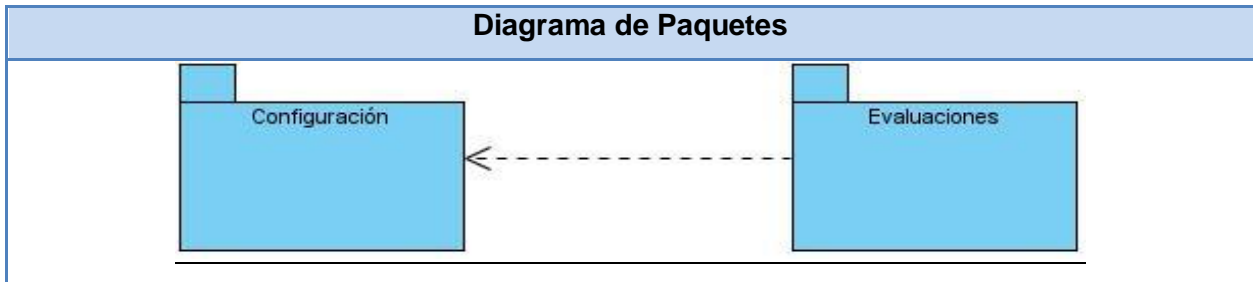
CU - 21	Buscar Pacientes a atender Antropométrica
Actor	Técnica en Antropometría.
Descripción	El caso de uso inicia cuando el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar un paciente que se pueda realizar la Evaluación Antropométrica , el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra el listado que cumple con los criterios de búsqueda, el caso de uso termina.

Referencia	RF 21
------------	-------

## 2.7.4 Organización de los Elementos: Casos de Uso en Paquetes.

Los paquetes son usados para organizar y manipular la complejidad de los modelos al alcanzar un tamaño considerable, agrupando los casos de uso en subconjuntos más pequeños. Para agrupar los casos de uso en paquetes se debe tener en cuenta que los elementos estén estrechamente relacionados entre sí.

Dependencias de paquetes: Es la relación que se establece entre paquetes, significa que los elementos de un paquete conocen de alguna forma los del otro y estos a su vez se encuentran acoplados.



### Casos de uso expandidos

A continuación se presenta el CU *Crear Resumen Nutrición*. Los demás pueden ser vistos en el SVN del Área Temática: Sistemas Especializados.

#### *Crear Resumen Nutrición*

<b>CASO DE USO:</b>	Crear Resumen Nutrición	
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción Crear Resumen Nutrición, el sistema brinda la posibilidad de introducir los datos para realizar el Resumen Nutrición, el actor introduce los datos del Resumen Nutrición, el sistema crea el Resumen Nutrición, el caso de uso termina.	
<b>Complejidad:</b>	Alta	
<b>Prioridad:</b>	1	
<b>Precondiciones:</b>	El paciente tiene que tener una cita para esa fecha y estar en el listado de pacientes atendidos en Nutrición.	
<b>REFERENCIAS</b>		
<b>Actores:</b>	Especialista en Nutrición	
<b>Requisitos:</b>	RF 11; RF 20	
<b>Entidades:</b>	Resumen	
<b>Casos de Uso:</b>	Listar pacientes atendidos en Nutrición Ver Detalles Resumen Nutrición	
<b>FLUJO NORMAL DE EVENTOS</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El caso de uso inicia cuando el actor accede a la opción Crear Resumen Nutrición.		

## Capítulo 2: Características del Sistema.

---

	<p>2. Muestra los datos predeterminados.</p> <ul style="list-style-type: none"><li>• Nombre</li><li>• Primer Apellido</li><li>• Segundo Apellido</li><li>• CI</li><li>• Fecha de nacimiento</li><li>• Sexo</li></ul> <p>Brinda la posibilidad de introducir los datos:</p> <ul style="list-style-type: none"><li>• Observaciones</li><li>• Aceptar crear Resumen.</li><li>• Cancelar operación. Ver <b>Alternativa 1</b>: “Cancelar operación.”</li></ul>
<p>3. Introduce los datos del resumen.</p> <p>Observaciones</p>	
<p>4. Selecciona la opción de aceptar Crear Resumen.</p>	
	<p>5. Valida los datos. Si hay datos incompletos, ver <b>Alternativa 2</b>: “Existen datos incompletos”. Si hay datos incorrectos, ver <b>Alternativa 3</b>: “Existen datos incorrectos.”</p>
	<p>6. Crea el resumen.</p>

## *Capítulo 2: Características del Sistema.*

	7. Muestra el mensaje de información "Se ha creado el resumen."
	8. El caso de uso termina.
<b>[Prototipo de Interfaz]</b>	
<b>FLUJOS ALTERNOS</b>	
<b>Alternativa 1. "Cancelar operación."</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción de Cancelar operación.	
	2. Regresa a la vista anterior.
	3. El caso de uso termina.
<b>Alternativa 2. "Existen datos incompletos."</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. Muestra el mensaje de error "Existen campos vacíos que son obligatorios, por favor, complete estos datos."
	2. Muestra un indicador sobre los campos vacíos.
<b>[Prototipo de Interfaz]</b>	
<b>Alternativa 3. "Existen datos incorrectos."</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. Muestra el mensaje de error "Existen campos escritos incorrectamente, por favor, rectifique estos datos."
	2. Muestra un indicador sobre los campos incorrectos.
<b>[Prototipo de Interfaz]</b>	
<b>Poscondiciones</b>	Se creó el resumen.



## *Capítulo 2: Características del Sistema.*

---

Con el desarrollo de este capítulo se concluye con una perspectiva del sistema que se desea desarrollar, en términos de requerimientos funcionales y no funcionales. Para ello se obtuvo el modelo del negocio, definiéndose actores, trabajadores y procesos en el mismo, con el fin de satisfacer al cliente.

### Capítulo 3: Diseño del Sistema

En este capítulo se realiza el diseño de la propuesta de solución, definiéndose los diagramas de clases que se producen como parte del flujo de procesos, lo que permite un mejor entendimiento y una mayor funcionalidad del sistema que se propone.

#### 3.1 Arquitectura

La arquitectura de software se define como la estructura de los componentes de un programa o sistema, sus interrelaciones, los principios y reglas que administran su diseño.

Para la realización del sistema “Módulo de Nutrición del Sistema de Evaluación del Neurodesarrollo en Niños” se propone el uso de los patrones arquitectónicos Modelo Vista Controlador (MVC) y en capas, debido a que son los más utilizados y a su vez permiten separar los elementos de la presentación, el negocio y el acceso a datos, para lograr que cada capa se comuniquen con sus adyacentes, permitiendo que los cambios de una capa puedan realizarse sin afectar los restantes niveles.

El patrón MVC se evidencia de la siguiente forma, la Vista se corresponde con las páginas XHTML las cuales son interfaces de usuario que le presenta el sistema a este, manejan las acciones realizadas sobre la interfaz por el usuario y recogen la información entrada por este. El controlador se corresponde con las clases controladoras para cada caso de uso, que se encargan del procesamiento de la información en correspondencia con la lógica del negocio en cuestión. Y la información manejada en todo el sistema coincide con el modelo el cual es una representación orientada a objetos, en forma de clases de entidad, de las tablas de la base de datos del sistema.

#### 3.2 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema.

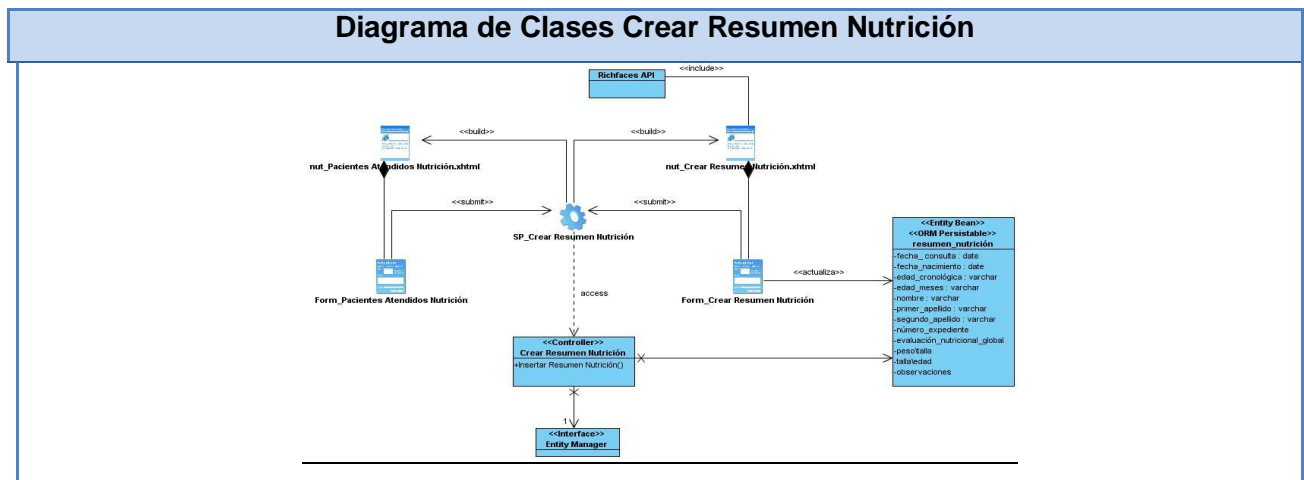
Además, describe las clases más importantes, su organización en paquetes y subsistemas, sirviendo como abstracción de la implementación y utilizándose como entrada fundamental de las actividades de implementación.



Estos diagramas son importantes para visualizar, especificar, documentar modelos estructurales y construir sistemas ejecutables aplicando ingeniería directa e inversa.

Cada caso de uso cuenta con un diagrama de clases del diseño y pueden organizarse en paquetes o subsistemas para hacer más fácil su comprensión.

A continuación se muestran el diagrama de clases del diseño Crear Resumen Nutrición. Los demás pueden ser vistos en el SVN del Área Temática: Sistemas Especializados.



En este capítulo se realizaron los diagramas de clases del diseño. Se pudo definir las clases más significativas del mismo, con sus atributos y métodos, permitiendo al implementador un mejor entendimiento del diseño del sistema.

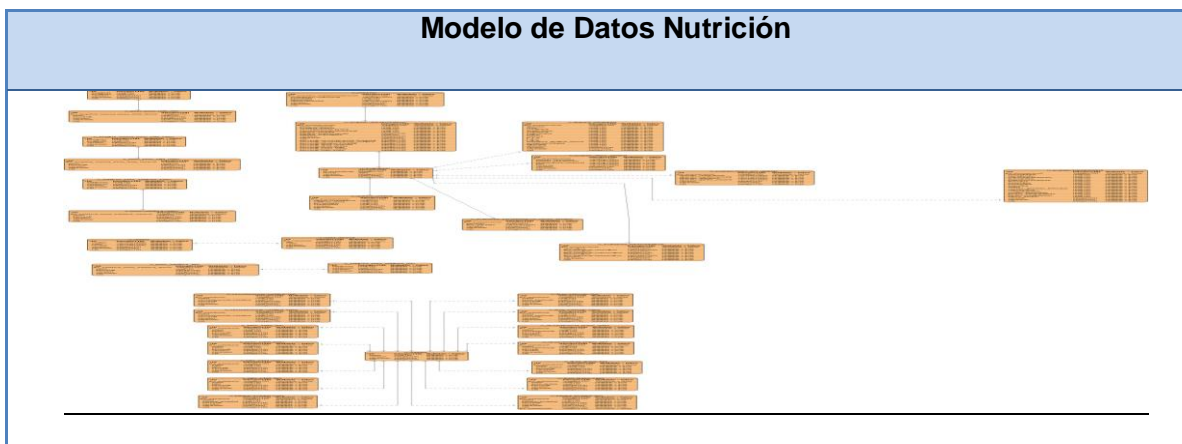
## Capítulo 4: Implementación.

En el presente capítulo se modelan los artefactos correspondientes al flujo de trabajo: Implementación. Se explica la seguridad del sistema, así como los estándares de codificación que se utilizan. Además, se describe el Diagrama de Componentes y el Diagrama de Despliegue, ambos diagramas conforman el Modelo de Implementación.

### 4.1 Modelo de datos

Es el modelo que describe de manera abstracta cómo se representan los datos de una aplicación o sistema de información. Consiste en una descripción de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores.

El Modelo de Datos tiene gran importancia en el ciclo de desarrollo de software, y de manera particular para la fase de implementación, pues define formalmente las estructuras permitidas y las restricciones que se aplican con el fin de representar los datos del dominio de la aplicación. Está compuesto por objetos: entidades que existen y se manipulan; y atributos: características básicas de dichos objetos y relaciones: forma en que se enlazan los objetos entre sí. (16)



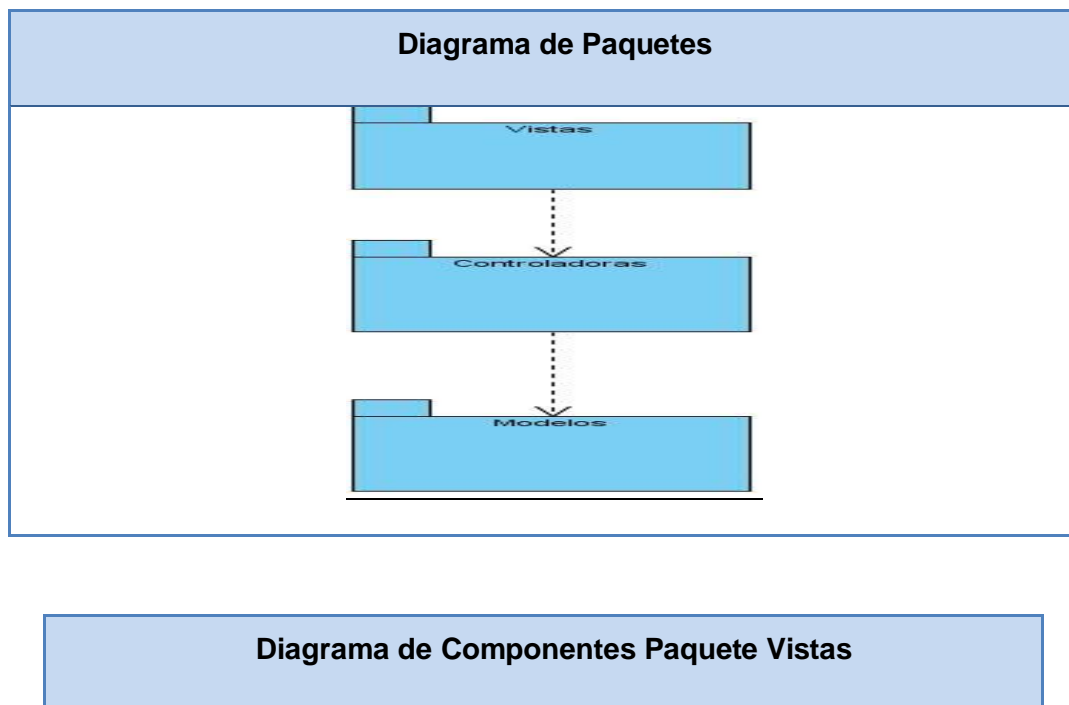
### 4.2 Modelo de implementación

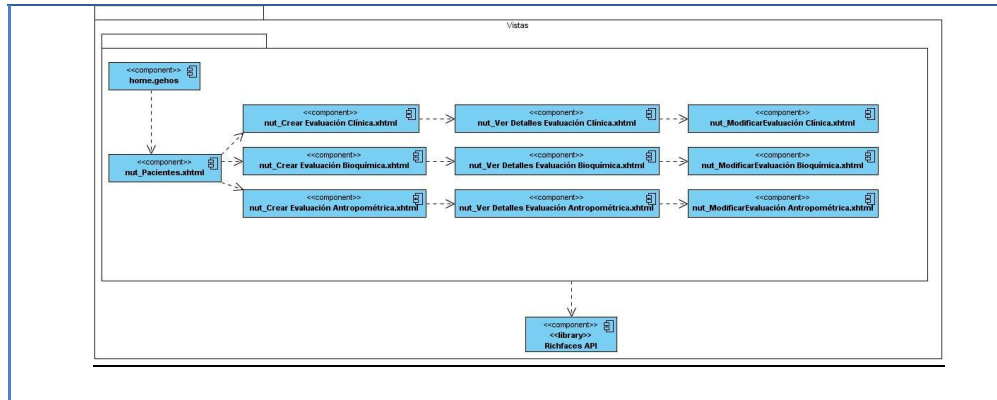
Está conformado por los Diagramas de Componentes y de Despliegue, describiendo cómo los elementos del Modelo de Diseño se implementan en términos de componentes, ficheros de código fuente y ejecutables. El Modelo de Implementación describe además cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización, disponibles en el entorno de

implementación y en el lenguaje de programación utilizado, y cómo dependen los componentes unos de otros. (17)

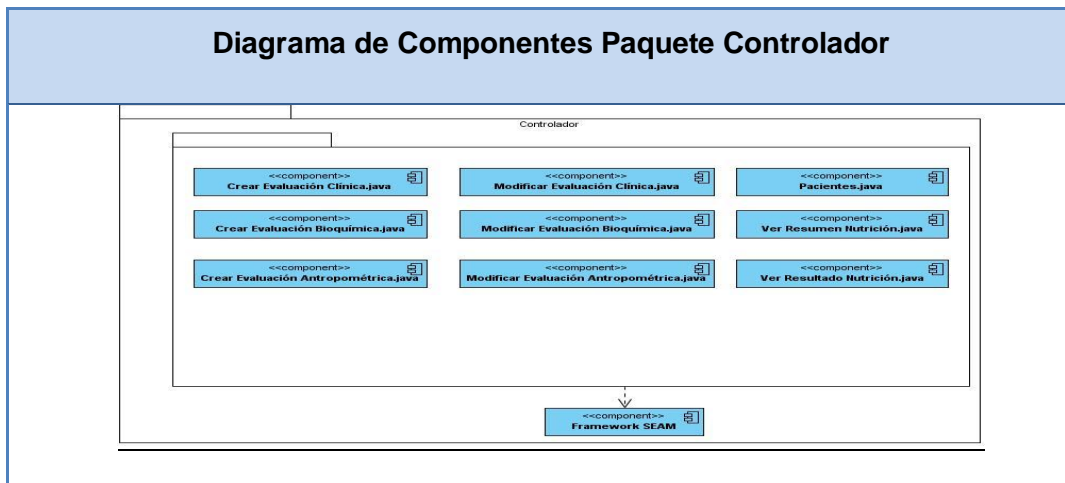
### 4.2.1 Diagrama de componentes.

Es una representación gráfica que muestra un conjunto de elementos del modelo tales como componentes, subsistemas o paquetes de implementación y sus relaciones. Permite modelar la vista estática del sistema mostrando las dependencias lógicas entre un conjunto de componentes de software. Este diagrama se estructura en paquetes, que son divisiones físicas del sistema. Los paquetes están organizados en una jerarquía de capas donde cada capa tiene una interfaz bien definida. Los componentes pueden ser de código fuente, librerías, binarios o ejecutables y tienen relaciones de traza con los elementos del modelo que implementan. En el Diagrama de Componentes elaborado se evidencian las relaciones existentes entre los paquetes y los componentes. Dichas relaciones son de inclusión y pueden ser identificadas por el estereotipado <<import>>. (18)

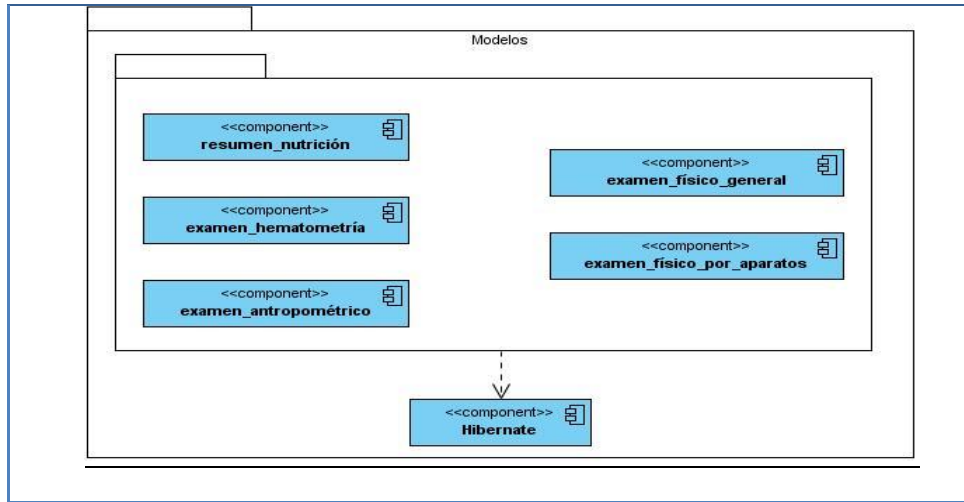




### Diagrama de Componentes Paquete Controlador



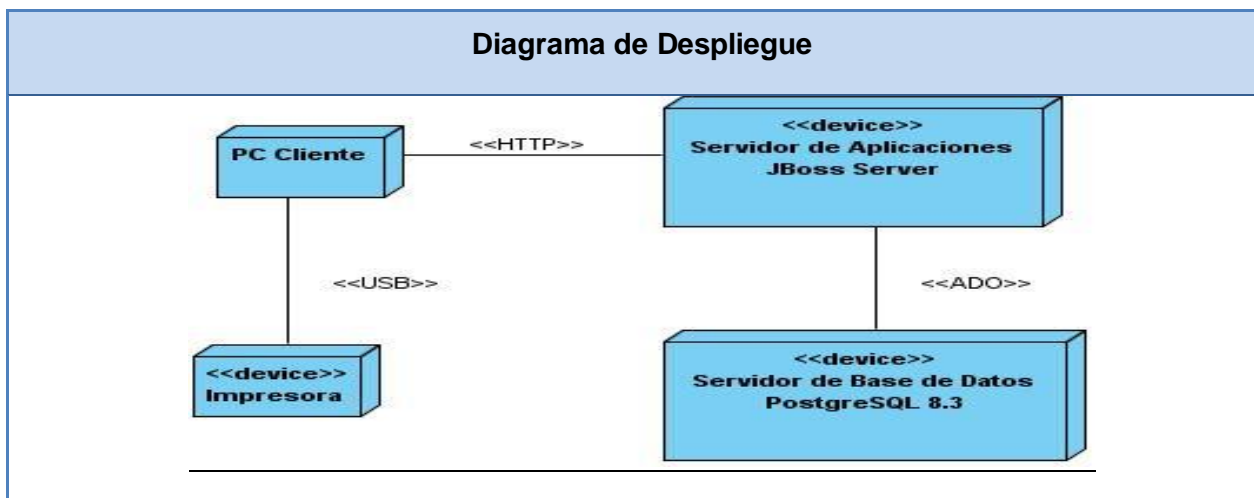
### Diagrama de Componentes Paquete Modelo



## 4.2.2 Diagrama de despliegue.

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación.

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.





El módulo de Nutrición contará con dos servidores, uno de aplicaciones (web) y otro de base de datos, conectados entre sí mediante el Protocolo de Control de Transmisión/ Protocolo de Internet (TCP/IP, por sus siglas en inglés). Además contará con las estaciones clientes que se comunicarán con el servidor web mediante el Protocolo de Transferencia de Híper Texto (HTTP, por sus siglas en inglés).

### 4.3 Seguridad

La seguridad es un tema de gran relevancia para cualquier Sistema de Información y toma mayor importancia cuando se gestiona información relacionada con la salud de las personas.

En el caso del sistema en cuestión, la seguridad es llevada a cabo por el proyecto alas-HIS. Para esto definieron diferentes tipos de seguridad: acceso al sistema, registro de trazas, administración de seguridad (vista lógica y vista física) y configuración de funcionalidades.

**Acceso al sistema:** Se definieron diferentes roles que permiten un nivel de acceso distinto para cada uno de ellos, los cuales podrán entrar a los diferentes módulos y podrán usar las funcionalidades de acuerdo a los permisos de su rol, realizando todo esto a través de un usuario y contraseña.

**Registro de trazas:** Se registran en la base de datos acciones llevadas a cabo por los usuarios como: inicio o cierre de sesión, acceso a un módulo, modificación de un atributo, entidad o cualquier otra operación sobre el sistema.

**Administración de seguridad:** El sistema brinda la posibilidad de asignar o denegar permiso a los diferentes roles y usuarios, en los módulos y funcionalidades dentro de estos y también, la eliminación de roles y usuarios de las listas de los que se le negó o permitió algún permiso. Todos estos permisos son registrados por el sistema.

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

En todas las capas de la aplicación se lleva a cabo la seguridad. Para garantizar la seguridad toda la autorización está basada en reglas que restringen el acceso a directorios, páginas, controles, opciones del menú y servicios del negocio.

Ninguna de estas reglas del negocio está embebida en el código de la aplicación, posibilitando que en caso de ocurrir un cambio en alguna de estas, no se requiera el cambio ni la respectiva recompilación del código, sino solo en la definición de la propia regla en su fichero de configuración. Esto puede llevarse a cabo por la posibilidad de integración con el motor de reglas JBoss Rules, que brinda el Framework de Seguridad de JBoss Seam.

### **4.4 Estándares utilizados.**

Los estándares han sido la herramienta base de la interoperabilidad informática. Son los que han permitido definir cómo interactuarán los miles o millones de componentes informáticos que existen. La especificación de un estándar, a su vez, es aquel conjunto de documentos donde se define cómo llevar a cabo un desarrollo de software o hardware que siga ese estándar.

#### **4.4.1 Estándares de Diseño.**

Son ciertas pautas que se establecen para conseguir uniformidad en el desarrollo de un sitio web. El diseño de una interfaz eficaz va a estar dado también por la posibilidad de que diferentes usuarios con posibilidades diversas tengan el acceso a los contenidos que un sitio ofrece (diseño web centrado en el usuario), sin que esto conlleve a tener más de un diseño web para cada tipo de visitante.

El Módulo de Nutrición del Sistema de Evaluación del Neurodesarrollo en Niños sigue las pautas de diseño del Sistema de Información Hospitalaria (HIS). Estas pautas permiten lograr una mayor efectividad en el proceso de trabajo al existir una mayor coherencia formal entre los módulos y páginas del sistema, y que éstos sean identificados como parte de un todo. Se han pautado una serie de elementos comunes que facilitarán su reconocimiento y el uso que se haga de ellos.

#### **General**






- ✓ El género de todos los actores que aparezcan en el prototipo será masculino.
- ✓ El formato de la fecha es dd/mm/yyyy, en el caso de la hora es hh:mm a.m. /p.m., en el caso de que se requieran ambos campos se manipularán por separado, es decir, se tratarán como dos atributos diferentes. En ambos casos se utilizará para ello el rich: calendar.
- ✓ En el título de los formularios se pondrá el nombre de la acción en infinitivo, la capitalización es igual a la de las etiquetas. Ejemplo: crear resumen, realizar evaluación.

- ✓ No se podrán utilizar datos pertenecientes a miembros del grupo de desarrollo como juego de datos.
- ✓ En el caso de usar un componente que permita la selección y deselección entre elementos de dos listas se utilizará el list shuttle no poniéndose etiquetas en los íconos de acción (>>, >, <, <<).
- ✓ Todos los datos de muestra que se pongan en el prototipo cumplirán con las mismas reglas de capitalización que las etiquetas.
- ✓ En los casos en que se requiera seleccionar un rango de fechas éste se pondrá mediante las etiquetas desde: y hasta: asociadas a los calendarios correspondientes. En ambos casos se utilizará para ello el rich: calendar.
- ✓ El simpletogglepanel de asociado a los criterios de búsqueda o de criterios generación de los reportes se titulará “criterios de búsqueda”, apareciendo inicialmente en la página los componentes asociados a los parámetros de búsqueda y los botones correspondientes, una vez que se presione el botón de buscar o generar entonces se mostrará la tabla con los resultados de la búsqueda.

### Mensajería

- ✓ El texto del mensaje será el que está identificado en el documento “Pautas de Mensajes del Sistema.doc” y se mostrará alineado a la izquierda.
- ✓ Los botones de los mensajes deben cumplir las mismas reglas que la sección de Botones.
- ✓ En las ventanas de Información y Error se mostrará solo un botón de “Aceptar”, centrado en la parte inferior.
- ✓ En las ventanas de Advertencia se utilizarán dos botones centrados en la parte inferior y contendrán los textos: “Sí” y “No”. El botón “Sí” estará a la izquierda y el “No” a la derecha.
- ✓ A la izquierda del mensaje aparecerá una imagen que sería el ícono que indica el tipo de mensaje (advertencia (⚠), información (💡) y error (❌)).
- ✓ En el título de la ventana aparecerá el tipo de mensaje que se muestra.
- ✓ Los botones que llevarán estos mensajes son los que se describen en “pautas de mensajes del sistema.doc” en la sección de acciones.

### Tablas

- ✓ Las tablas aparecerán centradas con los colores definidos en el estilo CCS.
- ✓ El nombre de la tabla aparecerá en texto alineado a la izquierda y en negrita. Cumpliendo las mismas reglas de capitalización que las etiquetas.
- ✓ Las tablas que representen listados de pacientes, medicamentos, productos o cualquier otra entidad su nombre comenzará precedido por el texto Listado de <entidad>.
- ✓ El nombre de la columna aparecerá alineado a la izquierda y en negro (R0 G0 B0 - #000000). Cumpliendo las mismas reglas de capitalización que las etiquetas. Se le ubicará el título de “Foto” a la columna en la que se muestre la foto del paciente, no así en las columnas donde se ubiquen los íconos de: “Ver” () , “Modificar” () , “Eliminar” () , “Seleccionar” () , Crear () , donde el encabezado de la columna quedará en blanco.
- ✓ El contenido de la tabla será alineado a la izquierda. Excepto en las columnas que se muestren datos numéricos que el contenido se alineará a la derecha.
- ✓ Las tablas resultantes de una búsqueda utilizarán columnas diferentes para cada uno de los atributos a mostrar. Los campos de fecha y hora siempre irán en columnas diferentes igualmente sucederá con los campos asociados a los datos del paciente, leyendo los atributos por columnas. En el caso de mostrar el nombre y apellidos de los actores del sistema se mostrarán todos en una misma columna.
- ✓ En el caso de los atributos que permitan valores nulos y que no posean valor alguno en la Base de Datos no se mostrará nada en la celda asociada al mismo en los reportes, es decir, se dejará en blanco.
- ✓ Para todos los reportes se debe definir un orden homogéneo para los parámetros de salida, evitando que reportes que muestran información común visualmente los atributos se muestren en un orden diferente, pudiera coincidir con el orden en que se muestran los criterios de búsqueda.
- ✓ En el caso de reportes que no lleven parámetros de búsqueda no se colocará un botón de generar innecesario después de seleccionar el reporte en el Menú, sino que a partir de esta acción se obtendrá directamente el reporte sin la acción del botón de generar intermedia.
- ✓ En el caso de reportes con criterios de búsqueda contendrá los botones de Generar y Cancelar debajo de los criterios de búsqueda, en el caso de que no posea parámetros de búsqueda los

botones serán Imprimir y Salir, en caso de que el reporte sea imprimible y solo Salir en caso de no poseer esta acción.

### **Botones**

- ✓ Los colores están definidos en el estilo CSS.
- ✓ El tamaño será el estándar del IDE.
- ✓ Aparecerán alineados en la parte inferior a la derecha y el orden será de tal forma que las acciones positivas al flujo sea de izquierda a derecha.
- ✓ El nombre de los botones tiene que cumplir las reglas de las “Pautas de Etiquetado.doc”. Por lo que solo se pueden usar textos que se muestran en este juego de etiquetas, es decir, solo las acciones especificadas en este documento.
- ✓ El orden de los botones será de máxima a mínima prioridad de izquierda a derecha, es decir, a la izquierda siempre estará el botón de mayor prioridad.
- ✓ La ubicación de la sección de botones será la que se muestra teniendo en cuenta los siguientes casos, las etiquetas no tendrán separación alguna con el componente asociado. (Así debe quedar la organización de los componentes en las interfaces, como se observa la búsqueda avanzada va debajo de la línea de botones y los casos en que los botones van debajo de los componentes estos deben ir alineados a la derecha al último componente. Lo más importante es la separación en píxel dentro de la interfaz, entre el header del panel y la primera línea de componentes 13 píxeles, la primera línea de componentes estará a 8 píxeles del lado izquierdo del panel, entre la primera columna de componentes y la segunda 25 píxeles).

### **Editores de Texto.**

- ✓ Tendrán tamaño estándar del IDE.
- ✓ Se pondrán alineados a la izquierda.
- ✓ Los editores asociados con etiquetas se pondrán debajo de la etiqueta, alineados ambos a la izquierda.

### **Etiquetas.**

- ✓ El tamaño y el color están definidos en el estilo CSS.

- ✓ Todas las etiquetas estáticas irán en negro (**R0 G0 B0 - #000000**). La clase de estilo a utilizar es la siguiente:

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

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

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

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

### Selectores

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

- ✓ En los selectores donde se permita seleccionar todos los elementos asociados a este se colocará una opción de Todos o Todas en dependencia del género del campo asociado.
- ✓ El orden de las opciones será: <Selecione>, Todos o Todas (en caso de ser necesario) y el resto de las opciones en orden alfabético ascendente.

### Secciones

- ✓ Serán nombradas con un texto que sea afín con la sección el cual no terminará en ningún signo de puntuación.
- ✓ El nombre estará alineado a la izquierda, Verdana 12 en negrita, no terminando ni en (.) ni en (:).
- ✓ Se utilizarán los paneles.
- ✓ El tamaño será a 100% del espacio asignado.
- ✓ Los nombres cumplirán con la misma capitalización que las etiquetas.
- ✓ Los paneles contenedores tendrán tamaño variable el cual se ajustará en la medida de la cantidad de componentes visuales que contendrá.

#### 4.4.2 Estándares de codificación

Los estándares de codificación permiten que todos los desarrolladores de un software, puedan entender en menos tiempo un estilo de programación homogéneo en un proyecto y que el código en consecuencia sea mantenible.

Para definir el estilo de codificación a seguir en la aplicación se utilizó la notación estándar establecida para las aplicaciones desarrolladas en JAVA (Java Code Conventions), definido en las especificaciones del propio lenguaje.

### Nomenclatura

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

### Archivos fuente

Cada programa en Java es una colección de uno o más archivos. El programa ejecutable se obtiene compilando estos archivos. En cada archivo especifica su contenido como sigue:

1. Los paquetes (instrucción package).
2. Los archivos de biblioteca (Instrucciones import).
3. Un comentario explicando el objetivo del archivo.
4. Las clases que defines en ese archivo.

### Clases

Cada clase debe ir precedida por un comentario que explique su objetivo. Es recomendable especificar sus elementos como sigue:

1. Estructura de los objetos. Primero las variables y luego las constantes.
2. Elementos estáticos.
3. Constructores.
4. Métodos públicos y privados.
5. Métodos estáticos.
6. Clases internas.

Dejar una línea en blanco después de cada método.

Todos los elementos deben estar precedidos por public, private o protected. Las variables deben ser privadas. Los métodos y las constantes pueden ser privados o públicos, según se requiera.

### Tamaño de Líneas

Las líneas deben ser de máximo 99 caracteres. Si es necesario partir la línea, la siguiente línea debe alinearse dejando doble sangría.

### Generalidades

Se empleará el formalismo de precondition, poscondición e invariante.



El código debe comentarse utilizando la sintaxis apropiada para uso de javadoc, teniendo en cuenta que para la producción de la documentación deben incorporarse los tags particulares que no hagan parte del estándar.

En el caso de las poscondiciones, se utilizará la notación prima ( $x'$ ) para referirse al valor de una variable al finalizar el método, y sin prima ( $x$ ) para su valor antes de iniciarse la ejecución de la rutina. Por ejemplo una rutina que incrementa el valor de  $x$  en uno puede especificarse con: `@pos x' == x + 1`. Los conectores lógicos que se deben emplear son los de Java: `&&`, `||`.

Se asume que los atributos que no figuran en la poscondición, no sufren modificaciones. En el caso de funciones analizadoras, éstas no deben afectar atributos, y su poscondición se especifica con el tag `@return` de javadoc.

### **Sangría y Ablocamiento**

Todos los archivos fuentes deben seguir el estándar de sangría. Cada entrada corresponde a 4 espacios. No debe usarse el caracter de tabulación, pues es dependiente de la configuración del editor. Todos los constructores que admiten bloques de instrucciones delimitados por `{...}`, deben usarlos aún si éstos tienen una sola instrucción.

Las funciones deben tener un solo punto de retorno, y el flujo normal de las estructuras de iteración no debe alterarse mediante instrucciones `break` o `continue`.

### **Aserciones**

Se recomienda el uso de aserciones para validación de poscondiciones, invariantes y otros puntos críticos según sea el caso. Para el caso del invariante de la representación se recomienda la implantación del método `invarainteOk`, de modo que se pueda invocar desde las aserciones de poscondición de todos los métodos.

Con este capítulo se ha concluido el proceso de Implementación y se obtuvo los artefactos que conforman el Modelo de Implementación, Diagrama de Despliegue y Componentes. Además, se refleja el uso de los estándares de diseño y codificación, así como de la seguridad del sistema.

### Conclusiones

- ✓ Como resultado de la investigación realizada se encontraron diferentes sistemas automatizados que se encargan de evaluar a los niños con problemas en el neurodesarrollo, tanto a nivel internacional como nacional. El estudio de dichos sistemas permitió determinar que éstos no cumplen con las características del Programa de Atención y Evaluación del Neurodesarrollo llevado a cabo en el Hospital Pediátrico “William Soler” de Cuba.
- ✓ El Sistema de Evaluación del Neurodesarrollo en Niños pertenece al Sistema de Información Hospitalaria por lo que se adoptó la arquitectura definida por el mismo y se utilizaron las herramientas, tecnologías y metodologías que esta propone.
- ✓ Como consecuencia del estudio del funcionamiento del Programa de Atención y Evaluación del niño en la especialidad de Nutrición, se identificaron los principales procesos del negocio y se modelaron los flujos de trabajo propuestos por el Proceso Unificado de Desarrollo: Modelamiento del Negocio, Gestión de Requerimientos, Diseño e Implementación, obteniéndose los artefactos que se generan en cada uno de ellos.
- ✓ Se realizó la implementación del Módulo de Nutrición del Sistema de Evaluación del Neurodesarrollo en Niños con todas las funcionalidades requeridas por el cliente, lo que permite una mayor eficiencia en la realización de las evaluaciones y éstas a su vez agilizarán la evaluación de los niños en la consulta de Nutrición.

**Recomendaciones**

Los autores del presente trabajo recomiendan:

- ✓ Incluir la Evaluación Dietética en el sistema para que se le pueda proponer una dieta balanceada al paciente dado sus requerimientos nutricionales.

### REFERENCIAS BIBLIOGRÁFICAS

1. **Atención Temprana.** Disponible en: [www.sld.cu/sitios/rehabilitacion-temprana/temas.php](http://www.sld.cu/sitios/rehabilitacion-temprana/temas.php)
2. **Organización Panamericana de la Salud.** Software para la salud: Disponible en: [http://www.nutrinfo.com/pagina/programa\\_nutricion.php](http://www.nutrinfo.com/pagina/programa_nutricion.php)
3. **Organización Panamericana de la Salud.** Software para la salud: Disponible en: [http://www.nutrinfo.com/pagina/programa\\_nutricion.php](http://www.nutrinfo.com/pagina/programa_nutricion.php)
4. **Reynoso, Carlos; Kicillof, Nicolás.** Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Universidad de Buenos Aires. 2004. Disponible en: [http://www.microsoft.com/spanish/msdn/arquitectura/roadmap\\_arq/style.asp#10](http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp#10)
5. **Arquitectura de programación en 3 capas.** <http://www.elcodigok.com.ar>. [En línea]. [Citado el: 25 de noviembre del 2009]. Disponible en: <http://www.elcodigok.com.ar/2007/09/arquitectura-de-programacion-en-3-capas/>.
6. **Conceptos Básicos de Computación.** [En línea] [Citado el: 20 de noviembre del 2009.] <http://haideeperez75.blogspot.com/feeds/posts/default/3466693163993704565>.
7. **Pons Capote, Olga, y otros.** Introducción a las Bases de Datos. El Modelo Relacional. s.l. : Thomson Learning Ibero, 2005. ISBN 8497323963.
8. **Ingeniería de Software 1. “UML y RUP.”** UCI. curso 2008\_2009. Clase Teórico Práctica # 1.
9. **Ingeniería de Software 1. “UML y RUP.”** UCI. curso 2008\_2009. Clase Teórico Práctica # 1.
10. **Presman Roger S.** Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
11. **Presman Roger S.** Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
12. **Presman Roger S.** Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.

13. **Geoffrey Sparks, Sparx Systems.** Una Introducción al UML. El Modelo de Casos de Uso. [www.sparxsystems.com.ar](http://www.sparxsystems.com.ar) - [www.sparxsystems.cl](http://www.sparxsystems.cl).
14. **Presman Roger S.** Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
15. **Ingeniería de Software 2.** “Continuación del FT Análisis y Diseño. Modelo de Diseño.” UCI. Curso 2008\_2009. Conferencia # 1.
16. **Presman Roger S.** Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
17. **Presman Roger S.** Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
18. **Presman Roger S.** Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.

## BIBLIOGRAFÍA

- **Arquitectura de programación en 3 capas.** <http://www.elcodigok.com.ar>. [En línea]. [Citado el: 25 de noviembre del 2009]. Disponible en: <http://www.elcodigok.com.ar/2007/09/arquitectura-de-programacion-en-3-capas/>
- **Atención Temprana.** Atención Temprana en la Provincia de Cienfuegos. [En línea] [Citado el: 10 de noviembre del 2009]. Disponible en: [www.sld.cu/sitios/rehabilitacion-temprana/temas.php](http://www.sld.cu/sitios/rehabilitacion-temprana/temas.php)
- **Atención Temprana e Intervención en niños con Trastornos Generalizados en el desarrollo.** Departamento de Psicología Evolutiva y de la Educación. [Citado el: 10 de diciembre del 2009] <http://www.acceso.av.es>
- **Ayuda extendida del Rational Rose Enterprise Edition 2003.**
- **Booch, G., Rumbaugh, J. y Jacobson, I.** El Lenguaje Unificado de Modelado. 2000.
- **Booch, Jacobson, I, G y Rumbaugh, J.** “El Proceso Unificado de Desarrollo de Software”. La Habana : Félix Varela, 2004. Vol. 1.
- **JOSÉ EMILIO LABRA GALLO, R. M.** Curso: Tecnología Web, [2007]. [Citado el: 12 de febrero del 2009] Disponible en: <http://www.di.uniovi.es/~labra/cursos/masterMTI/PDF/ServWeb.pdf>
- **MARIN D. M. E.** Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. La Habana. 2006.
- **Plataforma De Aplicaciones Jboss Enterprise.** [Citado el: 10 de marzo del 2010] [www.redhat.es/jboss](http://www.redhat.es/jboss)
- **PRESSMAN, R. S.** *Ingeniería del Software.* Un enfoque práctico. Ciudad de La Habana: 2005.Vol. 1
- **Revista Cubana de Informática Médica.** Revista Cubana de Informática Médica. [En línea] [Citado el: 15 de abril del 2010] [http://www.cecam.sld.cu/pages/rcim/revista\\_15/articulos.htm](http://www.cecam.sld.cu/pages/rcim/revista_15/articulos.htm)
- **Stevens, P. y Pooley, R.** Utilización de UML en ingeniería del software.

- **Software para el tratamiento terapéutico de niños con problemas cerebrales.** Grupo de investigación en Computación Gráfica y Procesamiento de Imágenes (OHWAHA).
- **The PostgreSQL Global Development Group.** About [En línea] [Citado el: 15 de abril del 2010]  
<http://www.postgresql.org/about/>
- **Tutorial sobre JBoss Seam:** [Citado el: 15 de abril del 2009]  
<http://docs.jboss.com/seam/2.0.1.GA/reference>

## Glosario de términos

- ✓ **Atención Temprana:** Permite conjuntamente con la familia y la comunidad ofrecer a los niños con déficit en su neurodesarrollo, un conjunto de acciones optimizadoras y compensadoras que faciliten su adecuada maduración en todos los ámbitos y que les permita alcanzar el máximo nivel de desarrollo personal y de integración social.
- ✓ **Framework:** En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.
- ✓ **MVC:** (*Modelo Vista Controlador*). Es un patrón de arquitectura de software compuesto de tres componentes distintos: datos, interfaz de usuario, y lógica del negocio.
- ✓ **Neurodesarrollo:** Adquisición de funciones, dependientes del sistema nervioso, que implican un incremento de estructuras orgánicas y funcionales a través de un proceso de maduración.
- ✓ **Nutrición:** Proceso biológico en el que los organismos asimilan los alimentos y los líquidos necesarios para el funcionamiento, el crecimiento y el mantenimiento de sus funciones vitales. También es la ciencia que estudia la relación que existe entre los alimentos y la salud, especialmente en la determinación de una dieta.
- ✓ **ORM:** (*Object-Relational Mapping*) El mapeo de objetos-relacional es una técnica de programación utilizada para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos, creando una base de datos orientada a objetos virtuales, por encima de la base de datos relacional.
- ✓ **Proceso:** Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí, que a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido.
- ✓ **SENDN:** Sistema de Evaluación del Neurodesarrollo en Niños.
- ✓ **Subproceso:** Son partes bien definidas en un proceso. Su identificación puede resultar útil para aislar los problemas que pueden presentarse y diferentes tratamientos dentro de un mismo proceso.
- ✓ **UCIN:** Unidades de Cuidados Intensivos Neonatales.
- ✓ **UCIP:** Unidades de Cuidados Intensivos Polivalentes.