

Universidad de las Ciencias Informáticas

Facultad 7



Título: *Módulo de Fisiatría del Sistema de Evaluación del
Neurodesarrollo en Niños*

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

Autores: *Solainy Fajardo Araujo*

Maikel Guerra Ferrer

Tutor: *Ing. Miguel Ángel Fernández Marín*

Ciudad de La Habana, Junio 2009

“Año del 50 aniversario del triunfo de la Revolución”



“Todos los días hay que luchar porque ese amor a la humanidad viviente se transforme en hechos concretos.”

Ernesto Guevara de la Serna

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos 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 29 días del mes de junio del año 2009.

Solainy Fajardo Araujo

Firma del Autor

Maikel Guerra Ferrer

Firma del Autor

Ing. Miguel Ángel Fernández Marín

Firma del Tutor

DATOS DE CONTACTO

Ingeniero: Miguel Ángel Fernández Marín.

Universidad de Ciencias Informáticas, Habana, Cuba.

E-mail: mafernandez@uci.cu

Ingeniero: Malena Guzmán Pérez

Universidad de Ciencias Informáticas, Habana, Cuba.

E-mail: mguzman@uci.cu

AGRADECIMIENTOS

De Solainy

Un agradecimiento muy especial a nuestro Comandante en Jefe y a la Revolución Cubana por darnos la oportunidad de estudiar en esta Universidad de Excelencia.

A mis padres, por haber sido fuente de inspiración y amor, por depositar toda su confianza en mí, por brindarme su apoyo incondicional, por ser mi ejemplo y guía.

A mis hermanos Solange y Javielito gracias por ser como son.

A mis abuelos por toda su paciencia y dedicación y a mis tías que siempre me dieron aliento para seguir adelante.

A toda mi familia por lo mucho que me quieren y por toda la ayuda que me han brindado.

A mis amigos, a los viejos y a los nuevos, quienes me ayudaron, apoyaron y aconsejaron en los momentos más difíciles, con los que compartí una larga etapa de la vida. A mi compañero de tesis, Maikel, por todo su apoyo. Ya lo logramos.

A Yarel por todo su apoyo y comprensión, a Yani, Lili, Yalinet y Javier por estar conmigo en todo momento. A Lisi por ser mi otra compañera de tesis. Los quiero mucho.

A la profesora Malena por toda su comprensión, ayuda incondicional y dedicación en todo este año, muchas gracias por todo.

A mi tutor por guiarme en esta investigación, por sus sugerencias siempre bien recibidas. Al tribunal por todos sus consejos, en especial a la profesora Niuris. Muchas gracias por todo.

A mis profesores, que a lo largo de toda mi vida estudiantil han contribuido a mi formación profesional.

AGRADECIMIENTOS

De Maikel

Antes que nada, dar las gracias a todas las personas que de una forma u otra me han ayudado a alcanzar este logro personal. A todos aquellos que en su momento me brindaron su ayuda y su apoyo, de corazón, gracias.

Agradecer al profesor Luis Mariano, mi primer jefe de proyecto, por toda su confianza y su ayuda. A la profesora Malena por todo su apoyo y consagración, todo esto en parte es gracias a ella, a Dunior por confiar en mi y en mi trabajo. Al profesor Alejandro Mario, por toda su ayuda.

Dar gracias a todos mis compañeros de aula, los que vienen conmigo desde 1er año y los que se han ido incorporando en el transcurso de los 5 años. A mi compañera de tesis Solainy, por todo el trabajo que hizo y por todo su apoyo. A los muchachos de hospitales, que prácticamente sin conocerme, me brindaron su ayuda.

A mis amigos Argente, Ernesto, Gallego, Foster, el Bolo, Jose, que hoy no están aquí conmigo, pero igual siempre me han ayudado y apoyado, a todos ellos, muchas gracias.

Agradecer al Chulo, al Charro y a Jorge por estar siempre ahí cuando los necesité, por todo su cariño, su apoyo y su ayuda, gracias de corazón. A Cheo, Bechito y Elen, que los quiero a todos como si fueran mis padres. A mi gente de la Biplanta, Olga, Rolando y Nani, gracias por todo su apoyo y su ayuda.

A Yunielys por todo el amor y el cariño que me ha brindado a mí y a mi familia. Gracias, por quererme tanto.

A toda mi familia, que siempre me han apoyado. A mis padres y a mi hermana, por todo el amor y cariño que siempre me han brindado, por siempre estar cuando los necesité y por guiarme. A ellos mi mayor agradecimiento.

DEDICATORIA

De Solainy

A mis padres, por apoyarme en todo momento, por ser los mejores del mundo. Quisiera decirles tantas cosas y no encuentro las palabras para expresar todo lo que siento; pero quiero que sepan que todo lo que soy se los debo a ustedes.

A mi abuelita Lolo, por depositar toda su confianza en mí, apoyarme y guiarme en todo momento. TE quiero mucho.

A mis hermanos por estar conmigo en todo momento, esto también es de ustedes.

A mi niña linda, mi sobrinita. TE adoro.

De Maikel

A mis padres, por ser tan especiales, por darme la vida y educarme como lo han hecho. Los quiero mucho.

RESUMEN

En el Hospital Pediátrico William Soler se lleva a cabo un programa de atención y evaluación del neurodesarrollo en niños, en el cual las pruebas aplicadas por los especialistas en fisiatría son muy extensas y realizadas de forma manual. Las mismas requieren de mucho tiempo e impiden que un mismo día puedan atender a gran cantidad de niños. El presente trabajo se propone desarrollar una aplicación web que facilite el proceso de gestión de la información relacionada con los niños egresados de terapia intensiva, reportados como críticos durante su estadía.

Para el desarrollo de la aplicación se emplean como herramientas a utilizar PostgreSQL de Gestor de Base de Datos, el Lenguaje de Programación Java, la Metodología de Desarrollo de Software el Proceso Unificado de Desarrollo (RUP) para especificar, construir y documentar el sistema. Se hace uso del Lenguaje Unificado de Modelado (UML) 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).

La puesta en marcha de esta aplicación permitirá una mayor eficiencia en la realización de las pruebas del Peabody Motor Grueso y Fino agilizando la evaluación de los niños en la consulta de fisiatría.

Palabras Claves: Neurodesarrollo, Peabody, Framework, Software.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Antecedentes de la Atención Temprana	5
1.2 Conceptos Básicos.....	6
1.3 Sistemas Automatizados existentes	7
1.4 Metodología de Desarrollo.....	10
1.5 Tecnologías	11
1.5.1 Patrones de arquitectura y diseño	11
1.6 Lenguajes.....	18
1.7 Servidor de aplicaciones.....	19
1.8 Sistema Gestor de Bases de Datos	20
1.9 Herramientas a utilizar.....	22
CAPÍTULO CARACTERÍSTICAS DEL SISTEMA.....	25
2.1 Flujo actual de los procesos involucrados en el campo de acción.....	25
1.5.2 Descripción de los Procesos de Negocio	25
1.5.3 Análisis crítico de ejecución de los procesos actuales	27
1.5.4 Objeto de Automatización	27
2.4 Modelo del negocio.....	28
1.5.1 Identificación de roles del entorno del negocio	28
1.5.2 Diagrama de proceso del negocio	29
2.5 Especificación de los requisitos de software.....	36
1.5.1 Requerimientos Funcionales.....	36
5.2.1 Requisitos no funcionales	39
5.3 Modelo de casos de uso del sistema	44
5.3.1 Diagrama de Casos de Uso del Sistema	45
5.3.2 Listado de Casos de Usos	46
5.3.3 Casos de uso por ciclos	60
5.3.4 Casos de uso expandidos	64
CAPÍTULO ANÁLISIS Y DISEÑO DEL SISTEMA.....	82
3.1 Arquitectura	82
3.2 Modelo de Análisis.....	83
3.3 Modelo de Diseño	87

Tabla de Contenidos

CAPÍTULO IMPLEMENTACIÓN.....	100
4.1 Modelo de datos.....	100
4.2 Modelo de implementación.....	115
4.2.1 Diagrama de componentes.....	115
4.2.2 Diagrama de despliegue.....	118
4.3 Estándares utilizados.....	118
4.3.1 Estándares de Diseño.....	119
4.3.2 Estándares de codificación.....	125
4.4 Seguridad.....	127
CONCLUSIONES.....	129
RECOMENDACIONES.....	130
REFERENCIAS BIBLIOGRÁFICAS.....	131
BIBLIOGRAFÍA.....	133
ANEXO 1.....	136
ANEXO 2.....	141
GLOSARIO DE TÉRMINOS.....	146

INTRODUCCIÓN

El Ministerio de Salud Pública (MINSAP), órgano cubano rector del Sistema Nacional de Salud (SNS), ha sido en los últimos 49 años un proyecto social que ha garantizado la equidad, accesibilidad y seguridad médica de todos los ciudadanos. Para mantener y garantizar la excelencia de los servicios brindados en el área de salud, fue necesario realizar cambios que permitieran la transformación tecnológica, donde el empleo de la informática desempeña un papel determinante.

El programa para la informatización de la sociedad cubana persigue promover el uso masivo de las Tecnologías de la Información y las Comunicaciones (TIC) a escala nacional, teniendo en cuenta los objetivos generales estratégicos que el país se ha propuesto, además de impulsar de manera coherente todos los sectores de la sociedad.

“Una profunda revolución en los servicios de salud tendrá lugar en nuestra Patria”, [1] estas palabras expresadas por el Comandante en Jefe Fidel Castro, marcaron una nueva era en la salud pública en Cuba, ya que desde el año 2000 comienza el proceso para informatizar el SNS, Entre sus principales objetivos se encuentran incorporar la informática y los avances de las TIC a los procesos vinculados con la salud, apoyar la investigación científica, la docencia médica, la asistencia a enfermos y a tratamientos.

La Universidad de las Ciencias Informáticas (UCI) juega un papel fundamental en el desarrollo de la Industria Cubana del Software. Dentro de ella la Facultad 7 ha puesto en marcha una serie de proyectos productivos que apoyan el desarrollo de productos informáticos para la salud.

Como parte del SNS, las Unidades de Cuidados Intensivos Pediátricos (UCIP), surgieron para dar respuesta asistencial a las urgencias pediátricas, siendo el servicio hospitalario dedicado a la asistencia intensiva integral y a darle seguimiento al niño críticamente enfermo, ya que el mismo al padecer de una enfermedad grave, es propenso a que se comprometa su calidad de vida.

En el hospital Pediátrico Universitario William Soler no solo se brinda una atención especializada al niño grave, sino que se trabaja para lograr una calidad de vida óptima en los afectados, lo que redundará en una población sana en un futuro, porque como expresara el comandante en jefe: “Debemos pensar en los niños de hoy, que son el pueblo de mañana. Hay que cuidarlos y velar por ellos como los pilares con que se funda toda obra verdaderamente hermosa y útil”. [2]

Entre las principales metas del centro se encuentran: evaluar la calidad del neurodesarrollo de los niños de 0 a 5 años de edad egresados de las Unidades de Terapia Intensiva Polivalente y Neonatal (UCIP/UCIN), determinar grupos de tratamiento según la edad neurológica del niño y valorar la efectividad del Programa de Atención Temprana en dependencia de los grupos de tratamientos.

La evaluación la realizará un equipo interdisciplinario, que determinará la presencia o no de afecciones del neurodesarrollo tras su período de gravedad, el cual está compuesto por los especialistas en: fisioterapia, neurología, nutrición, neurofisiología, logopedia, psicología.

Específicamente los especialistas en Fisiología realizan las siguientes actividades:

- Valoración inicial según el diagnóstico que viene de la sala y análisis de los datos aportados por las planillas de recolección de datos primarios.
- Interrogatorio y examen físico.
- Cita para reconsulta con el objetivo de valorar resultados y decidir tratamiento del neurodesarrollo, según grupo de tratamiento.
- Evaluaciones trimestrales.

Las pruebas de evaluación del neurodesarrollo aplicadas por los fisiatras son muy extensas y todas son realizadas de forma manual. Las mismas requieren de mucho tiempo e impiden que en el mismo día puedan atender a gran cantidad de niños. Para realizar la evaluación de las pruebas el evaluador debe consultar tablas matemáticas demasiado grandes y complejas por lo que puede cometer algún error humano, además se dificulta la comparación de diagnósticos emitidos en diferentes consultas.

Por lo antes planteado se identifica como **problema a resolver** ¿Cómo viabilizar el proceso de gestión de la información que realiza el especialista en Fisiología de los niños egresados de terapia intensiva reportados como críticos durante su estadía?

Este problema se enmarca en el **objeto de estudio**: El proceso de gestión de la información relacionada con la evaluación y atención temprana de los niños egresados de terapia intensiva.

El objeto delimita el **campo de acción**: Los procesos que intervienen en la gestión de la información para la evaluación y atención temprana a los niños egresados de terapia intensiva por los especialistas en fisioterapia del Hospital Pediátrico William Soler.

Para la solución del problema se plantea como **objetivo**: Desarrollar una aplicación web que facilite el proceso de gestión de la información relacionada con los niños egresados de terapia intensiva, reportados como críticos durante su estadía, que realiza el especialista en Fisiatría.

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

1. Realizar un análisis de los sistemas informáticos de salud existentes a nivel nacional e internacional relacionados con la evaluación de los niños egresados de terapia intensiva por parte del especialista en Fisiatría.
2. Investigar la metodología y las herramientas a utilizar para el modelado del sistema informático, así como el lenguaje, la plataforma y las librerías propuestas por el grupo de arquitectura de la facultad.
3. Modelar el proceso actual de evaluación de los niños egresados de terapia intensiva al ser atendidos por el especialista en Fisiatría.
4. Analizar las necesidades de funcionamiento de la aplicación describiendo la Especificación de Requisitos de Software.
5. Realizar el prototipo no funcional del sistema.
6. Diseñar el sistema informático utilizando la metodología definida por el grupo de arquitectura de la facultad.
7. Implementar una aplicación web que siga las necesidades de funcionamiento establecidas en la Especificación de Requisitos de Software.

Con el despliegue de la aplicación se esperan los beneficios siguientes:

- Una atención rápida y profesional.
- Calidad en el diagnóstico.
- Mayor eficiencia en la realización de las pruebas del Peabody Motor Grueso y Fino.
- El país contará con una aplicación gratuita y de libre distribución.
- El software al ser realizado en la Universidad de las Ciencias Informáticas, garantiza soporte técnico de forma gratuita.

- Facilitará que el programa de evaluación y atención temprana se pueda realizar en todos los hospitales pediátricos de Cuba.

Para lograr una correcta organización del documento, se estructuró en cuatro capítulos que se describen a continuación:

Capítulo I: Fundamentación teórica. En este capítulo se realiza un estudio del estado del arte y de los antecedentes de la atención temprana. Se describen las herramientas, tecnologías y metodología escogidas para la solución del problema.

Capítulo 2: Características del Sistema. Se expone una definición del objeto de estudio del problema, se plantean los objetivos estratégicos de la organización y 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, como debe funcionar. Definición del Modelo de negocio. Especificación de los requisitos del sistema.

Capítulo 3: Descripción de la Solución Propuesta. Se realizan las definiciones del Modelo de Análisis y de Diseño. Comprende además los diferentes diagramas de diseño, así como los diagramas de secuencia y las descripciones de las diferentes clases utilizadas en el diseño.

Capítulo 4: Implementación. Se analizan los aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de componentes y despliegue. Se obtiene el modelo de datos para generar a partir del mismo la base de datos, además de la estrategia de codificación, el tratamiento de errores y como se garantiza la seguridad en la solución del sistema.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Este capítulo es el resultado de la búsqueda y del análisis de los antecedentes de la atención temprana, de los conceptos más importantes relacionados con el programa de atención y evaluación que se lleva a cabo en el hospital universitario William Soler y el estudio del estado del arte. Se describen las tecnologías y metodologías propuestas por el grupo de arquitectura de la facultad.

1.1 Antecedentes de la Atención Temprana

La discapacidad infantil constituye un problema de salud, detectarla precozmente permite el inicio de la atención temprana.

La Atención Temprana surge en el siglo XX, en países desarrollados donde la mortalidad infantil descendía cada año por el desarrollo de los cuidados intensivos neonatales por tanto el índice discapacidad en estas edades ascendían. Esto hizo necesario crear un programa encaminado a detectar prematuramente y a tratar en momento oportuno las atenciones del neurodesarrollo en los infantes, minimizando las secuelas neurológicas de los mismos. Logrando una adecuada incorporación de estos a la sociedad. A nivel mundial entre los países que reportan resultados positivos en la Atención Temprana se encuentra Ecuador, Argentina, España, Chile, Venezuela y Cuba. (3)

Esta Constituye una preocupación no solo para los estudiosos de las Ciencias Médicas sino también para las Ciencias Pedagógicas siendo así en la década de los 50, la experiencia temprana se convirtió en el principal competidor de criterios determinantes del desarrollo expuesto por Galton y partidarios donde planteaban la importancia de la herencia que aceptaba cambios en la estructura y el tamaño, pero consideraban que estos se encuentran encapsulados relativamente y no se modifican con la experiencia temprana.

No es hasta la década del 80 en que surge el modelo sistémico teniendo como principales exponentes a Shameroff y Chandler quienes plantean: "Las características biológicas y conductuales de los hijos y la calidad del entorno hogareño transactuan y rigen resultados del desarrollo." (4)

Este modelo está centrado en los padres como proveedores de estimulación.

Capítulo 1: *Fundamentación Teórica*

Para L. S. Vigotski (Científico Pedagogo soviético) los años más tempranos de la vida constituyen "el período más saturado y rico en contenido, más denso y lleno de valor del desarrollo en general." (5)

La marcada plasticidad en los primeros años de vida del cerebro es señalado por L. S. Vigotski y su escuela a la hora de sustentar la importancia de la detención y atención temprana en los niños con riesgo de alteraciones del neurodesarrollo y/o necesidades educativas especiales.

La Atención Temprana, 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. Esto facilita su adecuada maduración en todos los ámbitos y les permite alcanzar el máximo nivel de desarrollo personal y de integración social. No debe ser vista como vertiente de la rehabilitación, intervención social o educacional, sino como integradora para el desarrollo armónico de los niños con su entorno. Siendo este una de las actuaciones más rentables a largo plazo, por su eficacia, carácter preventivo y repercusión directa en la calidad de vida presente y futura de los atendidos.

Cuba a pesar de ser un país del Tercer Mundo tiene índices bajos de mortalidad comparados con los países desarrollados, salvándose niños de alto riesgo biológico lo que ha obligado a desarrollar un amplio programa de Atención Temprana que se ha extendido a lo largo del país.

1.2 Conceptos Básicos

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. El término precoz en los países de habla hispana en los setenta, se comenzó a utilizar para denominar a programas de prevención secundaria (compensatorios) para niños de alto riesgo biológico establecido o probable, orientados a contrarrestar el efecto de un daño orgánico estable. También para prevenir el apareamiento de una alteración en niños que por sus antecedentes negativos de embarazo o parto, tenían una alta probabilidad de que éste se produjera.

Programa de Estimulación Portage Project: Es un programa creado en Wisconsin, EE.UU., en 1976; que abarca 5 áreas de estimulación: Cognitiva, Sociabilización, Lenguaje, Motora y Validismo; éstas comprenden el desarrollo de un niño normal desde el nacimiento hasta los 5 años de vida. Está dividida en períodos de 1 año de edad. La estructura interna presenta un título con un número que va en aumento y que son las conductas a estimular en el niño. Después aparece un acápite de descripción de la misma y

Capítulo 1: Fundamentación Teórica

materiales sugeridos. Se dan instrucciones para desarrollar la conducta. Ha sido utilizado en muchos países en niños con riesgos biológico y ambiental. Cuba forma parte del Programa de Pesquisaje Auditivo del Hospital Pediátrico William Soler.

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

Nivel de edad basal: El primer nivel en el cual el niño obtenga una puntuación de 2 en todos los ítems o el nivel por debajo del cual el niño obtenga 0 ó 1 en uno solo de los ítems y 2 en el resto.

Nivel de edad superior: El nivel en el cual el niño puntee 0 ó 1 en todos los ítems o puntee 2 en un ítem y 0 ó 1 en el resto de los ítems.

1.3 Sistemas Automatizados existentes

ETIOBE (e-terapia inteligente para el tratamiento de la obesidad infantil)

El Consorcio Hospital General Universitario de Valencia aplica un programa informático que se fomenta entre los niños y adolescentes con problemas de obesidad. Es la primera solución informática a nivel mundial que motiva a los niños obesos a llevar un tratamiento adecuado, a la vez que favorece cambios en su estilo de vida.

Este novedoso sistema permite tomar datos de los pacientes y en función de estos ofrecer tratamientos personalizados. Para ello, utiliza las últimas tecnologías de inteligencia ambiental, entornos virtuales y agentes virtuales con los que se consigue poder personalizar los tratamientos de forma individual y aumentar la motivación y adherencia a los mismos.

ETIOBE ofrece supervisión, apoyo y ayuda a cada usuario de esta herramienta en todo momento, y en tiempo real, por medio de un sistema de seguimiento GPS (Sistema de Posicionamiento Global), permite al usuario comunicarse en cualquier momento con el centro sanitario y los facultativos que lo cuidan. Está compuesto por tres módulos que abarcan los distintos entornos de la terapia:

- El "apoyo clínico", la parte en la que el terapeuta diseña el protocolo adecuado para cada paciente, el seguimiento y el control de sus efectos.
- El "apoyo en hogar", con el que se presenta al paciente contenidos que le ayudan en el proceso del tratamiento y con el que se permite la comunicación paciente-terapeuta.

Capítulo 1: Fundamentación Teórica

- EL "Apoyo móvil", aplicación en dispositivos móviles para que el niño acceda a consejos.

ETIOBE es un programa informático muy costoso, de difícil alcance por numerosos países del mundo, entre los que se encuentra Cuba. 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 en el cual se desea implantar, ya que solo cuenta con la especialidad de nutrición.

ADE (Ambiente de Desarrollo de Evaluaciones)

Programa destinado al diseño y presentación de exámenes interactivos. Creado por ADIOware Ltda y su licencia es Freeware (es una licencia que distribuye el software sin coste y por tiempo ilimitado).

ADE es un entorno virtual encargado de facilitar todo el proceso de evaluación tanto para el docente como para el alumno. Está pensado para ser utilizado por cualquier persona porque presenta una interfaz amigable, pretende dar al usuario los máximos beneficios. Está dividido en dos módulos que facilitan la utilización del programa. Cada uno de ellos tiene funciones específicas e independientes. Los módulos son: ADE Docente y ADE Estudiante.

ADE Docente permite:

- Crear un nuevo examen.
- Editar un examen ya existente.
- Visualizar los resultados de los estudiantes en el examen creado

ADE Estudiante es el módulo destinado a la evaluación y calificación de los exámenes y permite:

- Presentar exámenes.
- Visualizar los resultados en el examen.

ADE no cuenta con técnicas o herramientas que permitan la realización de test complejos como el Peabody motor fino y grueso aplicados por los fisiatras durante el tratamiento, lo que lo hace inadaptable a otras pruebas realizadas por diferentes especialistas con los que cuenta el programa.

Software para el tratamiento terapéutico de niños con problemas cerebrales

Programa desarrollado por el Grupo de Investigación en Computación Gráfica y Procesamiento de Imágenes de OHWAHA para la atención terapéutica de niños con dificultades cerebrales reflejadas en el

Capítulo 1: Fundamentación Teórica

aprendizaje en el área de las matemáticas. Este software desarrolla habilidades como aprender a clasificar, Ordenar por cantidad, aprender a distinguir formas, tamaños y colores, asociación de figuras, identificación de números, suma y resta entre otros.

Tiene como inconvenientes que no cubre ninguna de las especialidades que necesita el programa de atención y evaluación que se lleva a cabo en el Hospital Pediátrico William Soler. Tampoco cumple con la arquitectura definida para el proyecto SENDN.

Proyecto Esperanza

El Proyecto Esperanza que se desarrolla en la provincia de Camagüey tiene como objetivos:

- Unir voluntades y divulgar información científica en la tarea de incrementar la calidad de la atención al niño discapacitado.
- Caracterizar las diferentes formas de discapacidad que afectan a la población infantil en el municipio Camagüey, creando una base de datos al respecto.
- Desarrollar una labor educativa en la comunidad sobre el tema de la prevención de la discapacidad.
- Divulgar los adelantos en el campo del diagnóstico precoz de la discapacidad.
- Desarrollar un programa de intervención temprana, adaptado a las características propias del sistema de salud.
- Implementar la intervención familiar comunitaria, mediante la creación de una escuela para padres de niños discapacitados y una línea telefónica de ayuda.
- Facilitar la integración social, educando a la comunidad.

El Proyecto Esperanza divulga información desarrollando investigaciones en el marco de Proyectos Ramales orientadas a incrementar la calidad de la atención a los niños con discapacidad, entre ellos se encuentra el seguimiento de todos los egresados de la terapia intensiva neonatológica, la creación de bases de datos en formato digital, y el desarrollo de la expresión artística en el discapacitado son las metas fundamentales a seguir en este proyecto.

Capítulo 1: *Fundamentación Teórica*

A pesar de cumplir con las características identificadas para el programa llevado a cabo en el Hospital William Soler, no cuenta con un sistema informático capaz de gestionar la información.

1.4 Metodología de Desarrollo

Una Metodología de Desarrollo de Software es un conjunto de procedimientos, técnicas, herramientas y un soporte de documentos que ayuda a los desarrolladores a realizar un software. Una metodología puede seguir uno o varios modelos de ciclo de vida, esta indica cómo hay que obtener los distintos productos parciales y finales en el desarrollo de un software.

El resultado final de un proyecto de software, es un producto donde intervienen las personas a través de un proceso de desarrollo que guía los esfuerzos de estas, a modo de plantilla, explicando los pasos necesarios para terminar el proyecto.

Proceso Unificado de Desarrollo RUP

“El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software.”

(6) El Proceso Unificado de Desarrollo (RUP) es un proceso bien definido, estructurado y adaptable a las características y necesidades de cada proyecto específico.

RUP fue creado por Ivar Jacobson, Grady Booch y James Rumbaugh en 1998, su objetivo es producir software de alta calidad, que cumpla con los requerimientos de los usuarios dentro de la planificación y presupuesto establecidos. (7) RUP es un proceso de desarrollo de software y junto al Lenguaje Unificado de Modelado (UML) constituye la metodología estándar más utilizada para el desarrollo de un producto de software. El ciclo de vida de RUP se caracteriza por:

Dirigido por casos de uso: El proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso son un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

Capítulo 1: *Fundamentación Teórica*

Iterativo e incremental: En donde el trabajo se divide en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

Define cuatro elementos: los roles, que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los artefactos, que responden a la pregunta ¿Qué? y los flujos de trabajo que responde a la pregunta ¿Cuándo?

Esta metodología está formada por nueve flujos de trabajos y cuatro fases, basándose en UML como lenguaje de modelado.

1.5 Tecnologías

En este epígrafe se tratarán una serie de conceptos relacionados con las tecnologías a utilizar en el proceso de desarrollo de las funcionalidades para el Sistema de Evaluación del neurodesarrollo en Niños. Estas tecnologías 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 se define como la estructura de los componentes de un programa o sistema, sus interrelaciones, y los principios y reglas que gobiernan su diseño y evolución en el tiempo. Aporta además una visión abstracta de alto nivel, postergando el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. Establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema. Para el desarrollo de las funcionalidades se propone la utilización del patrón de diseño Modelo-Vista-Controlador y el Patrón en capas.

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.

Capítulo 1: *Fundamentación Teórica*

El MVC en una aplicación web, se representa a través de la Vista que constituyen las páginas HTML y el código que provee de datos dinámicos a la página, que permite mostrarle la información al usuario.

El Controlador que gestiona todos los eventos de entrada del usuario y es el intermediario entre el Modelo y la Vista. El Modelo contiene toda la información persistente y reglas de negocio.

El uso de este patrón es favorable; establece una separación entre los componentes de un programa permitiendo implementarlos por separado; esto posibilita que si uno de estos componentes tiene un mal funcionamiento puede reemplazarse sin que se vean afectados los demás. Permite que no haya ninguna dependencia directa entre la Vista y la Modelo, esto beneficia que haya un mayor soporte a los cambios en las interfaces, que tienden a cambiar con más frecuencia que la lógica del negocio. Se puede adicionar nuevas vistas sin afectar la Modelo, es decir, múltiples páginas en la aplicación web pueden emplear los mismos elementos del Modelo.

Patrón en capas

El patrón en capas, es un estilo de programación cuyo objetivo principal es la separación y agrupamiento de los componentes del software, atendiendo a su función en el mismo, con relación al usuario del sistema, la información que este maneja y las operaciones que el usuario realiza sobre la misma. Esta división muchas veces se hace en tres capas: la capa de presentación, capa de negocio y la capa de datos.

Los tres niveles o capas son: (8)

Capa de presentación: Presenta el sistema al usuario, comunica y captura la información del usuario dando un mínimo de. 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

Capítulo 1: *Fundamentación Teórica*

interactuar con la base de datos) generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.

Capa de presentación

Java Server Faces (JSF)

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. Permite desarrollar rápidamente aplicaciones de negocio dinámicas en las que toda la lógica de negocio se implementa en Java, o es llamada desde Java, creando páginas para las vistas muy sencillas. Además resuelve validaciones, conversiones, mensajes de error e internacionalización.

Richfaces

Richfaces es una biblioteca de componentes para JSF y un avanzado framework para la integración de AJAX con facilidad en la capacidad de desarrollo de aplicaciones de negocio. Los componentes de Richfaces 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. Richfaces también incluye un fuerte apoyo para la skinnability de aplicaciones JSF. Richfaces también aprovecha al máximo los beneficios del framework JSF incluyendo, la validación y conversión de instalaciones, junto con la gestión de estática y dinámica los recursos.

Ajax4JSF

Ajax4JSF es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript.

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

Capítulo 1: Fundamentación Teórica

Facelets

JavaServer Facelets es un framework para plantillas (templates) centrado en la tecnología JSF (JavaServer Faces), por lo cual se integran de manera muy fácil.

Características de JavaServer Facelets:

- Facilidad en la creación de plantillas para los componentes y páginas.
- Un buen sistema de reporte de errores.
- No es necesaria configuración XML.

Las principales ventajas de Facelets son:

- Construcción de interfaces basadas en plantillas.
- Rápida creación de componentes por composición.
- Fácil creación de funciones y librerías de componentes.

Lenguaje Extensible de Marcado de Hipertexto (XHTML)

XHTML es un lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. Es la versión XML de HTML con las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas.

Capa de negocio

JBoss Seam

Seam es un framework OpenSource desarrollado por la empresa JBoss, con el fin de unir diferentes tecnologías y estándares de Java en un solo framework, a la vez que añade algunas funcionalidades no contempladas por ellos.

Seam permite usar EJB3 (Enterprise JavaBeans) + JSF (Java Server Faces) de una forma muy sencilla, además de añadirte herramientas tremendamente útiles para el desarrollo de aplicaciones Web, todo en un solo framework bien acoplado y basado en estándares ampliamente utilizados y probados (escalables, portables y reusables).

Capítulo 1: Fundamentación Teórica

Drools

Son reglas de negocio manejadas por un motor de reglas de aplicación en un sistema. El motor de reglas se encuentra implementado por ReteOO, que es una base del algoritmo Rete de Charles Forgy adaptados para el lenguaje Java.

Es lo suficientemente flexible como para que coincida con la semántica del dominio de su problema en determinados idiomas, provee herramientas de edición gráfica, herramientas basadas en Web y desarrolladores de herramientas de productividad.

Características

- Código abierto: Tiene licencia de fuente abierta que le hace libre para descargar, usar, integrar y distribuir.
- Programación Declarativa: Permite decir “que hacer” y no decir “como hacerlo”.
- Accesibles reglas de negocio: Es amigable tanto a desarrolladores como administradores de las reglas de negocio. Permite a los desarrolladores a escribir casi todas las reglas de negocio como un lenguaje natural. Lógica y los datos por separado
- Basado en Web: Basado en AJAX, construye desde cero hasta avanzadas reglas de aplicación, control de versiones, y de administración. (9)

JBPM

Es una plataforma de Workflow y automatización de procesos que habilita la coordinación entre aplicaciones y servicios desintegrados. JBPM combina la facilidad de desarrollo de aplicaciones Workflow, con un motor flexible y escalable para la ejecución de transacciones de negocio. (10)

Capa de acceso a datos

Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Capítulo 1: *Fundamentación Teórica*

Es una herramienta ORM completa que ha conseguido en un tiempo record una excelente reputación en la comunidad de desarrollo posicionándose claramente como el producto Opensource líder en este campo gracias a sus prestaciones, buena documentación y estabilidad. Es valorado por muchos incluso como solución superior a productos comerciales dentro de su enfoque, siendo una muestra clara de su reputación y soporte la reciente integración dentro del grupo JBOSS que seguramente generará iniciativas muy interesantes para el uso de Hibernate dentro de este servidor de aplicaciones.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la programación orientada a objetos. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL.

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.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criterio").

Enterprise JavaBeans (EJB3)

Los Enterprise JavaBeans (EJB3) son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems (ahora JEE 5.0). Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB.

Los EJB proporcionan un modelo distribuido y estándar de componentes que se ejecutan en el servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad) para

Capítulo 1: *Fundamentación Teórica*

centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

Java Persistence API (JPA)

JPA es la API para la persistencia de objetos Java a cualquier base de datos relacional. Esta API fue desarrollada para la plataforma Java EE e incluida en el estándar de EJB 3.0. Busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí ocurría con EJB2, y permitir usar objetos regulares.

Tecnologías horizontales

Existen un conjunto de tecnologías horizontalmente que se extienden 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)

JavaEE5 o Java versión 5, es una plataforma de Software usada ampliamente para programación del lado del servidor (programación de aplicaciones empresariales) en el lenguaje de programación Java. Permitir la tolerancia a fallos en aplicaciones Java de NCapas (multi-tier), basadas en componentes modulares que se ejecutan en un servidor de aplicaciones. Java EE es un estándar para desarrollar aplicaciones Java portátiles, robustas, escalables y seguras para el servidor. La plataforma Java EE 5 proporciona a los desarrolladores un potente conjunto de API que reducen el tiempo de desarrollo y la complejidad, al tiempo que mejoran el rendimiento de las aplicaciones.

Java Runtime Environment (JRE)

JRE es el acrónimo de Java Runtime Environment (entorno en tiempo de ejecución Java) y se corresponde con un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas.

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

Capítulo 1: *Fundamentación Teórica*

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 (Java Development Kit), que además del JRE (mínimo imprescindible) incluye, entre otros, un compilador para Java.

1.6 Lenguajes

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. 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. (11)

Java

Java es un lenguaje de programación orientado a objetos, toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, los programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware. La recolección de basura de Java es un proceso prácticamente invisible al desarrollador. Es decir, el programador no tiene conciencia de cuándo la recolección de basura tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente.

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

BPMN es un nuevo estándar para modelar flujos de procesos de negocio y servicios web. Su meta principal es definir una notación entendible para todos los usuarios del negocio, desde los analistas del negocio que modelan los procesos hasta los desarrolladores técnicos responsables de implementarlos, garantizando la mejora de los mismos y finalmente a los usuarios del negocio quienes monitorearán y administrarán dichos procesos. BPMN crea un puente estándar entre el diseño del proceso de negocio y su implementación.

BPMN es un facilitador de fondo para una nueva iniciativa en el mundo de la arquitectura empresarial, la Administración de Procesos de Negocio (BPM). Esta permite gestionar el cambio para mejorar los procesos de negocio y unifica las disciplinas de Modelación de Procesos, Simulación, Flujo de trabajo,

Capítulo 1: Fundamentación Teórica

Integración de Aplicaciones Empresariales (EAI) y la integración Business-to-Business (B2B) en un único estándar. (12)

Lenguaje Unificado de Modelado (UML)

“El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual para especificar, visualizar, construir y documentar artefactos de un sistema de software.” (13).UML proporciona una forma estándar de representar los planos de un sistema, y comprende tanto elementos conceptuales, como los procesos de negocio y las funciones del sistema, cuando elementos concretos, como las clases escritas de un lenguaje de programación específicos, esquemas de bases de datos y componentes software reutilizables.

RUP emplea UML como base para el desarrollo de software en cada una de sus fases y disciplinas. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo.

Dentro de las ventajas que proporciona este lenguaje de modelado se puede destacar las siguientes:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyendo así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo.
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

1.7 Servidor de aplicaciones.

Un servidor de aplicaciones es un software que proporciona aplicaciones a los equipos o dispositivos cliente, por lo general a través de Internet y utilizando el protocolo http (HyperText Transfer Protocol o

Capítulo 1: *Fundamentación Teórica*

protocolo de transferencia de hipertexto). Es un producto basado en un componente que se encuentra en el plano medio de la arquitectura central de un servidor. Proporciona servicios de 'middleware', es decir, trabaja como un intermediario para la seguridad y el mantenimiento, además de proveer acceso a los datos.

JBoss Server o JBoss AS

JBoss es el servidor de aplicaciones de código abierto más utilizado actualmente en todo el mundo. Este servidor de aplicaciones se encuentra certificado J2EE y soporta sistemas de gran complejidad y alta concurrencia. Al estar basado en Java, puede ser utilizado en cualquier sistema operativo que lo soporte. Implementa todo el paquete de servicios de J2EE. Las características destacadas de JBoss incluyen: Producto de licencia de código abierto sin coste adicional. Cumple los estándares. Confiable a nivel de empresa. Orientado a arquitectura de servicios. Flexibilidad consistente. Servicios del middleware para cualquier objeto de Java. Ayuda profesional 24x7 de la fuente. Soporte completo para JMX.

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 revolucionaria con una licencia de código abierto, puede ser descargado, utilizado, instalado, y distribuido sin restricciones por la licencia. Por este motivo es la plataforma más popular de middleware para desarrolladores, vendedores de software independientes, así como para grandes empresas.

1.8 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." (14)

Los SGBD están constituidos por un paquete de software cuya función es la gestión del acceso a la BD, las operaciones fundamentales son: crear, modificar, eliminar y obtener la estructura asociada al esquema lógico de una BD. El objetivo principal es proporcionar un entorno práctico y eficiente a la hora de almacenar y recuperar la información de la base de datos, lo que facilita la gestión de grandes volúmenes de datos.

Capítulo 1: *Fundamentación Teórica*

PostgreSQL

PostgreSQL es un potente Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) de código abierto. Tiene más de 15 años de activo proceso de desarrollo a nivel mundial y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de los datos, y la corrección. Es multiplataforma y funciona en los principales sistemas operativos, como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. También apoya el almacenamiento de grandes objetos binarios, imágenes, sonidos o vídeo.

Características de PostgreSQL:

- Objeto-relacionales: PostgreSQL en cada tabla define una clase que permite implementar la herencia entre tablas o clases, funciones y operadores polimórficos.
- Implementación del estándar SQL92/SQL99.
- Código abierto: Debido a la licencia liberal, PostgreSQL puede ser usado, modificado y distribuido por todo el mundo de forma gratuita para cualquier fin, ya sea de datos, comerciales o académicas.
- Múltiples lenguajes de procedimientos: Los disparadores y otros procedimientos pueden ser escritos en varios lenguajes de procedimientos. Código del lado del servidor es comúnmente escrito en PL / PostgreSQL, un lenguaje de procedimiento similar al de Oracle PL / SQL. También se puede desarrollar código del lado del servidor en Tcl, Perl, incluso bash (el de código abierto Linux / Unix shell).
- Múltiples-cliente API: PostgreSQL soporta el desarrollo de aplicaciones cliente en varios lenguajes, interfaz para PostgreSQL desde C, C ++, ODBC, Perl, PHP, Tcl / Tk, y Python.
- Tipos de datos: integer, string, numeric, boolean, char, varchar, date, interval, y timestamp, tipos geométrica, tipo de datos booleanos y tipos de datos diseñados específicamente para hacer frente a las direcciones de red.
- La extensibilidad: Es una de las características más importantes de PostgreSQL ya que puede ser ampliado, se pueden añadir nuevos tipos de datos, nuevas funciones y operadores, e incluso nuevos lenguajes de procedimiento y de cliente.

Capítulo 1: *Fundamentación Teórica*

- Tiene un soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados.
- MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios, es decir, mientras un proceso escribe no se bloquean las tablas, ni siquiera las tuplas. Mediante el uso de MVCC, PostgreSQL evita el problema de que procesos lectores esperen a que se termine de escribir por completo. MVCC está considerado mejor que el bloqueo a nivel de tupla porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.
- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

1.9 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

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Visual Paradigm

La herramienta CASE Visual Paradigm for UML Enterprise Edition utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de modelado. Está diseñada para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocios, arquitectos y desarrolladores. Está orientada a la creación de diseños y se usa el paradigma de programación orientada a objetos. Visual Paradigm (VP) incluye una herramienta llamada Visual Architect que permite la generación de código para el manejo de la base de datos.

Capítulo 1: *Fundamentación Teórica*

Con esta herramienta se puede generar código para los lenguajes PHP, JAVA y C# y para los gestores de base de datos DB2, Informix, SQL Server, MySQL, Oracle y PostgreSQL.

Beneficios de Visual Paradigm para UML:

- Persistencia de forma fácil.

Los desarrolladores emplean mucho esfuerzo en salvar y cargar objetos entre la memoria y la base de datos lo que hace que el programa sea complicado y difícil de mantener. VP simplifica estas tareas mediante la generación de una capa de persistencia entre objeto y modelos de datos.

- Generador de mapeo objeto-relacional sofisticado.

La capa de mapeo objeto-relacional que se genera incorpora características como soporte de transacciones, capa de conectar en caché, agrupación de conexiones y personalización de sentencias SQL.

- Amplia cobertura para bases de datos.

Soporta una amplia gama de base de datos, incluidos Oracle, DB2, Cloudscape / Derby, Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Microsoft SQL Server, PostgreSQL, MySQL y otros.

- Base de datos de ingeniería inversa.

Permite la ingeniería inversa existente en una base de datos a través JDBC en el modelo entidad-relación. Los desarrolladores pueden transformar el modelo entidad-relación al modelo de objetos y rediseñar la base de datos para un mayor desarrollo.

- Integración con IDE.

VP-UML no sólo es una aplicación independiente, se puede integrar a los principales Integrated Development Environments (IDEs): Eclipse/WebSphere®, Borland.

Ambiente de Desarrollo Integrado (IDE)

Eclipse

Eclipse es una plataforma de software de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano"

Capítulo 1: Fundamentación Teórica

basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar un entorno integrado de desarrollo. Este IDE es más general para el desarrollo de aplicaciones en Java, en la que se ha especializado, también permite programar en PHP, aunque con muy pocas opciones de revisión y corrección de errores y sin completamiento de código. Es una herramienta que necesita de mucha ayuda del hardware para realizar la compilación del código fuente escrito, además de ser multiplataforma.

En este capítulo, se hizo referencia a los principales conceptos relacionados con el dominio del problema. Además, se realizó un estudio sobre los sistemas existentes a nivel nacional e internacional. Se realizó un profundo análisis de las metodologías, herramientas y tecnologías a utilizar propuestas por el grupo de arquitectura de la facultad y la dirección del proyecto.

CAPÍTULO 2

CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se realiza un análisis de la propuesta del sistema. Para ello se describen los procesos, las reglas del negocio, se especifican los requisitos funcionales y no funcionales, se identifica mediante el diagrama de casos de uso del sistema las relaciones entre los actores y casos de uso del sistema, así como las descripciones textuales de los casos de uso.

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

En el Hospital Pediátrico William Soler es llevado a cabo un proceso de evaluación de la calidad del neurodesarrollo en los niños de 0-5 años de edad egresados de las Unidades de Cuidados Intensivos Polivalente y Neonatal (UCIP/UCIN) persiguiendo los siguientes objetivos:

- 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 a los hallazgos encontrados.

Para lograr dichos objetivos se cuenta con un grupo interdisciplinario quienes determinarán la presencia o no de afecciones del neurodesarrollo tras su período de gravedad, el cual está compuesto por los especialistas en: fisiatría, neurología, nutrición, neurofisiología, logopedia, psicología.

1.5.2 Descripción de los Procesos de Negocio

Los especialistas en fisiatría para evaluar a los niños necesitan aplicarle una serie de pruebas entre las que se encuentran: Peabody Motor Fino y Grueso.

El PDMS (Peabody Motor Fino y Grueso) es un test estandarizado de administración individual, que mide las habilidades motoras finas y gruesas de los niños desde su nacimiento hasta la edad de 83 meses. Puede en algunas condiciones aplicarse de manera grupal. (15)

Capítulo 2: Características del Sistema

Este tipo de escala es necesaria para determinar el lugar de desarrollo de un niño comparado con los niños de su edad. Cuando se detecta una anomalía en el desarrollo de un niño, se debe programar una intervención terapéutica para influir en dicho desarrollo a través de la adquisición de las habilidades necesarias para esa edad.

La Escala Motora Gruesa consiste en tareas que requieren movimientos precisos de los músculos largos del cuerpo, estos ítems se clasifican en 5 categorías de habilidades. Cuenta con 170 ítems.

- (A) Reflejos
- (B) Equilibrio
- (C) No locomotora
- (D) Locomotora
- (E) Recepción y propulsión de objetos

La Escala Motora Fina consiste en tareas que requiere movimientos precisos de los músculos pequeños del cuerpo. Sus ítems se clasifican en 4 categorías de habilidades. Cuenta con 112 ítems.

- (A) Agarre
- (B) Uso de las manos
- (C) Coordinación óculo manual
- (D) Destreza manual

Ambas escalas pueden ser administradas al niño entre 45 y 60 minutos, cada una requiere de 20 a 30 minutos para su aplicación. Ambas escalas se administran y califican por separado. El examinador puede administrar una sola, en dependencia de sus necesidades. No es necesario aplicar ambas escalas en la misma sesión, aunque se recomienda un margen no mayor de 5 días entre una y otra parte.

Antes de aplicar el test, se debe obtener toda la información preliminar acerca del niño, y escribirla en la hoja de respuestas. Debe calcularse la edad cronológica del niño en meses, pues sus calificaciones pueden ser comparadas con la de niños de su misma edad de la norma popular.

Capítulo 2: Características del Sistema

Las normas PDMS se basan en puntuar cada ítem 0,1 ó 2. El examinador debe tomar la decisión del puntaje, basado en su opinión acerca del desempeño del niño en ítem y el criterio general para los puntajes de 0, 1 ó 2.

El primer nivel administrado debe ser lo suficientemente fácil como para restaurar confianza y éxito en el niño. Se debe comenzar administrando cada escala un nivel por debajo de la edad motora esperada (En el caso de los niños sin dificultades). Sí el niño puntea 0 ó 1 en los ítems de ese nivel, se descontinúa ese nivel y se pasa a uno previo o anterior al mismo. Se continúa probando con los niveles anteriores hasta que se encuentre uno en el cual el niño puntée 2 en todos los ítems o 0 ó 1 en 1 ítem y 2 en el resto, ese nivel es el nivel de edad basal.

Después de determinar la edad basal, se procede con el primer ítem no administrado durante la determinación de la edad superior y completar este nivel. Continuar administrando niveles hasta que aparezca un nivel en el cual el niño puntée 0 ó 1 en todos los ítems o 2 en un ítem y 0 ó 1 en el resto. Ese nivel es el nivel de edad superior, y en ese momento se detiene la aplicación del test.

La evaluadora que aplique el test debe elaborar un sumario con el resultado de las pruebas el cual será analizado por la fisiatra quién examinará físicamente al paciente y emitirá resumen final.

1.5.3 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 pruebas aplicadas por los fisiatras ya que estas 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.

1.5.4 Objeto de Automatización

Con el fin 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 fisioterapia, se propone realizar una aplicación web que tiene como objeto de automatización inicial las pruebas que se realizan en esta especialidad, dando la posibilidad de la generación automática de los sumarios, posibilitando así una

Capítulo 2: Características del Sistema

mayor confidencialidad, seguridad y control de la información que se manipula, así como la disminución del tiempo de aplicación de dichas pruebas.

2.4 Modelo del negocio

El Modelo del Negocio es una técnica para describir los procesos de la organización bajo estudio que permite la especificación de los requisitos más importantes del sistema determinados a través del propio negocio. La finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades, roles y las reglas del mismo. Proporciona un entendimiento común entre clientes y desarrolladores en la comprensión de la estructura y la dinámica de la organización donde se va a implantar el sistema. Se enfoca en comprender los problemas actuales de la organización e identifica mejoras potenciales.

1.5.1 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 (juega un rol).

Actor del Negocio: Cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. (16)

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. (17)

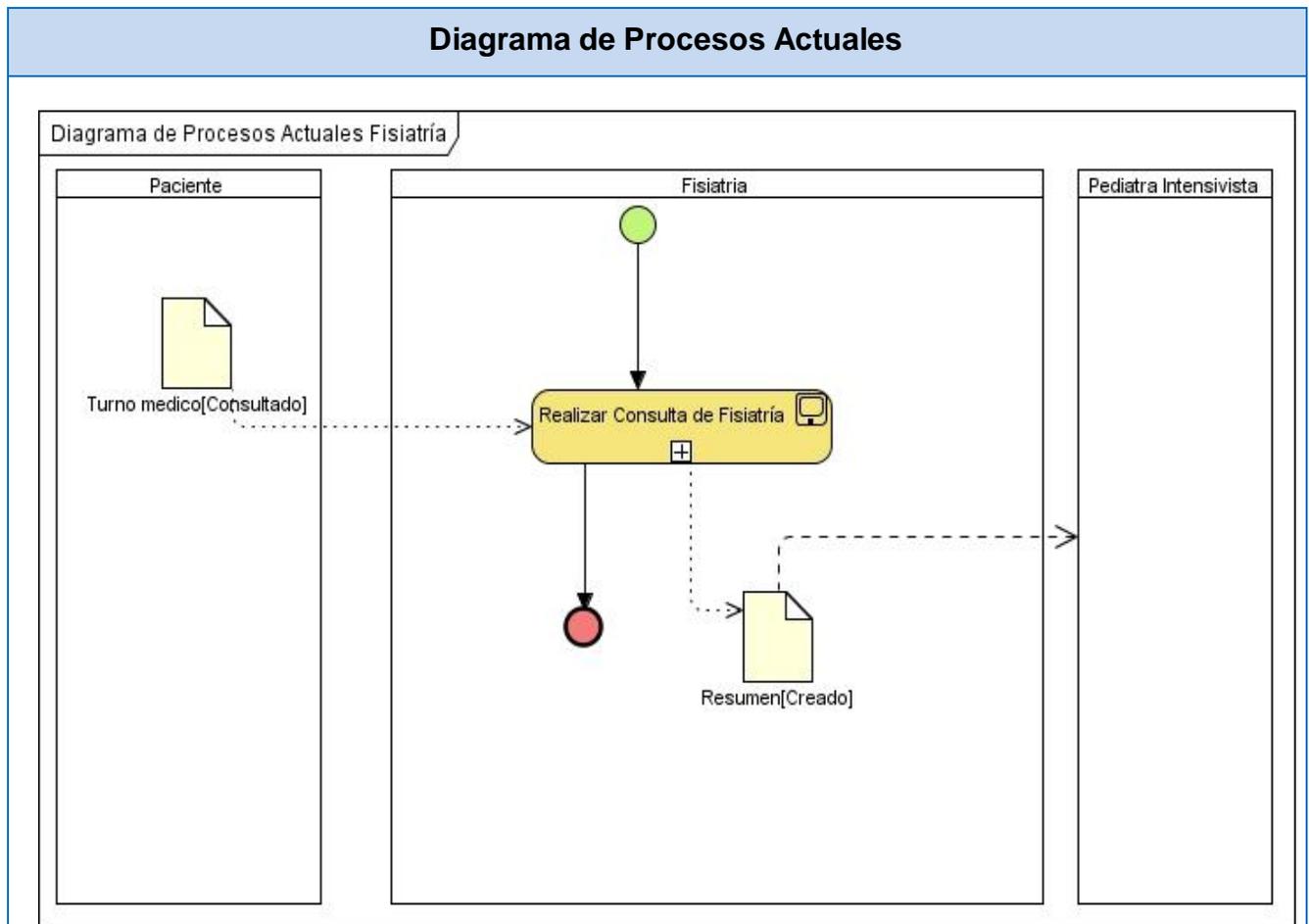
Capítulo 2: Características del Sistema

Trabajador del negocio	Descripción
Evaluador	Es el técnico encargado de aplicar las pruebas y emitir sumario.
Fisiatra	Es el especialista que analiza el sumario de la prueba, examina físicamente al paciente y emite resumen final de la consulta.

1.5.2 Diagrama de proceso del negocio

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. (18)

Subproceso: Son partes bien definidas en un proceso. Su identificación puede resultar útil para aislar los problemas que pueden presentarse y posibilitar diferentes tratamientos dentro de un mismo proceso. (19)



Descripción de los Procesos del Negocio

PROCESO:	Realizar Consulta de Fisiatría.
Misión:	Aplicar las pruebas necesarias para obtener la evaluación neurológica de los pacientes en la especialidad de Fisiatría.
Responsable:	Dra. Alexis Montero Terry, Fisiatra del hospital William Soler.

Capítulo 2: Características del Sistema

ACTORES INVOLUCRADOS				
Rol		Funciones		
Evaluador(a)		Es el/la principal ejecutor (a) de las actividades relacionadas con la aplicación de las pruebas que se realizan dentro del proceso realizar consulta de Fisiatría.		
Fisiatra		Es el/la principal ejecutor(a) de las actividades relacionadas con la realización del examen físico y el resumen de la evaluación.		
ACTIVIDADES				
Actividad:		Verificar Turno		
Flujo de Información				
El/La paciente llega a la consulta y entrega su turno médico. El/La evaluador(a) verifica que le corresponda el turno ese día y en caso de no corresponderle este se retira de la consulta.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Turno médico	Pediatra intensivista	Evaluador(a)	Papel	Cada vez que se planifique una consulta.
Actividad		Aplicar test.		
Flujo de Información				
Para realizar la evaluación del paciente se aplican las pruebas del Peabody Motor Grueso y Fino.				

Capítulo 2: Características del Sistema

Para la aplicación de ambas pruebas el/ la evaluador(a) busca los test correspondientes al paciente; según la edad cronológica del mismo y los aplica consultando el instructivo y la Escala Motora Gruesa o Fina según la prueba que se este realizando, luego escribe los resultados en la hoja de puntaje.

Artefacto	Emisor	Receptor	Formato	Frecuencia
Hoja de puntaje	Evaluador(a)	Evaluador(a)	Papel	Cada vez que se realice una prueba.
Actividad:	Evaluar test.			

Flujo de Información

Para evaluar los resultados de los test se calcula el puntaje bruto, este valor es buscado en la tabla Percentil determinando así el percentil por cada una de las habilidades y luego mediante la tabla Coeficiente de Desarrollo Motor se calcula el DQM para estos percentiles. Después se toma el puntaje bruto total y mediante las tablas Puntaje de Escala y Edad Equivalente se calcula la edad neurológica del paciente y se emite el sumario con los resultados de las pruebas.

Artefacto	Emisor	Receptor	Formato	Frecuencia
Sumario	Evaluador(a)	Evaluador(a)	papel	Cada vez que se concluyan las pruebas de un paciente.
Actividad:	Analizar sumario			

Flujo de Información

El/La fisiatra analiza el sumario correspondiente al paciente.

Capítulo 2: Características del Sistema

Artefacto	Emisor	Receptor	Formato	Frecuencia
Sumario	Evaluador (a)	fisiatra	Papel	Cada vez que se concluyan las pruebas de un paciente.
Actividad:	Evaluar paciente			
Flujo de Información				
El/La fisiatra evalúa físicamente al paciente, analizando una serie de parámetros de vital importancia para determinar un diagnóstico.				
Actividad:	Emitir resumen			
Flujo de Información				
El/La fisiatra crea el resumen final de la consulta donde escribe sus observaciones, teniendo en cuenta los resultados de las pruebas y el examen físico del paciente.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Resumen	Fisiatra	Fisiatra	Papel	Cada vez que se concluyan las pruebas de un paciente.
REGLAS DEL NEGOCIO				
Regla		Actividad		
<ul style="list-style-type: none"> • Para determinar las edades del niño 		Aplicar Test		

se debe tener en cuenta:

- Edad Basal: El primer nivel en el cual el niño obtenga una puntuación de 2 en todos los ítems o el nivel por debajo del cual el niño obtenga 0 ó 1 en uno solo de los ítems y 2 en el resto.
- Nivel de edad superior: El nivel en el cual el niño puntée 0 ó 1 en todos los ítems o puntée 2 en un ítem y 0 ó 1 en el resto de los ítems.

• **Calificación por puntos**

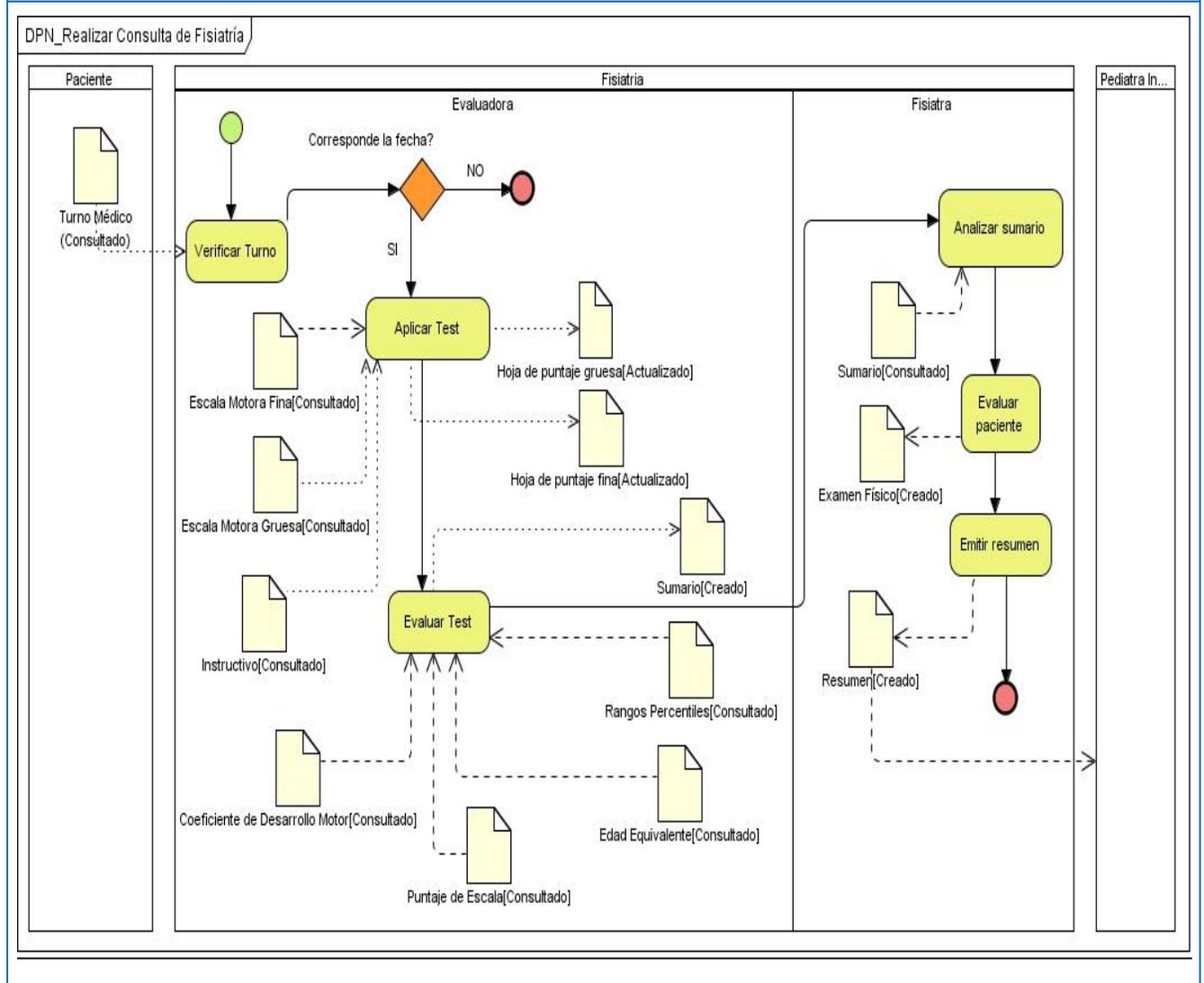
- 0-El niño no puede o no intenta realizar el ítem, o el intento no muestra que la habilidad está emergiendo.
- 1- El desempeño del ítem por el niño muestra una clara semejanza con el criterio máximo del ítem, pero no cumple por completo la realización del mismo.
- 2 - El niño resuelve el ítem tal y como aparece en el criterio para la realización del mismo.

REFERENCIAS A PROCEDIMIENTOS

Capítulo 2: Características del Sistema

Procedimiento	Descripción
Instructivo	Es el documento que describe las pruebas.

DIAGRAMA DE PROCESO



2.5 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.

1.5.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, indican que es lo que el software debe hacer, especifican como debe comportarse el sistema en situaciones particulares y como debe ser el comportamiento de entrada y salida del sistema.

RF 1-Realizar Test Fisiatría

RF 2-Actualizar Sumario Fisiatría

RF 3-Resumen

3.1-Crear Resumen Fisiatría

3.2- Visualizar Resumen Fisiatría

3.3 - Modificar Resumen Fisiatría

3.4 - Buscar Resumen Fisiatría

RF 4-Listar pacientes atendidos

RF 5-Reportes

5.1- Generar Reporte de Signos de alarma neuromotores

5.2- Generar Reporte de Asimetría postural de los miembros

5.3- Generar Reporte de Tono Pasivo.

5.4- Generar Reporte de Postura y Actividad motora espontánea

5.5- Generar Reporte de Reflejos Primarios

RF 6- Gestionar Configuración Test Fisiatría (Nomenclador)

Capítulo 2: Características del Sistema

6.1- Crear Configuración Test Fisiatría

6.2- Modificar Configuración Test Fisiatría

6.3- Listar Configuración Test Fisiatría

6.5- Eliminar Configuración Test Fisiatría

RF 7- Gestionar Edad Equivalente (Nomenclador)

7.1- Insertar Edad Equivalente

7.2- Modificar Edad Equivalente

7.3 Listar Edad Equivalente

7.4 Eliminar Edad Equivalente

RF 8 - Gestionar Rangos Percentiles (Nomenclador)

8.1- Insertar rangos percentiles

8.2- Modificar rangos percentiles

8.3 Listar rangos percentiles

8.4 Eliminar rangos percentiles

RF 9 - Gestionar Percentil Total (Nomenclador)

9.1 Crear Percentil Total

9.2 Modificar Percentil Total

9.3 Listar Percentil Total

9.4 Eliminar Percentil Total

RF 10- Gestionar Puntaje de Escala Grueso y Fino (Nomenclador)

10.1- Insertar rangos percentiles

10.2- Modificar rangos percentiles

Capítulo 2: Características del Sistema

10.3 Listar rangos percentiles

10.4 Eliminar rangos percentiles

RF 11- Gestionar Edad (Nomenclador)

11.1 Crear Edad

11.2 Modificar Edad

11.3 Listar Edad

11.4 Eliminar Edad

RF 12- Gestionar Asociaciones de Operaciones (Nomenclador)

12.1 Crear Asociaciones de Operaciones

12.2 Modificar Asociaciones de Operaciones

12.3 Listar Asociaciones de Operaciones

12.4 Eliminar Asociaciones de Operaciones

RF 13-Gestionar DMQ (Nomenclador)

13.1- Listar DMQ.

13.2- Insertar DMQ.

13.3- Modificar DMQ.

13.4 Eliminar DMQ.

RF 14- Gestionar Habilidad (Nomenclador)

14.1 Crear Habilidad

14.2 Modificar Habilidad

14.3 Listar Habilidad

14.4 Eliminar Habilidad

Capítulo 2: Características del Sistema

RF 15- Gestionar Actividad Espontánea (Nomenclador)

15.1 Crear Actividad Espontánea

15.2 Modificar Actividad Espontánea

15.3 Listar Actividad Espontánea

15.4 Eliminar Actividad Espontánea

RF 16- Gestionar Ítems (Nomenclador)

16.1 Crear Ítems

16.2 Modificar Ítems

16.3 Listar Ítems

16.4 Eliminar Ítems

5.2.1 Requisitos no funcionales

Los requerimientos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.

Usabilidad

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

Usuarios normales: 20 días

Usuarios avanzados: 30 días

Fiabilidad

En los servidores de los hospitales y en el Centro de Datos Nacional del MPPS (Ministerio del Poder Popular para la Salud) se garantizará una arquitectura de máxima disponibilidad, tanto de servidores de aplicación como de base de datos. Se garantizarán además, políticas de respaldo a toda la información, evitando pérdidas en caso de desastres ajenos al sistema.

Capítulo 2: Características del Sistema

Los estudios imagenológicos y otros datos que por su tamaño no se puedan replicar hacia el Centro de Datos, se almacenarán localmente en los hospitales; quedando la referencia a dicho estudio en el Centro de Datos, de tal forma que se pueda acceder a dichos estudios mediante una transmisión directa entre los hospitales, sin que medie para esto el Centro de Datos Nacional.

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

Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo 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 BD, independientemente de que para el sistema, este elemento ya no exista.

Capítulo 2: Características del Sistema

El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvallas realizadas.

Eficiencia

El Centro de Datos permitirá agregar recursos para aumentar el poder de procesamiento y almacenamiento sin afectar los sistemas, garantizando expansiones motivadas por futuros requerimientos.

El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.

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.

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

Capítulo 2: Características del Sistema

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.

Capítulo 2: Características del Sistema

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.

Requisitos 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.

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 desplegables que faciliten y aceleren su utilización.

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

Todos los textos y mensajes en pantalla aparecerán en idioma español.

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.

Capítulo 2: Características del Sistema

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.

Permite el desarrollo y mantenimiento de una interface única, común a todas las implementaciones, por tanto no es necesaria la definición de mensajería específica para cada sistema que se desee interconectar.

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.

5.3 Modelo de casos de uso del sistema

El Modelo de casos de uso del sistema está compuesto por actores y casos de uso. 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

Capítulo 2: Características del Sistema

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.

Actores del sistema

Actor	Descripción
Evaluador	Es el técnico encargado de aplicar las pruebas y emitir un sumario con el resultado de la misma.
Fisiatra	Es el especialista que analiza el sumario de la prueba y emite resumen final de la consulta.
Administrador	Es el encargado de gestionar los nomencladores.

5.3.1 Diagrama de Casos de Uso 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. [20]

Capítulo 2: Características del Sistema

5.3.2 Listado de Casos de Usos

CU - 1	Realizar Test Fisiatría
Actor	Evaluadora
Descripción	El caso de uso inicia cuando el actor accede a la opción Realizar Test, el sistema brinda la posibilidad de introducir los datos para realizar la prueba el actor introduce los datos de la test, el sistema crea el sumario, el caso de uso termina.
Referencia	RF 1

CU - 2	Actualizar Sumario Fisiatría
Actor	Evaluadora
Descripción	El caso de uso inicia cuando el actor accede a la opción Actualizar Datos del sumario, el sistema muestra los datos del sumario y brinda la posibilidad de escribir las observaciones de la evaluadora, el sistema actualiza los datos del sumario, el caso de uso termina.
Referencia	RF 2

CU - 3	Crear Resumen Fisiatría
---------------	-------------------------

Capítulo 2: Características del Sistema

Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear resumen, el sistema brinda la posibilidad de introducir los datos para crear el resumen, el actor introduce los datos, el sistema crea el resumen, el caso de uso termina.
Referencia	RF 3.1

CU - 4	Modificar Resumen Fisiatría
Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor selecciona un resumen y accede a la opción Modificar Datos del resumen, 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 del sumario, el caso de uso termina.
Referencia	RF 3.3, CU- 6

CU - 5	Ver datos del Resumen Fisiatría
Actor	Fisiatra

Capítulo 2: Características del Sistema

Descripción	El caso de uso inicia cuando el actor selecciona un resumen y accede a la opción de Ver datos del resumen, el sistema muestra los datos del resumen, el caso de uso termina.
Referencia	RF 3.2, CU-6

CU - 6	Buscar Resumen Fisiatría
Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor accede a la opción Buscar Resumen Fisiatría, 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 3.4

CU - 7	Listar pacientes atendidos
Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor accede a la opción Listar pacientes atendidos, el sistema muestra el listado, el

Capítulo 2: Características del Sistema

	caso de uso termina.
Referencia	RF 5

CU - 8	Generar Reporte de Signos de alarma neuromotores
Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de Signos de alarma neuromotores, el sistema consulta la entidad Resumen Fisiatría y genera el reporte de información. El actor selecciona la opción de imprimir el reporte, el sistema imprime los datos del reporte, el caso de uso termina.
Referencia	RF 5.1

CU - 9	Generar Reporte de Asimetría postural de los miembros
Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de Asimetría postural de los miembros, el sistema consulta la entidad Resumen Fisiatría y genera el reporte de información. El actor selecciona la opción de imprimir el reporte, el sistema imprime los datos del reporte,

Capítulo 2: Características del Sistema

	el caso de uso termina.
Referencia	RF 5.2

CU - 10	Generar Reporte de Tono Pasivo.
Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de Tono Pasivo, el sistema consulta la entidad Resumen Fisiatría y genera el reporte de información. El actor selecciona la opción de imprimir el reporte, el sistema imprime los datos del reporte, el caso de uso termina.
Referencia	RF 5.3

CU - 11	Generar Reporte de Postura y Actividad motora espontánea
Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de Postura y Actividad motora espontánea, el sistema consulta la entidad Resumen Fisiatría y genera el reporte de información. El actor selecciona la opción de imprimir el reporte, el sistema

Capítulo 2: Características del Sistema

	imprime los datos del reporte, el caso de uso termina.
Referencia	RF 5.4

CU - 12	Generar Reporte de Reflejos Primarios
Actor	Fisiatra
Descripción	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de Reflejos Primarios, el sistema consulta la entidad Resumen Fisiatría y genera el reporte de información. El actor selecciona la opción de imprimir el reporte, el sistema imprime los datos del reporte, el caso de uso termina.
Referencia	RF 5.5

CU - 13	Gestionar Configuración Test Fisiatría
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear Configuración Test Fisiatría, Modificar Configuración Test Fisiatría, Listar Configuración Test Fisiatría, Eliminar Configuración Test Fisiatría. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de la Configuración Test Fisiatría, mostrando las pantallas

Capítulo 2: Características del Sistema

	correspondientes para cada una de estas funcionalidades.
Referencia	RF 6

CU - 14	Gestionar Edad Equivalente
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar o Eliminar una Edad Equivalente El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de una Edad Equivalente, mostrando las pantallas correspondientes para cada una de estas funcionalidades.
Referencia	RF 7

CU - 15	Gestionar Rangos Percentiles
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar o Eliminar un Rango Percentil. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de un Rango Percentil, mostrando las pantallas correspondientes para cada una de estas

Capítulo 2: Características del Sistema

	funcionalidades.
Referencia	RF 8

CU - 16	Gestionar Percentil Total
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar o Eliminar un Percentil Total. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de un Percentil Total, mostrando las pantallas correspondientes para cada una de estas funcionalidades.
Referencia	RF 9

CU - 17	Gestionar Puntaje de Escala Grueso y Fino
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar o Eliminar un Puntaje de Escala Grueso y Fino. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de un Puntaje de Escala Grueso y Fino, mostrando las pantallas

Capítulo 2: Características del Sistema

	correspondientes para cada una de estas funcionalidades.
Referencia	RF 10

CU - 18	Gestionar Edad
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar o Eliminar una Edad. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de una Edad, mostrando las pantallas correspondientes para cada una de estas funcionalidades.
Referencia	RF 11

CU - 19	Gestionar Asociaciones de Operaciones
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar, Eliminar una Asociación de Operaciones. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de una Asociación de Operaciones, mostrando las pantallas correspondientes para cada una de estas funcionalidades.

Capítulo 2: Características del Sistema

Referencia	RF 12
-------------------	--------------

CU - 20	Gestionar DQM
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear DQM, Modificar DQM, Listar DQM, Eliminar DQM. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de un DQM, mostrando las pantallas correspondientes para cada una de estas funcionalidades.
Referencia	RF 13

CU - 21	Gestionar Habilidad
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar o Eliminar una Habilidad. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de una Habilidad, mostrando las pantallas correspondientes para cada una de estas funcionalidades.
Referencia	RF 14

Capítulo 2: Características del Sistema

CU - 22	Gestionar Actividad Espontánea
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar o Eliminar una Actividad Espontánea. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de una Actividad Espontánea, mostrando las pantallas correspondientes para cada una de estas funcionalidades.
Referencia	RF 15

CU - 23	Gestionar Ítems
Actor	Administrador
Descripción	El caso de uso inicia cuando el actor accede a la opción Crear, Modificar, Listar o Eliminar un Ítems. El sistema le brinda la posibilidad de Crear, modificar, Listar y Eliminar los datos de Ítems, mostrando las pantallas correspondientes para cada una de estas funcionalidades.
Referencia	RF 16

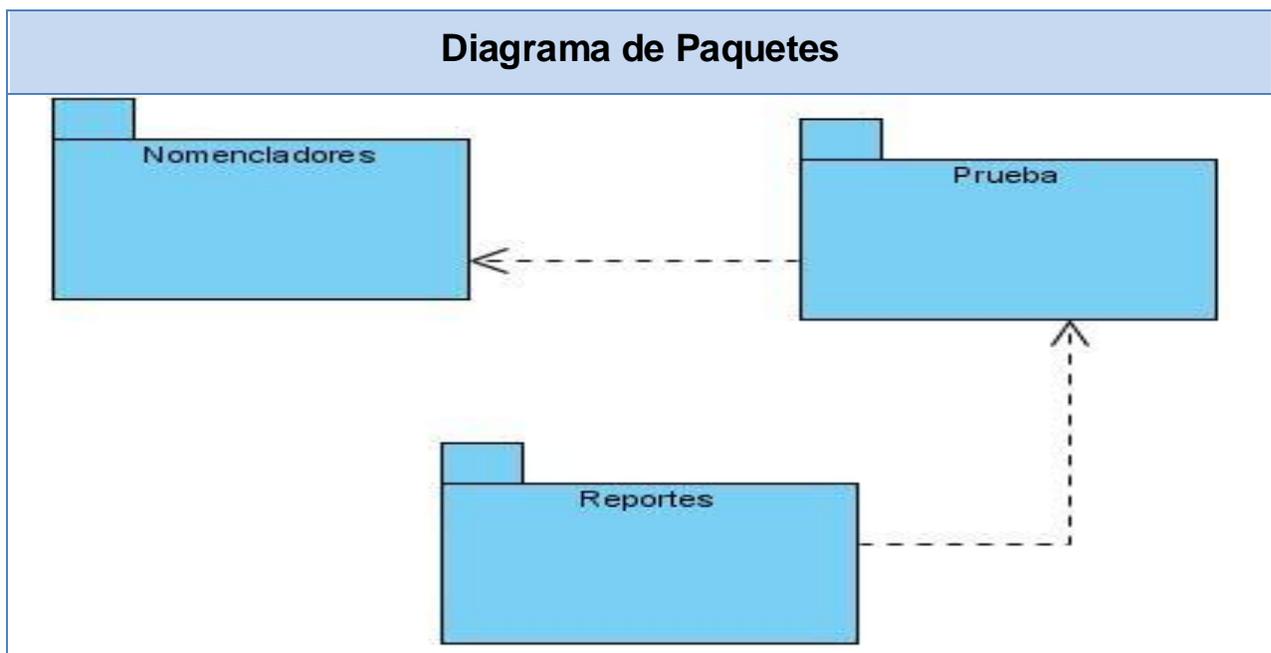
Organización de los Elementos: Casos de Uso en Paquetes.

Capítulo 2: Características del Sistema

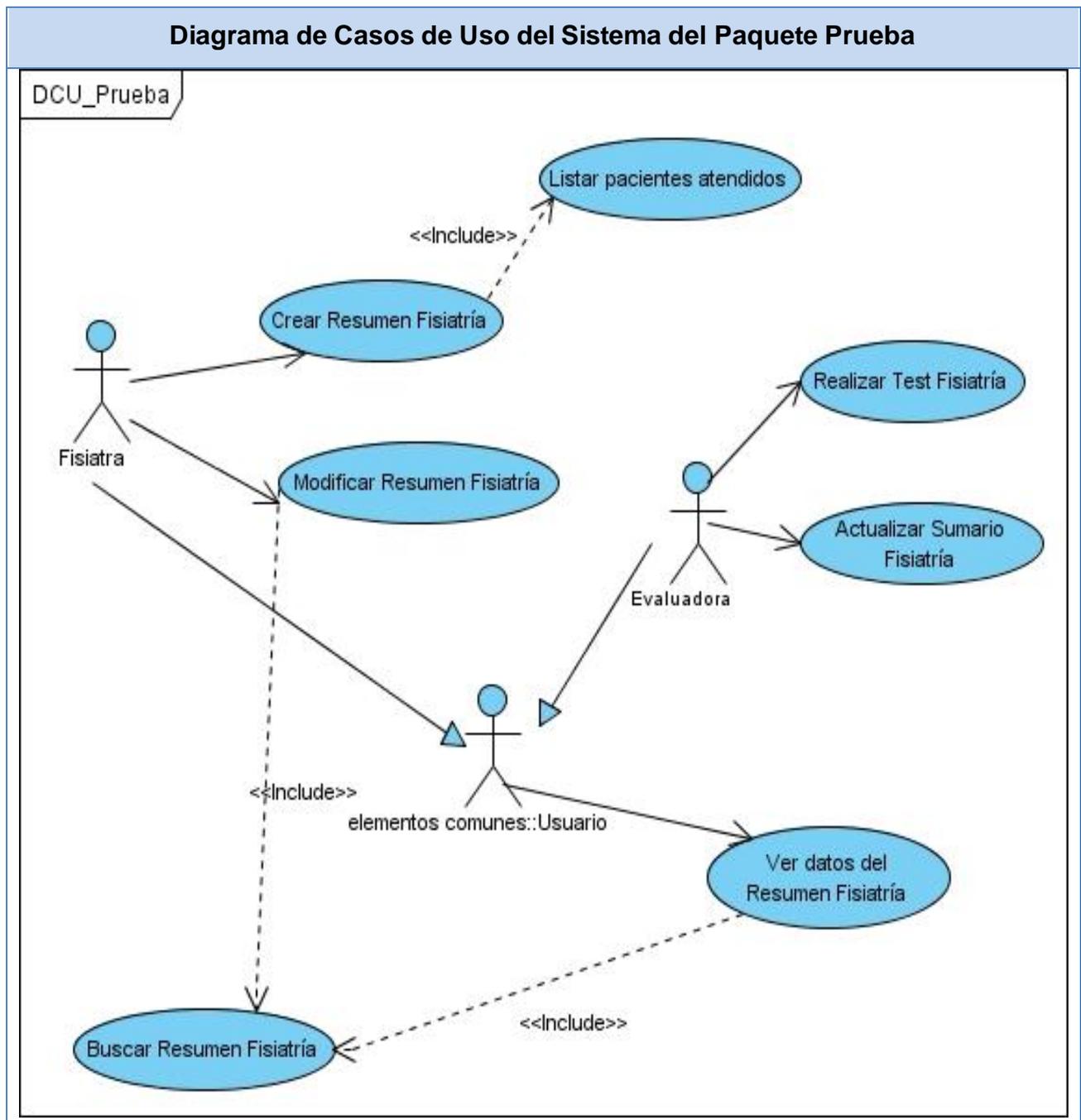
Los paquetes son usados para organizar y manipular la complejidad de los modelos cuando estos alcanzan un tamaño considerable, agrupando los casos de uso en subconjuntos más pequeños. Este ofrece una vista más detallada del modelo de casos de uso del sistema.

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.



Capítulo 2: Características del Sistema



Capítulo 2: Características del Sistema

Diagrama de Casos de Uso del Sistema del Paquete Reportes

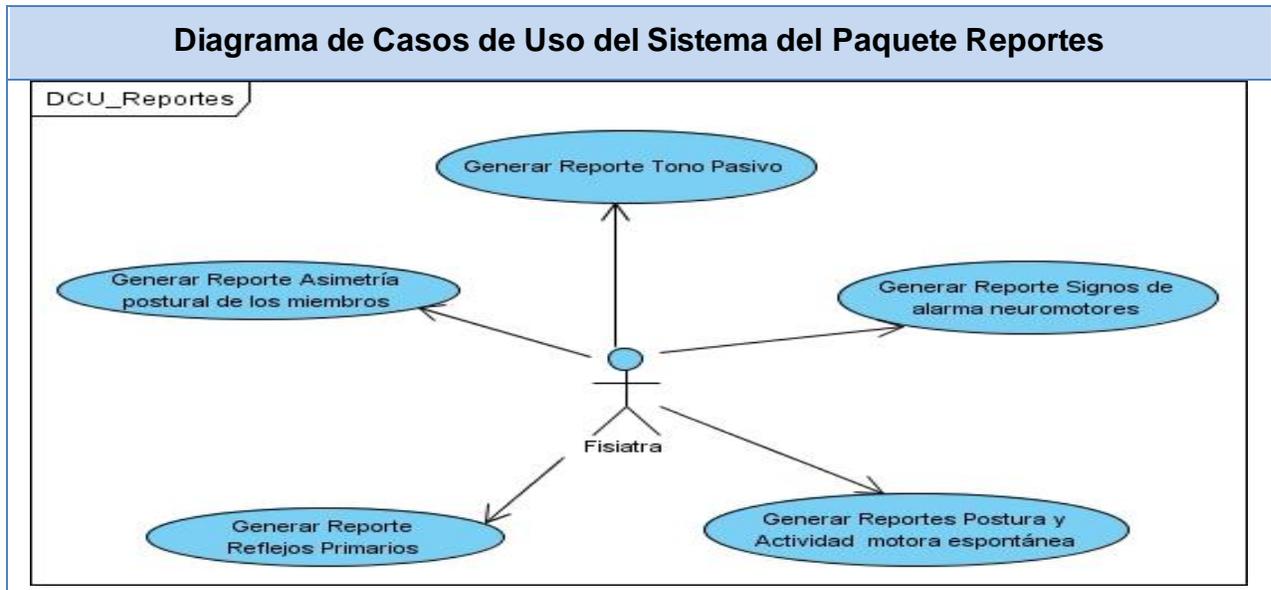
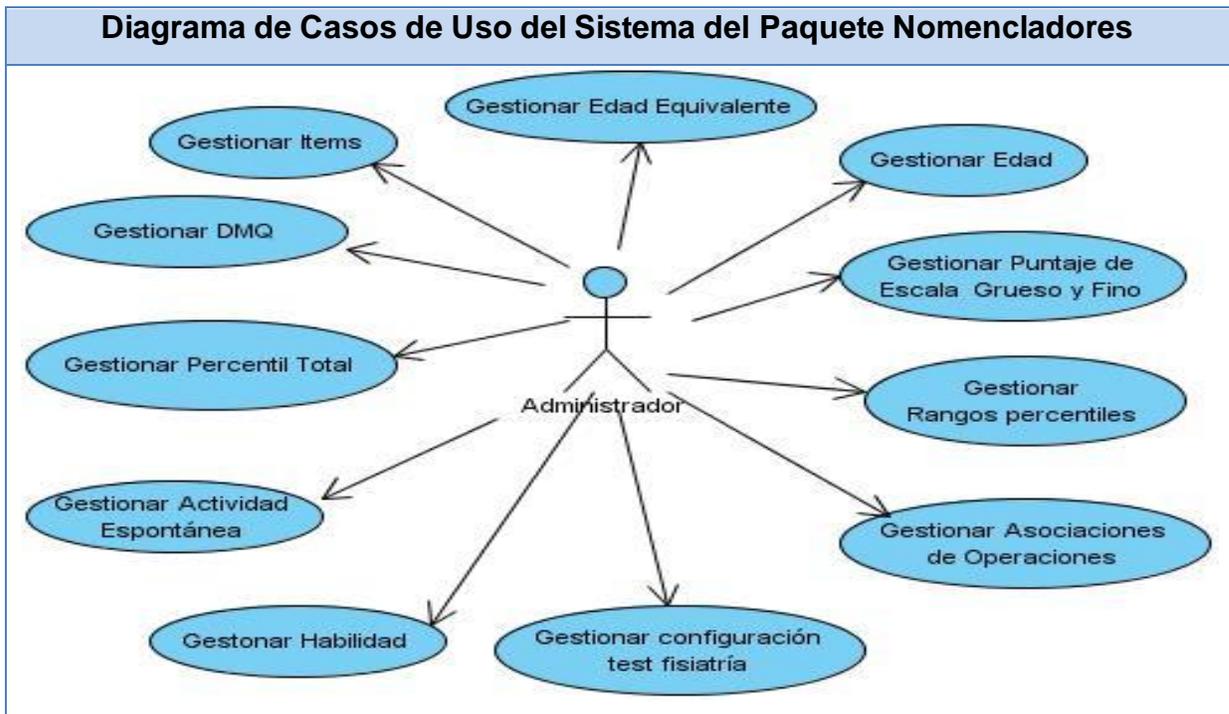


Diagrama de Casos de Uso del Sistema del Paquete Nomencladores



Capítulo 2: Características del Sistema

5.3.3 Casos de uso por ciclos

Código	Caso de Uso	Ciclo	Descripción
CU-1	Realizar Test Fisiatría	1	Se decide desarrollar en el primer ciclo por ser una actividad fundamental a la hora de atender a los pacientes.
CU-2	Actualizar Sumario Fisiatría	1	Se decide desarrollar en el primer ciclo por ser una actividad fundamental a la hora de atender a los pacientes.
CU-3	Crear Resumen Fisiatría	1	Se decide desarrollar en el primer ciclo por ser una actividad fundamental a la hora de atender a los pacientes.
CU-4	Modificar Resumen Fisiatría	1	Se decide desarrollar en el primer ciclo por ser una actividad fundamental a la hora de atender a los pacientes.
CU-5	Ver datos del Resumen Fisiatría	1	Se decide desarrollar en el primer ciclo por ser una actividad fundamental a la hora de atender a los pacientes.
CU-6	Buscar Resumen Fisiatría	1	Se decide desarrollar en el primer ciclo por ser una actividad

Capítulo 2: Características del Sistema

			fundamental a la hora de atender a los pacientes.
CU-7	Listar pacientes atendidos	1	Se decide desarrollar en el primer ciclo por ser una actividad fundamental a la hora de atender a los pacientes.
CU-8	Generar Reporte de Signos de alarma neuromotores	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-9	Generar Reporte de Asimetría postural de los miembros	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-10	Generar Reporte de Tono Pasivo Limitado o exagerado	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-11	Generar Reporte de Postura y Actividad motora espontánea.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-12	Generar Reporte de Reflejos Primarios.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a

Capítulo 2: Características del Sistema

			los casos de uso críticos.
CU-13	Gestionar Configuración Test Fisiatría.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-14	Gestionar Edad Equivalente.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-15	Gestionar Rangos Percentiles.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-16	Gestionar Percentil Total.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-17	Gestionar puntaje de escala grueso y fino.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-18	Gestionar Edad.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a

Capítulo 2: Características del Sistema

			los casos de uso críticos.
CU-18	Gestionar Asociaciones de Operaciones.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-19	Gestionar DMQ.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-20	Gestionar Habilidad.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-21	Gestionar Actividad Espontánea.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.
CU-22	Gestionar Ítems.	2	Se decide desarrollar en el segundo ciclo por ser un caso de uso secundario ya que sirve de apoyo a los casos de uso críticos.

5.3.4 Casos de uso expandidos

Realizar Test Fisiatría

CASO DE USO:	Realizar Test Fisiatría	
Resumen:	El caso de uso inicia cuando el actor accede a la opción Realizar Test, el sistema brinda la posibilidad de introducir los datos para realizar la prueba el actor introduce los datos de la test, el sistema crea el sumario, el caso de uso termina.	
Complejidad:	Alta	
Prioridad:	1	
Precondiciones:	El paciente tiene que tener creado el expediente	
REFERENCIAS		
Actores:	Evaluadora	
Requisitos:	5	
Entidades:	Sumario, Coeficiente de Desarrollo Motor, Rangos Percentiles, Edad Equivalente, Puntaje de Escala, Escala Motora Gruesa y Fina.	
Casos de Uso:	Realizar Test Fisiatría	
FLUJO NORMAL DE EVENTOS		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el actor accede a la opción Realizar Test.		

Capítulo 2: Características del Sistema

	<p>2. Muestra las pruebas a aplicar.</p> <ul style="list-style-type: none"> • Aceptar crear test. • Cancelar operación. Ver Alternativa 1: “Cancelar operación.”
3. Introduce los datos del test.	
4. Selecciona la opción de aceptar crear sumario.	
	<p>5. Valida los datos. Si hay datos incompletos, ver Alternativa 2: “Existen datos incompletos”.</p> <p>Si hay datos incorrectos, ver Alternativa 3: “Existen datos incorrectos.”</p>
	6. Crea el sumario.
	7. Muestra el mensaje de información “Se ha creado el sumario.”
	8. El caso de uso termina.
FLUJOS ALTERNOS	
Alternativa 1. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de Cancelar operación.	

Capítulo 2: Características del Sistema

	2. Regresa a la vista anterior.
	3. El caso de uso termina.
Alternativa 2. "Existen datos incompletos."	
Acción del Actor	Respuesta del Sistema
	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.
Alternativa 3. "Existen datos incorrectos."	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje de error "Existen campos escritos incorrectamente, por favor, rectifique estos datos."
	2. Muestra un indicador sobre los campos incorrectos.
Poscondiciones	Se creó un sumario.

Actualizar Sumario Fisiatría.

CASO DE USO:	Actualizar Sumario Fisiatría.
---------------------	-------------------------------

Capítulo 2: Características del Sistema

Resumen:	El caso de uso inicia cuando el actor accede a la opción Actualizar Datos del sumario, el sistema muestra los datos del sumario y brinda la posibilidad de escribir las observaciones de la evaluadora, el sistema actualiza los datos del sumario, el caso de uso termina.	
Complejidad:	Media	
Prioridad:	1	
Precondiciones:	Para actualizar los datos de un sumario, se le deben haber aplicado las pruebas al paciente y creado el sumario.	
REFERENCIAS		
Actores:	Evaluadora	
Requisitos:	6	
Entidades:	Sumario	
Casos de Uso:	Actualizar Sumario Fisiatría	
FLUJO NORMAL DE EVENTOS		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el actor accede a la opción Actualizar datos del sumario.		

Capítulo 2: Características del Sistema

	<p>2. Muestra los todos los datos de las pruebas.</p> <p>Brinda la posibilidad de actualizarlo, introduciendo nuevos datos.</p> <p>y permite:</p> <ul style="list-style-type: none"> • Aceptar las modificaciones. • Cancelar la operación. Ver Alternativa 1: “Cancelar operación.”
3. Introduce sus observaciones y selecciona la opción de aceptar las modificaciones.	
	4. Actualiza los datos del sumario.
	5. Muestra el mensaje de información “Se han actualizado los cambios sobre el Sumario.”
	6. El caso de uso termina.
FLUJOS ALTERNOS	
Alternativa 1. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema
4. Selecciona la opción de Cancelar operación.	
	5. Regresa a la vista anterior.

Capítulo 2: Características del Sistema

	6. El caso de uso termina.
Poscondiciones	Se actualizó el sumario.

Crear Resumen Fisiatría

CASO DE USO:	Crear Resumen Fisiatría	
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear resumen, el sistema brinda la posibilidad de introducir los datos para crear el resumen, el actor introduce los datos, el sistema crea el resumen, el caso de uso termina.	
Complejidad:	Alta	
Prioridad:	1	
Precondiciones:	Se debe tener un listado de pacientes atendidos.	
REFERENCIAS		
Actores:	Fisiatra	
Requisitos:	7.1	
Entidades:	Resumen, Examen Físico.	
Casos de Uso:	Crear Resumen Fisiatría	
FLUJO NORMAL DE EVENTOS		
Acción del Actor	Respuesta del Sistema	

Capítulo 2: Características del Sistema

1. El caso de uso inicia cuando el actor accede a la opción Crear Resumen.	
	<ol style="list-style-type: none">2. Muestra los datos predeterminados.3. Brinda la posibilidad de introducir todos los datos del resumen.<ul style="list-style-type: none">• Aceptar crear Resumen.• Cancelar operación. Ver Alternativa 1: “Cancelar operación.”
4. Introduce los datos del resumen.	
5. Selecciona la opción de aceptar Crear Resumen.	
	<ol style="list-style-type: none">6. Valida los datos. Si hay datos incompletos, ver Alternativa 2: “Existen datos incompletos”. Si hay datos incorrectos, ver Alternativa 3: “Existen datos incorrectos.”
	<ol style="list-style-type: none">7. Crea el resumen.
	<ol style="list-style-type: none">8. Muestra el mensaje de información “Se ha creado el sumario.”
	<ol style="list-style-type: none">9. El caso de uso termina.
FLUJOS ALTERNOS	

Capítulo 2: Características del Sistema

Alternativa 1. "Cancelar operación."	
Acción del Actor	Respuesta del Sistema
7. Selecciona la opción de Cancelar operación.	
	8. Regresa a la vista anterior.
	9. El caso de uso termina.
Alternativa 2. "Existen datos incompletos."	
Acción del Actor	Respuesta del Sistema
	3. Muestra el mensaje de error "Existen campos vacíos que son obligatorios, por favor, complete estos datos."
	4. Muestra un indicador sobre los campos vacíos.
Alternativa 3. "Existen datos incorrectos."	
Acción del Actor	Respuesta del Sistema
	3. Muestra el mensaje de error "Existen campos escritos incorrectamente, por favor, rectifique estos datos."
	4. Muestra un indicador sobre los campos incorrectos.

Capítulo 2: Características del Sistema

Poscondiciones	Se creó un resumen.
-----------------------	---------------------

Modificar Resumen Fisiatría

CASO DE USO:	Modificar Resumen Fisiatría.	
Resumen:	El caso de uso inicia cuando el actor selecciona un resumen y accede a la opción Modificar Datos del resumen, 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 del sumario, el caso de uso termina.	
Complejidad:	Media	
Prioridad:	1	
Precondiciones:	Para modificar los datos de un resumen esta debe haber sido seleccionada.	
REFERENCIAS		
Actores:	Fisiatra	
Requisitos:	7.3	
Entidades:	Resumen, Examen Físico.	
Casos de Uso:	Modificar Resumen Fisiatría	
FLUJO NORMAL DE EVENTOS		
Acción del Actor	Respuesta del Sistema	

Capítulo 2: Características del Sistema

7. El caso de uso inicia cuando el actor selecciona un resumen y accede a la opción Modificar datos del resumen.	
	8. Muestra los todos los datos brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes y permite: <ul style="list-style-type: none">• Aceptar las modificaciones.• Cancelar la operación. Ver Alternativa 1: “Cancelar operación.”
9. Modifica los datos que necesita y selecciona la opción de aceptar las modificaciones.	
	10. Valida los datos. Si hay datos incompletos, ver Alternativa 2: “Existen datos incompletos.”. Si hay datos incorrectos, ver Alternativa 3: “Existen datos incorrectos.”
	11. Actualiza los datos del resumen.
	12. Muestra el mensaje de información “Se han actualizado los cambios sobre el resumen.”
	13. El caso de uso termina.

Capítulo 2: Características del Sistema

FLUJOS ALTERNOS	
Alternativa 1. “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema
10. Selecciona la opción de Cancelar operación.	
	11. Regresa a la vista anterior.
	12. El caso de uso termina.
Alternativa 2. “Existen datos incompletos.”	
Acción del Actor	Respuesta del Sistema
	5. Muestra el mensaje de error “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
	6. Muestra un indicador sobre los campos vacíos.
Alternativa 3. “Existen datos incorrectos.”	
Acción del Actor	Respuesta del Sistema
	5. Muestra el mensaje de error “Existen campos escritos incorrectamente, por favor, rectifique estos datos.”

Capítulo 2: Características del Sistema

	6. Muestra un indicador sobre los campos incorrectos.
Poscondiciones	Se modificó un resumen.

Ver datos del Resumen Fisiatría

CASO DE USO:	Ver datos del Resumen Fisiatría.
Resumen:	El caso de uso inicia cuando el actor selecciona un resumen y accede a la opción de Ver datos del resumen, el sistema muestra los datos del resumen, el caso de uso termina.
Complejidad:	Media
Prioridad:	2
Precondiciones:	Para ver los datos de un resumen, este debe estar seleccionada.
REFERENCIAS	
Actores:	Fisiatra Evaluadora
Requisitos:	7.2
Entidades:	Resumen, Examen Físico.
Casos de Uso:	Visualizar Remen Fisiatría
FLUJO NORMAL DE EVENTOS	

Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona un resumen y accede a la opción de Ver datos del resumen.	
	2. Muestra los datos del resumen y permite: <ul style="list-style-type: none">• Salir de la vista actual
3. Selecciona la opción de salir de la vista actual.	
	4. Muestra la vista anterior.
	5. El caso de uso termina.
Poscondiciones	Se vieron los datos del resumen de fisiatría.

Buscar Resumen Fisiatría

CASO DE USO:	Buscar Resumen Fisiatría
Resumen:	El caso de uso inicia cuando el actor accede a la opción Buscar Resumen Fisiatría, 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.
Complejidad:	media

Capítulo 2: Características del Sistema

Prioridad:	2
Precondiciones:	
REFERENCIAS	
Actores:	Fisiatra
Requisitos:	7.4
Entidades:	Resumen, Examen Físico.
Casos de Uso:	Buscar Resumen Fisiatría.
FLUJO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor accede a la opción Buscar Resumen.	
	<p>2. Brinda la posibilidad de introducir los criterios elementales de búsqueda:</p> <ul style="list-style-type: none"> • Número de expediente • Nombre del paciente • Fecha <p>y permite:</p> <ul style="list-style-type: none"> • Buscar Resumen dado criterios. • Cancelar operación. Ver Alternativa 2: “Cancelar operación.”

Capítulo 2: Características del Sistema

3. Introduce los datos que considera como criterios para realizar una búsqueda y selecciona la opción de Buscar Resumenes dado criterios.	
	4. Busca los datos de resúmenes que cumplen con los criterios de búsqueda.
	5. Si no se encuentra ningún resumen que cumpla con los criterios de búsqueda. Ver Alternativa 1: “No se encuentra información que cumpla con los criterios de búsqueda.”
	6. Muestra un listado de los resúmenes que cumplen con los criterios de búsqueda.
	7. El caso de uso termina.
Alternativa 1: “No se encuentra información que cumpla con los criterios de búsqueda.”	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje de información “No se encontró información que cumpla con los criterios de búsqueda.”
	2. Regresa al paso 2 del Flujo Normal de Eventos.
Alternativa 2: “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Características del Sistema

13. Selecciona la opción de Cancelar operación.	
	14. Regresa a la vista anterior.
	15. El caso de uso termina.
Poscondiciones	Se buscó resumen dado criterios.

Listar pacientes atendidos

CASO DE USO:	Listar pacientes atendidos.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Listar pacientes atendidos, el sistema muestra el listado, el caso de uso termina.
Complejidad:	Media
Prioridad:	1
Precondiciones:	Se le deben haber realizado todas las pruebas y creado el sumario.
REFERENCIAS	
Actores:	Fisiatra
Requisitos:	7.5
Entidades:	
Casos de Uso:	Listar pacientes atendidos
FLUJO NORMAL DE EVENTOS	

Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor accede a la opción Listar pacientes atendidos.	
	2. Muestra el listado de los pacientes y permite: <ul style="list-style-type: none"> • Cancelar operación. Ver Alternativa 1: "Cancelar operación."
	3. El caso de uso termina.
Alternativa 1. "Cancelar operación."	
Acción del Actor	Respuesta del Sistema
4. Selecciona la opción de Cancelar operación.	
	5. Regresa a la vista anterior.
	6. El caso de uso termina.
Poscondiciones	Se generó un listado.

Capítulo 2: Características del Sistema

En este capítulo, después de realizar un estudio sobre la situación actual del proceso de evaluación de la calidad del neurodesarrollo llevado a cabo en el Hospital pediátrico William Soler se propone realizar una aplicación informática que mejore el funcionamiento de dicha programa. Para ello se obtuvo el modelo del negocio, definiéndose actores, trabajadores y procesos en el mismo, así como los requerimientos funcionales y no funcionales del sistema basados en las necesidades del cliente.

CAPÍTULO 3

ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se realiza el análisis y diseño de la propuesta de solución, desarrollándose los diagramas de clases del análisis de los casos de usos definidos, así como los diagramas de interacción correspondientes a cada uno de ellos, refinando estos diagramas en el diseño para lograr 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 gobiernan su diseño. Aporta además una visión abstracta de alto nivel, postergando el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. Establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema.

Para la realización del sistema “Módulo de Fisiatría 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 comunique 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 Análisis

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo, que sólo se interesa por los requisitos funcionales.

En el flujo de trabajo de análisis se trabaja sobre los aspectos internos del sistema a desarrollar, se pueden estructurar los requisitos de manera que faciliten su comprensión, su preparación, su modificación y en general su mantenimiento. En el modelo de análisis se obtiene un mayor poder expresivo y una mayor formalización y se puede considerar como una primera aproximación al modelo de diseño.

Clases del Análisis

Una clase describe un grupo de objetos con estructura y comportamiento común. Las estructuras o propiedades de la clase se conocen como atributos y el comportamiento como operaciones.

Una clase define uno o más objetos, donde los objetos pertenecen a la clase, teniendo características comunes. (21)

Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación-composición, generalización-especialización y tipos asociativos. RUP propone clasificar a las clases en:

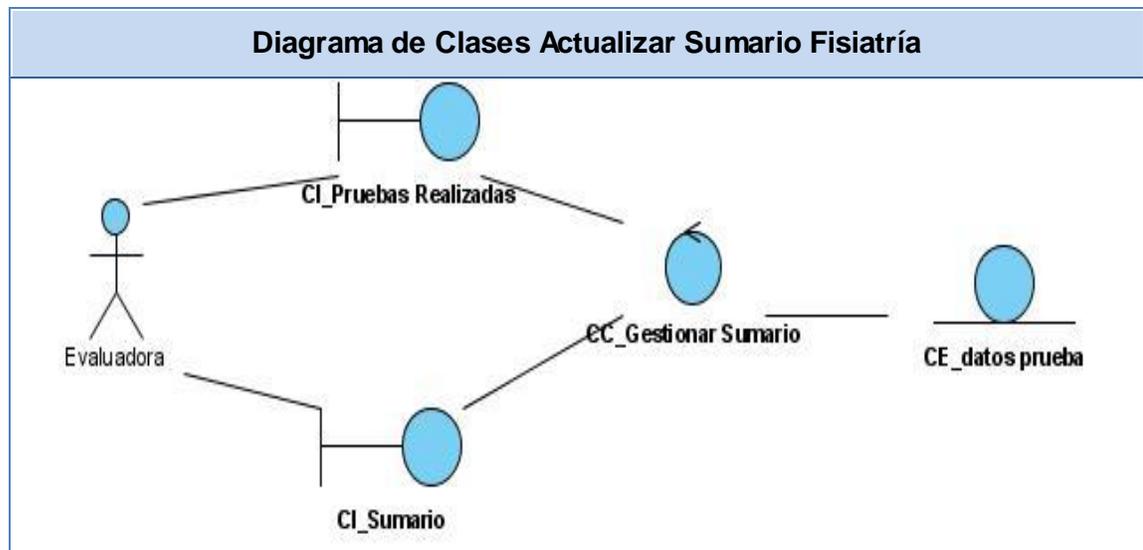
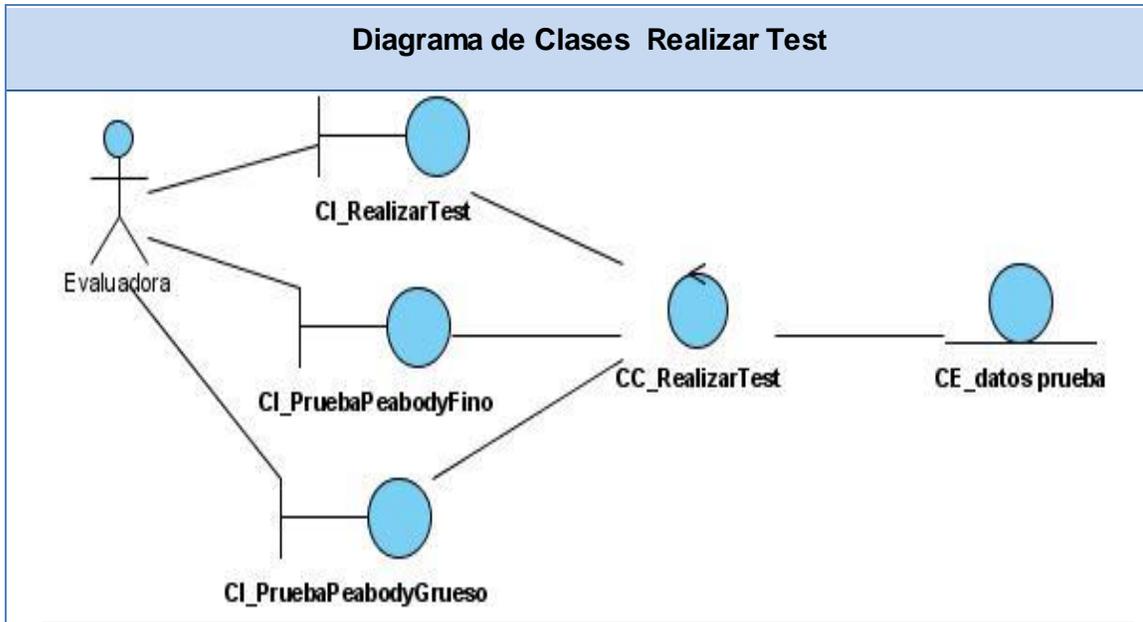
Clase Entidad: Modelan información que posee larga vida y que es a menudo persistente. La fuente principal de obtención son las clases entidades del negocio y el glosario de términos que se ha ido elaborando. Además, modelan el comportamiento asociado a algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.

Clase Control: Coordinan el trabajo de uno o unos pocos casos de uso, modelando las actividades de los objetos que implementan la funcionalidad del caso de uso, por lo que definen el flujo de control y las transacciones dentro de un caso de uso delegando el trabajo a otros objetos.

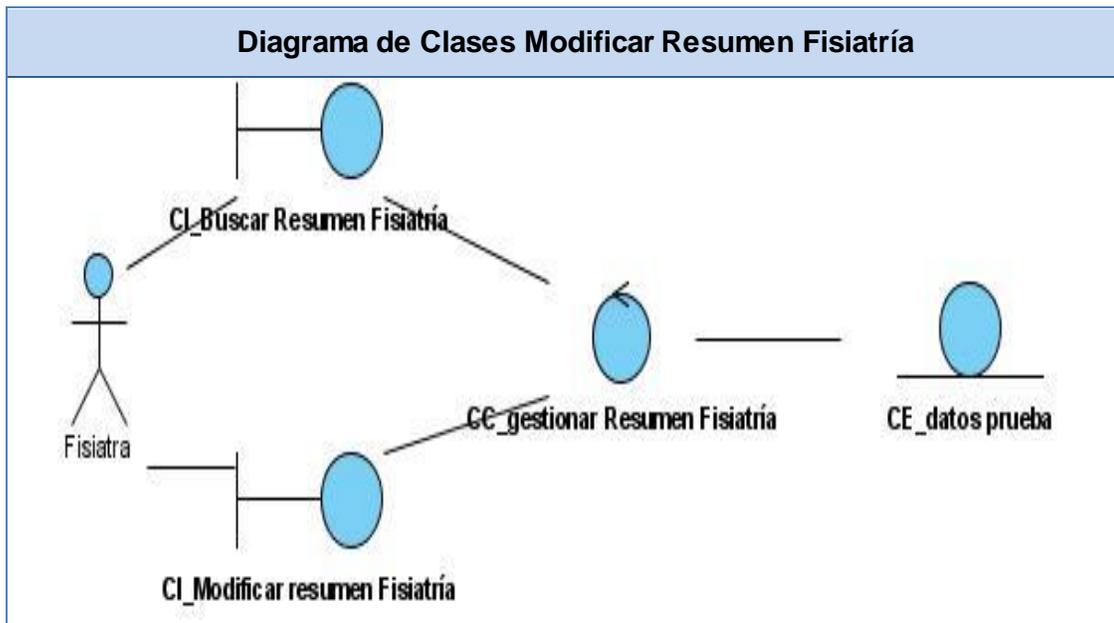
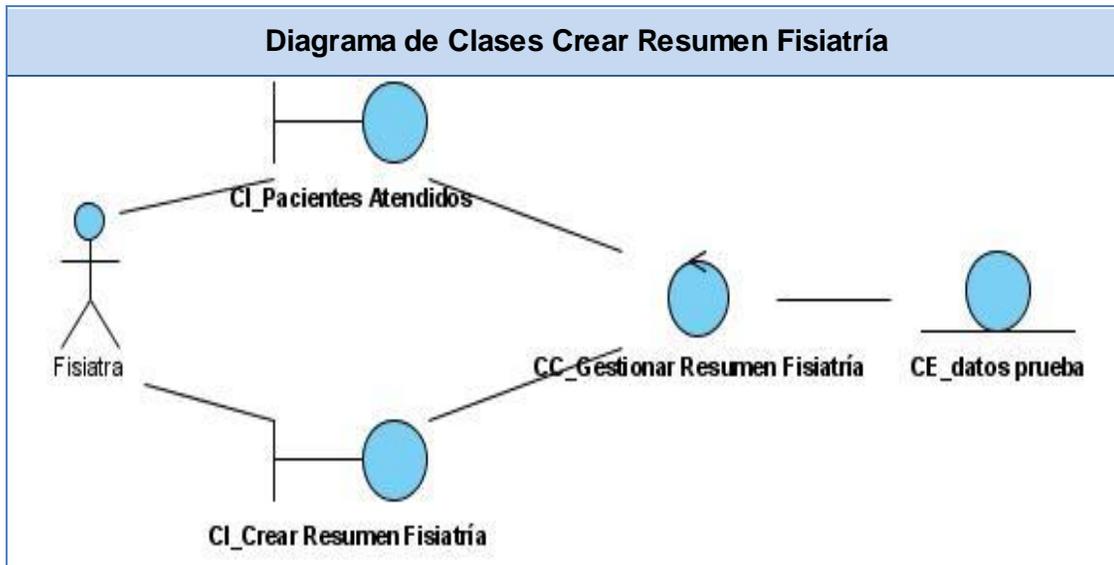
Clase interfaz: Se utilizan para modelar la interacción entre el sistema y sus actores. Esta interacción a menudo implica recibir información y peticiones de los usuarios y sistemas externos. Cada clase de interfaz debe asociarse con al menos un actor.

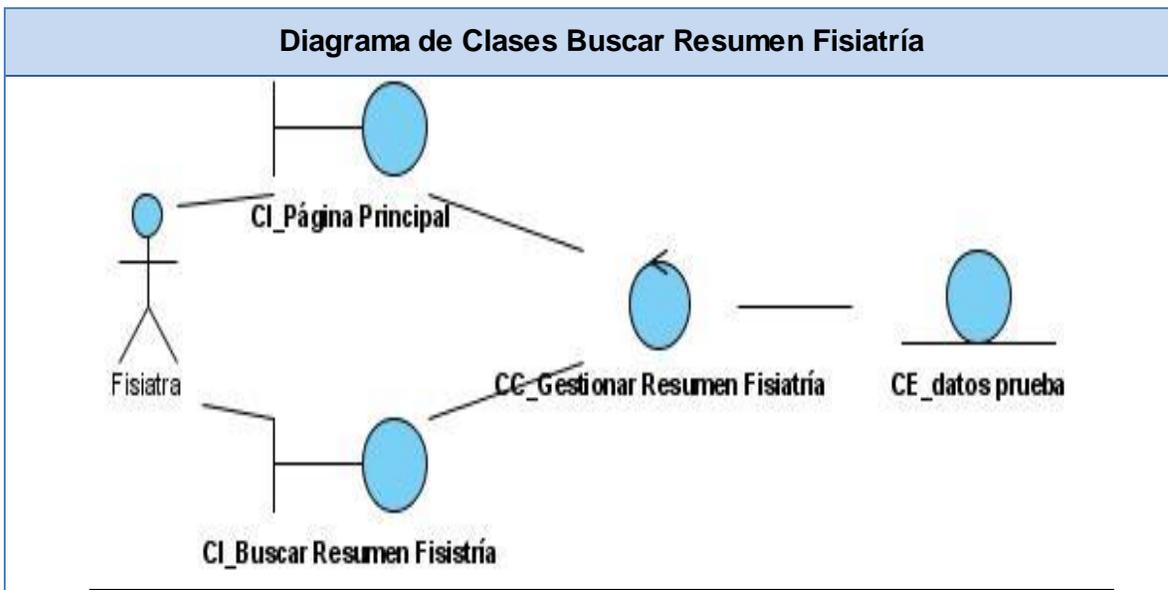
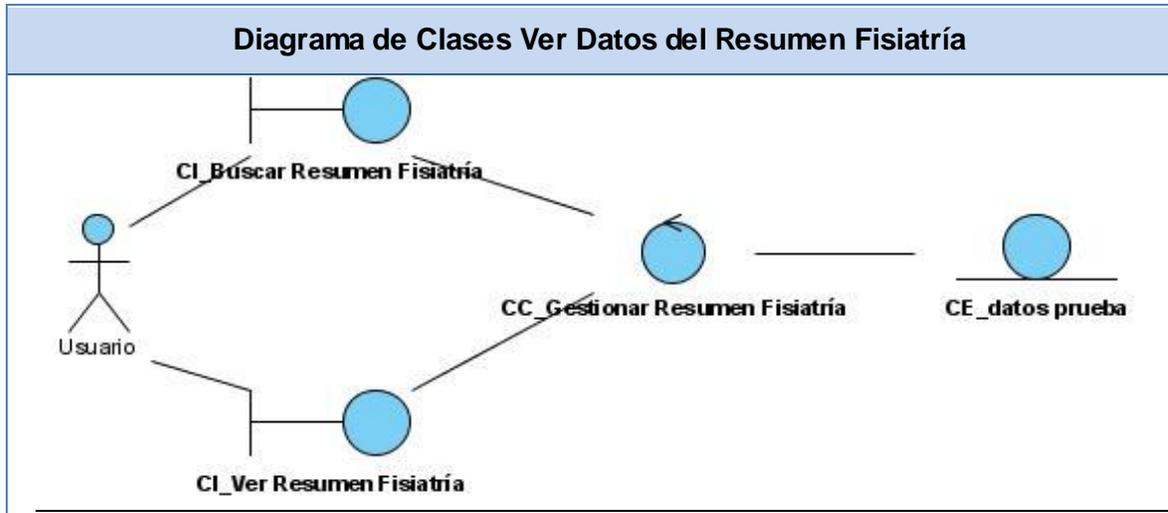
Capítulo 3: Análisis y Diseño del Sistema

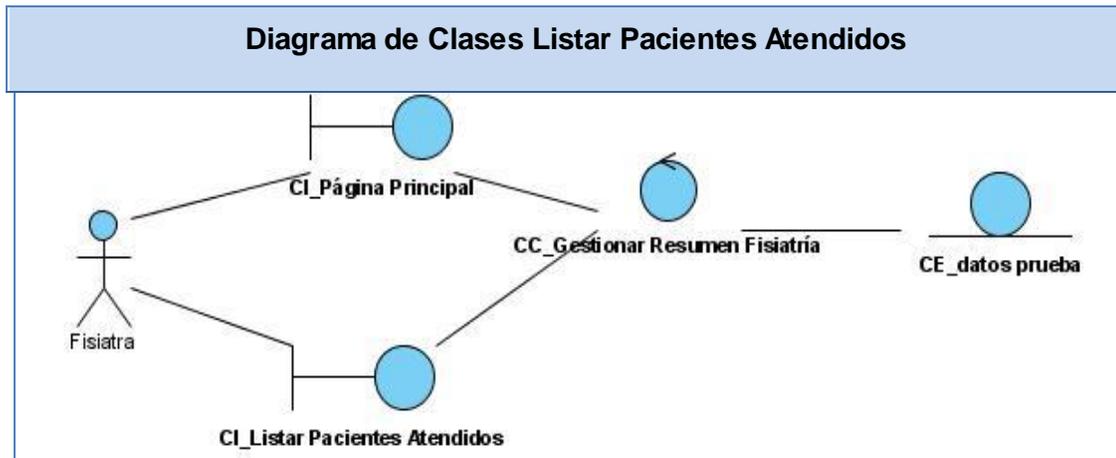
A continuación se muestran algunos diagramas de clases del análisis.



Capítulo 3: Análisis y Diseño del Sistema







3.3 Modelo de Diseño

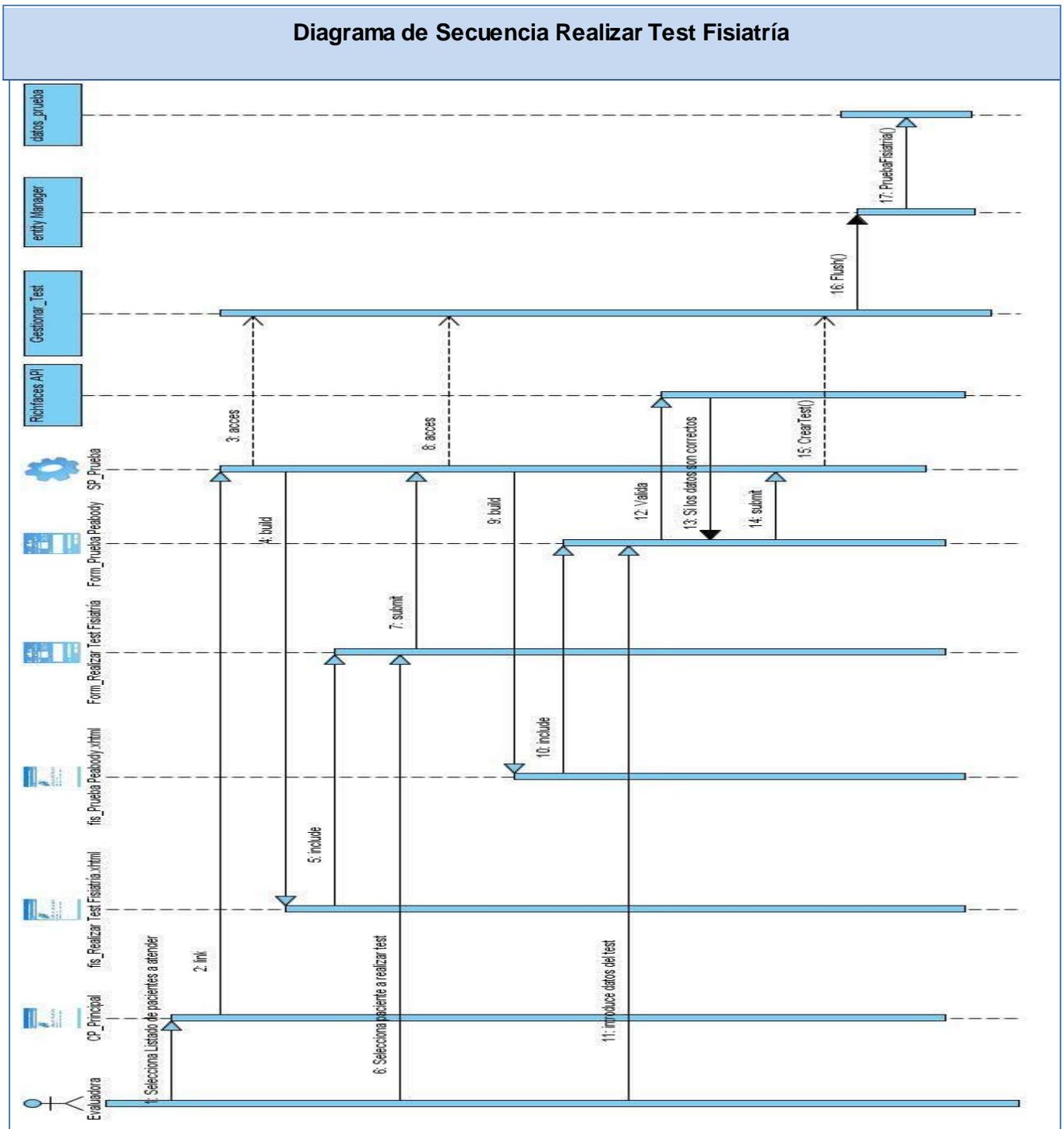
El diseño tiene como propósito formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución. Se basa en que el modelo del sistema es una colección de objetos que cooperan entre sí, donde cada objeto es una instancia de una clase en una jerarquía de clases. Además impone una estructura del sistema que se debe conservar lo más exacto posible. Es la entrada al flujo de trabajo de implementación.

Diagrama de Interacción (Secuencia)

Un diagrama de interacción es un artefacto del diseño que permite asignar las responsabilidades a cada clase, muestra: actores, objetos de las clases, eventos, orden de los eventos y pueden contener además notas y restricciones. Estos diagramas son muy útiles para visualizar, especificar, construir y documentar la dinámica entre dos objetos. Hay dos tipos de diagrama de interacción: Diagramas de Secuencia y Diagramas de Colaboración.

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes. Gráficamente, un diagrama de secuencia es una tabla que representa objetos, dispuestos a lo largo del eje X, y mensajes, ordenados según suceden en el tiempo, a lo largo del eje Y. (22)

A continuación se muestran el diagrama de secuencia Realizar Test Fisiatría, para obtener más información sobre los diagramas restantes consultar. Ver [Anexo 2](#).

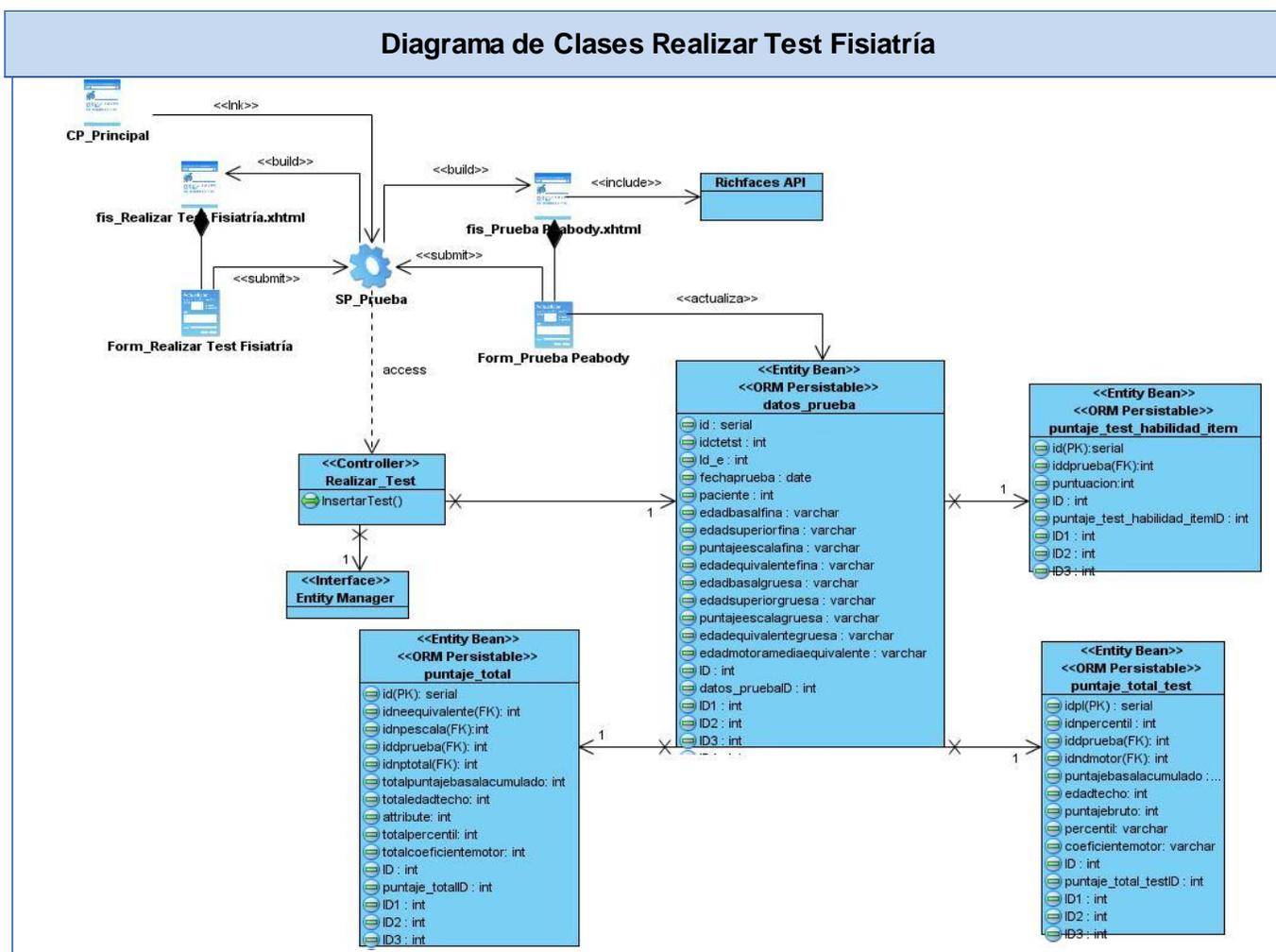


Capítulo 3: Análisis y Diseño del Sistema

Diagrama de Clases del Diseño

Los diagramas de clases de diseño exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema. 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 Realizar Test Fisiatría, para obtener más información sobre los diagramas restantes consultar [Ver Anexo 1](#).



Capítulo 3: Análisis y Diseño del Sistema

Descripción de las clases del diseño

Nombre de la Clase: datos_prueba	
Tipo de Clase: Modelo	
Atributo	Tipo
Id	serial
idctetst	int
Id_e	int
fechaprueba	date
paciente	int
edadbasalfina	varchar
edadsuperiorfina	varchar
puntajeescalafina	varchar
edadequivalentefina	varchar
edadbasalgruesa	varchar
edadsuperiorgruesa	varchar
Puntajeescalagruesa	varchar
edadequivalentegruesa	varchar
edadmotoramediaequivalente	varchar

Capítulo 3: Análisis y Diseño del Sistema

Nombre de la Clase: examen_fisico_general	
Tipo de Clase: Modelo	
Atributo	Tipo
Id	serial
idefisico	int
idamotora	int
idsaneuronales	int
facies	varchar
pdecubito	varchar
psedestacion	varchar
pbipedestacion	varchar
marcha	varchar
seguimientoocular	varchar
rcocleopalbebral	varchar
tacuello	varchar
apmiembros	varchar
presenteapmiembros	varchar
movimientosanormales	varchar

Capítulo 3: Análisis y Diseño del Sistema

presentemovimientosanormales	varchar
tpanguloaductores	varchar
limitadoanguloaductores	varchar
tptalonoreja	varchar
limitadotalonoreja	varchar
tpangulopopliteo	varchar
limitadoangulopopliteo	varchar
tpmaniobrabunganda	varchar
limitadomaniobrabunganda	varchar
tpflexinventral	varchar
limitadoflexinventral	varchar
balanceopie	varchar
rlcabeza	varchar
taposicionsentado	varchar
tacontrolcabeza	varchar
tasentado	varchar
tamiembrotronco	varchar
rpmarchaautomatica	varchar

Capítulo 3: Análisis y Diseño del Sistema

rpprensiondedos	varchar
ausenteprensiondedos	varchar
rprespuestatraccion	varchar
ausenterespuestatraccion	varchar

Nombre de la Clase: puntaje_test_habilidad_item	
Tipo de Clase: Modelo	
Atributo	Tipo
id	serial
iddprueba	int
puntuacion	int

Nombre de la Clase: puntaje_total_test	
Tipo de Clase: Modelo	
Atributo	Tipo
Id	serial
idnpercentil	int

Capítulo 3: Análisis y Diseño del Sistema

iddprueba	int
idndmotor	int
puntajebasalacumulado	int
edadtecho	int
puntajebruto	int
percentil	varchar
coeficientemotor	varchar

Nombre de la Clase: examen_fisico	
Tipo de Clase: Modelo	
Atributo	Tipo
id	Serial
iddprueba	int

Nombre de la Clase: signos_alarmas_neuronales	
Tipo de Clase: Modelo	
Atributo	Tipo
Id	serial

Capítulo 3: Análisis y Diseño del Sistema

idnaa	int
-------	-----

Nombre de la Clase: actividad_motora	
Tipo de Clase: Modelo	
Atributo	Tipo
Id	serial
idnamespontanea	int

Nombre de la Clase: puntaje_total	
Tipo de Clase: Modelo	
Atributo	Tipo
Id	serial
iddprueba	int
idneequivalente	int
idnpescala	int
idnptotal	int
totalpuntajebasalacumulado	int
totaledadtecho	int

Capítulo 3: Análisis y Diseño del Sistema

totalpuntajebruto	int
totalpercentil	varchar
totalcoeficientemotor	varchar

Nombre: Gestionar_Test	
Tipo de clase: Controladora	
Nombre:	InsertarTest()
Descripción:	Inserta las pruebas realizadas.

Nombre: Actualizar_Sumario	
Tipo de clase: Controladora	
Nombre:	ModificarSumario()
Descripción:	Modificar los datos de un sumario.

Nombre: Crear_Resumen	
Tipo de clase: Controladora	
Nombre:	InsertaResumen()
Descripción:	Insertar un resumen.

Capítulo 3: Análisis y Diseño del Sistema

Nombre: Pacientes_Atendidos	
Tipo de clase: Controladora	
Nombre:	Listar Pacientes():List
Descripción:	Lista los pacientes que ya se le han realizado las pruebas.

Nombre: fis_Modificar.xhtml	
Tipo de clase : Vista	
Descripción:	Se muestran los datos de un resumen para modificarlos.

Nombre: fis_Crear Resumen Fisiatría.xhtml	
Tipo de clase : Vista	
Descripción:	Se muestran los campos que debe llenar para crear el Resumen.

Nombre: fis_Pacientes Atendidos.xhtml	
Tipo de clase : Vista	
Descripción:	Se muestra un listado con los pacientes que se le han realizado las pruebas.

Capítulo 3: Análisis y Diseño del Sistema

Nombre: fis_Buscar.xhtml	
Tipo de clase : Vista	
Descripción:	Se muestran los criterios de búsqueda para buscar un resumen.

Nombre: fis_RealizarTest Fisiatría.xhtml	
Tipo de clase : Vista	
Descripción:	Se muestran los datos de de las pruebas.

Nombre: fis_Ver Resumen.xhtml	
Tipo de clase : Vista	
Descripción:	Se muestran los datos de un resumen.

Nombre: fis_PruebasRealizadas.xhtml	
Tipo de clase : Vista	
Descripción:	Se muestran un listado de los pacientes que ya se le han realizado las pruebas.

Capítulo 3: Análisis y Diseño del Sistema

Nombre: fis_Sumario.xhtml	
Tipo de clase : Vista	
Descripción:	Se muestran los campos para llenar un sumario.

En este capítulo se realizaron los diagramas de clases del análisis y los de interacción, que facilitaron una primera aproximación al modelo del diseño. Pudiendo 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 implementación. Se muestran los estándares de diseño y codificación y se explica la seguridad del sistema.

4.1 Modelo de datos

Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un ordenador; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas en el ordenador se debe realizar un modelo de datos.

El modelo de datos es el artefacto más importante concebido por RUP dentro del proceso que incluye generar la base de datos por esta metodología. Este modelo proporciona una representación visual y física de los datos persistentes del sistema, que en el futuro serán la base de datos. Se obtiene a partir del diagrama de clases persistentes y su forma se expresa mediante un diagrama de UML, siendo sus elementos esenciales del modelo son las entidades, los atributos y las relaciones entre las entidades. A continuación se muestra el modelo de datos del sistema.

Descripción de las tablas del Modelo de Datos.

Nombre de la Clase: datos_prueba		
Descripción: Tabla que almacena la información final de las pruebas.		
Atributo	Tipo	Descripción
iddatosprueba	serial	En este atributo se almacena el identificador de la prueba.
idctetst	int	Representa la configuración de test. Este atributo es llave foránea que se absorbe de la tabla configuracion_test.
ld_e	int	Representa el expediente del paciente. Este atributo es llave foránea que se absorbe de la tabla expediente.
fechapruueba	date	En este atributo se almacena la fecha en que se aplica el test.
edadbasalfina	varchar	En este atributo se almacena la edad basal al aplicar el test del Peabody Motor Fino.
edadsuperiorfina	varchar	En este atributo se almacena la edad superior al aplicar el test del Peabody Motor Fino.
puntajeescalafina	varchar	En este atributo se almacena el

Capítulo 4: Implementación

		valor del Puntaje de Escala Fino.
edadequivalentefina	varchar	En este atributo se almacena el valor de la Edad Equivalente Fina.
edadbasalgruesa	varchar	En este atributo se almacena la edad basal al aplicar el test del Peabody Motor Grueso.
edadsuperiorgruesa	varchar	En este atributo se almacena la edad superior al aplicar el test del Peabody Motor Grueso.
Puntajeescalagruesa	varchar	En este atributo se almacena el valor del Puntaje de Escala Grueso.
edadequivalentegruesa	varchar	En este atributo se almacena el valor de la Edad Equivalente Gruesa.
edadmotoramediaequivalente	varchar	En este atributo se almacena el valor de la Edad Media Equivalente.

Nombre de la Clase: examen_fisico_general

Descripción: Tabla que almacena la información general del Resumen de Fisiatría.

Atributo	Tipo	Descripción
----------	------	-------------

Capítulo 4: Implementación

Id	serial	En este atributo se almacena el identificador del resumen de Fisiatría.
idefisico	int	Representa el examen físico. Este atributo es llave foránea que se absorbe de la tabla examen_fisico.
idamotora	int	Representa la actividad motora del paciente. Este atributo es llave foránea que se absorbe de la tabla actividad_motora.
idsaneuronales	int	Representa los signos alarmas neuronales del paciente. Este atributo es llave foránea que se absorbe de la tabla signos_alarmas_neuronales.
facies	varchar	En este atributo se almacena la facies del paciente (puede ser Normal o Patológica).
pdecubito	varchar	En este atributo se almacena una descripción de la Postura Decúbito.
psedestacion	varchar	En este atributo se almacena una descripción de la Postura Sedestación.
pbipedestacion	varchar	En este atributo se almacena una descripción de la Postura

Capítulo 4: Implementación

		Bipedestación.
marcha	varchar	En este atributo se almacena la marcha del paciente (puede ser Normal o Patológica).
seguimientoocular	varchar	En este atributo se almacena el Seguimiento Ocular del paciente (puede estar Presente o Ausente).
rcocleopalbebral	varchar	En este atributo se almacena el Reflejo Cocco-Palbebral del paciente (puede estar Presente o Ausente).
tacuello	varchar	En este atributo se almacena el Tónico Asimétrico Del Cuello del paciente (puede estar Presente o Ausente).
apmiembros	varchar	En este atributo se almacena la Asimetría postural de los miembros del paciente (puede estar Presente o Ausente).
presenteapmiembros	varchar	En este atributo se almacena una descripción de la Asimetría postural de los miembros del paciente en caso de que esta este presente.
movimientosanormales	varchar	En este atributo se almacena los Movimientos anormales del paciente

Capítulo 4: Implementación

		(puede estar Presente o Ausente).
presentemovimientosanormales	varchar	En caso que los movimientos anormales estén presentes se almacenas si son Temblores o Mioclonías u Otros
tpanguloaductores	varchar	En este atributo se almacena el Ángulo de los Aductores del paciente (puede ser de Evolución Normal, Limitado o Exagerado).
limitadoanguloaductores	varchar	En caso que el Ángulo de los Aductores su evolución sea Limitado se especifica si es más limitado a la derecha o más Limitado a la Izquierda o Ambos.
tptalonoreja	varchar	En este atributo se almacena el Talón-Oreja del paciente (puede ser de Evolución Normal, Limitado o Exagerado).
limitadotalonoreja	varchar	En caso que el Talón-Oreja su evolución sea Limitado se especifica si es más limitado a la derecha o más Limitado a la Izquierda o Ambos.
tpangulopopliteo	varchar	En este atributo se almacena el ángulo poplíteo del paciente (puede ser de Evolución Normal, Limitado o

Capítulo 4: Implementación

		Exagerado).
limitadoangulopopliteo	varchar	En caso que el ángulo poplíteo su evolución sea Limitado se especifica si es más limitado a la derecha o más Limitado a la Izquierda o Ambos.
tpmaniobrabunganda	varchar	En este atributo se almacena la Maniobra de la Bufanda del paciente (puede ser de Evolución Normal, Limitado o Exagerado).
limitadomaniobrabunganda	varchar	En caso que la Maniobra de la Bufanda su evolución sea Limitado se especifica si es más limitado a la derecha o más Limitado a la Izquierda o Ambos.
tpflexinventral	varchar	En este atributo se almacena la Flexión ventral del tronco del paciente (puede ser de Evolución Normal, Exagerada, Imposible).
limitadoflexinventral	varchar	En este atributo se almacena la Extensión del Tronco del paciente (puede ser de Evolución Normal, Exagerada).
balanceopie	varchar	En este atributo se almacena el Balanceo del pie del paciente (puede ser más amplio pie izquierdo o más

Capítulo 4: Implementación

		amplio pie derecho).
rlcabeza	varchar	En este atributo se almacena la Rotación lateral de la cabeza del paciente (puede ser más limitada a la Izquierda o más limitada a la Derecha).
taposicionsentado	varchar	En este atributo se almacena la reacción del paciente al llevarlo a la posición de sentado (puede ser normal, penosa, paso pasivo, caída hacia adelante imposible).
tacontrolcabeza	varchar	En este atributo se almacena el Control de la cabeza del paciente (puede ser evolución normal o ausente).
tasentado	varchar	En este atributo se almacena si el paciente se mantiene sentado (puede ser evolución Normal o pocos segundos o 30 segundos o más o fracasa).
tamiembrotronco	varchar	En este atributo se almacena el enderezamiento de los miembros inferiores y el tronco del paciente (puede ser Evolución Normal o Ausente, o Actitud en tijera o Espontáneo en opistótonos casa).

Capítulo 4: Implementación

rpmarchaautomatica	varchar	En este atributo se almacena la marcha automática del paciente (puede ser Evolución Normal o Ausente).
rpprensiondedos	varchar	En este atributo se almacena la presión de los dedos del paciente (puede ser Evolución Normal o Ausente).
ausenteprensiondedos	varchar	En caso que la presión de los dedos sea presente se especifica se es Asimetría Derecha o Asimetría Izquierda
rpresuestatraccion	varchar	En este atributo se almacena respuesta a la tracción del paciente (puede ser Evolución Normal o Ausente).
ausenterespuestatraccion	varchar	En caso que la respuesta a la tracción del paciente sea presente se especifica se es Asimetría Derecha o Asimetría Izquierda

Nombre de la Clase: puntaje_test_habilidad_item

Descripción: Tabla que almacena la información del puntaje del ítem por Habilidad.

Atributo	Tipo	Descripción
----------	------	-------------

Capítulo 4: Implementación

id	serial	En este atributo se almacena el identificador del puntaje del ítem con la Habilidad.
iddprueba	int	Representa la prueba. Este atributo es llave foránea que se absorbe de la tabla datos_prueba.
puntuacion	int	En este atributo se almacena la puntuación del ítem por habilidad.

Nombre de la Clase: puntaje_total_test

Descripción: Tabla que almacena la información del puntaje total del test.

Atributo	Tipo	Descripción
Id	serial	En este atributo se almacena el identificador del puntaje total del test.
idnpercentil	int	Representa al nomenclador percentil. Este atributo es llave foránea que se absorbe de la tabla n_percentil.
iddprueba	int	Representa la prueba. Este atributo es llave foránea que se absorbe de la tabla datos_prueba.

Capítulo 4: Implementación

idndmotor	int	Representa al nomenclador desarrollo motor. Este atributo es llave foránea que se absorbe de la tabla n_desarrollo_motor.
puntajebasalacumulado	int	En este atributo se almacena el puntaje basal acumulado del paciente.
edadtecho	int	En este atributo se almacena la edad techo del paciente.
puntajebruto	int	En este atributo se almacena el puntaje bruto del paciente.
percentil	varchar	En este atributo se almacena el valor del percentil total del paciente.
coeficientemotor	varchar	En este atributo se almacena el valor del coeficiente motor total del paciente.

Nombre de la Clase: examen_fisico

Descripción: Tabla que almacena los identificadores correspondientes a tuplas que se encuentran en otras tablas que guardan información perteneciente al examen físico.

Atributo	Tipo	Descripción
----------	------	-------------

Capítulo 4: Implementación

id	Serial	En este atributo se almacena el identificador del Examen Físico.
iddprueba	int	Representa la prueba. Este atributo es llave foránea que se absorbe de la tabla datos prueba.

Nombre de la Clase: signos_alarmas_neuronales

Descripción: Tabla que almacena los identificadores correspondientes a tuplas que se encuentran en otras tablas que guardan información perteneciente al nomenclador signos alarmas neuronales.

Atributo	Tipo	Descripción
Id	serial	En este atributo se almacena el identificador del los signos de alarma neuronales.
idnaa	int	Representa al nomenclador de los signos de alarma neuronales. Este atributo es llave foránea que se absorbe de la tabla n_asociacion_alteraciones

Nombre de la Clase: actividad_motora

Descripción: Tabla que almacena los identificadores correspondientes a tuplas que se encuentran en otras tablas que guardan información perteneciente al nomenclador actividad

motora.		
Atributo	Tipo	Descripción
Id	serial	En este atributo se almacena el identificador de la actividad motora.
idnamespontanea	int	Representa al nomenclador actividad motora espontánea. Este atributo es llave foránea que se absorbe de la tabla n_actividad_motora_espontanea.

Nombre de la Clase: puntaje_total		
Descripción: Tabla que almacena la información del puntaje total.		
Atributo	Tipo	
Id	Serial	En este atributo se almacena el identificador del puntaje total.
iddprueba	int	Representa la prueba. Este atributo es llave foránea que se absorbe de la tabla datos_prueba.
idneequivalente	int	Representa al nomenclador Edad Equivalente. Este atributo es llave foránea que se absorbe de la tabla

Capítulo 4: Implementación

		n_edad_equivalente.
idnpescala	int	Representa al nomenclador Puntaje Escala. Este atributo es llave foránea que se absorbe de la tabla n_puntaje_escalas.
idnptotal	int	Representa al nomenclador Percentil Total. Este atributo es llave foránea que se absorbe de la tabla n_percentil_total.
totalpuntajebasalacumulado	int	En este atributo se almacena el puntaje basal acumulado total.
totaledadtecho	int	En este atributo se almacena la edad techo total.
totalpuntajebruto	int	En este atributo se almacena el puntaje bruto total.
totalpercentil	varchar	En este atributo se almacena el percentil total.
totalcoeficientemotor	varchar	En este atributo se almacena el coeficiente de desarrollo motor total.

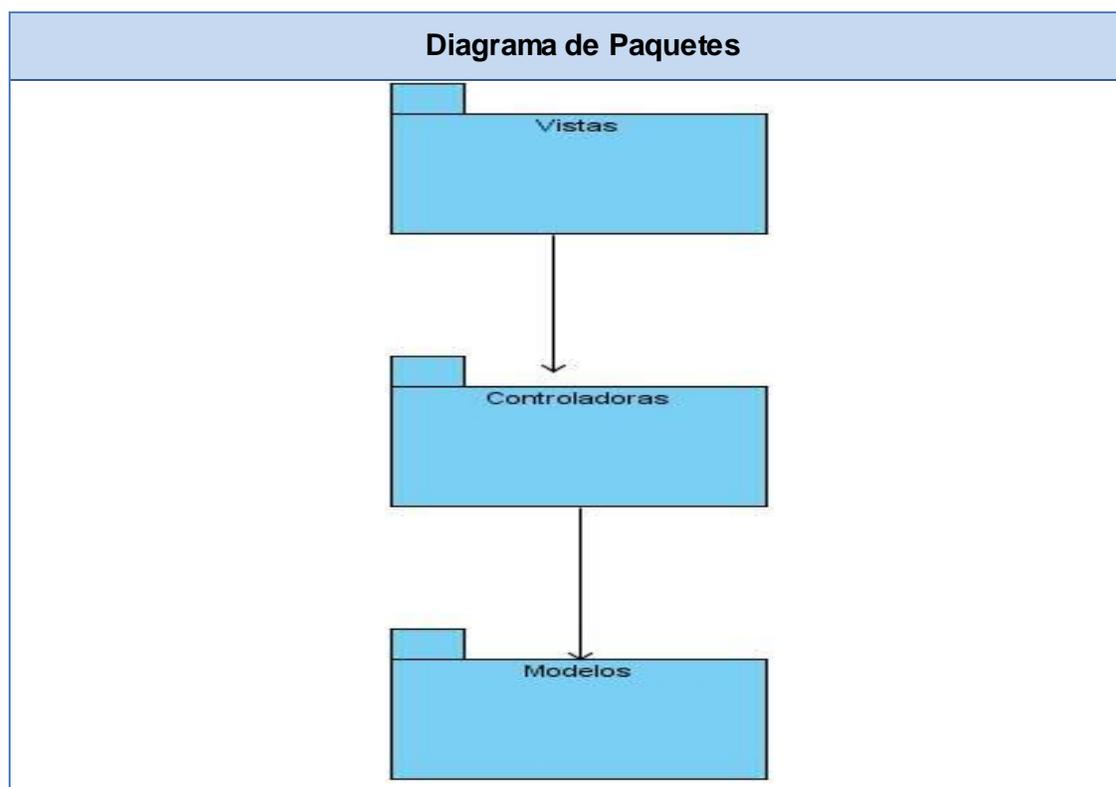
4.2 Modelo de implementación

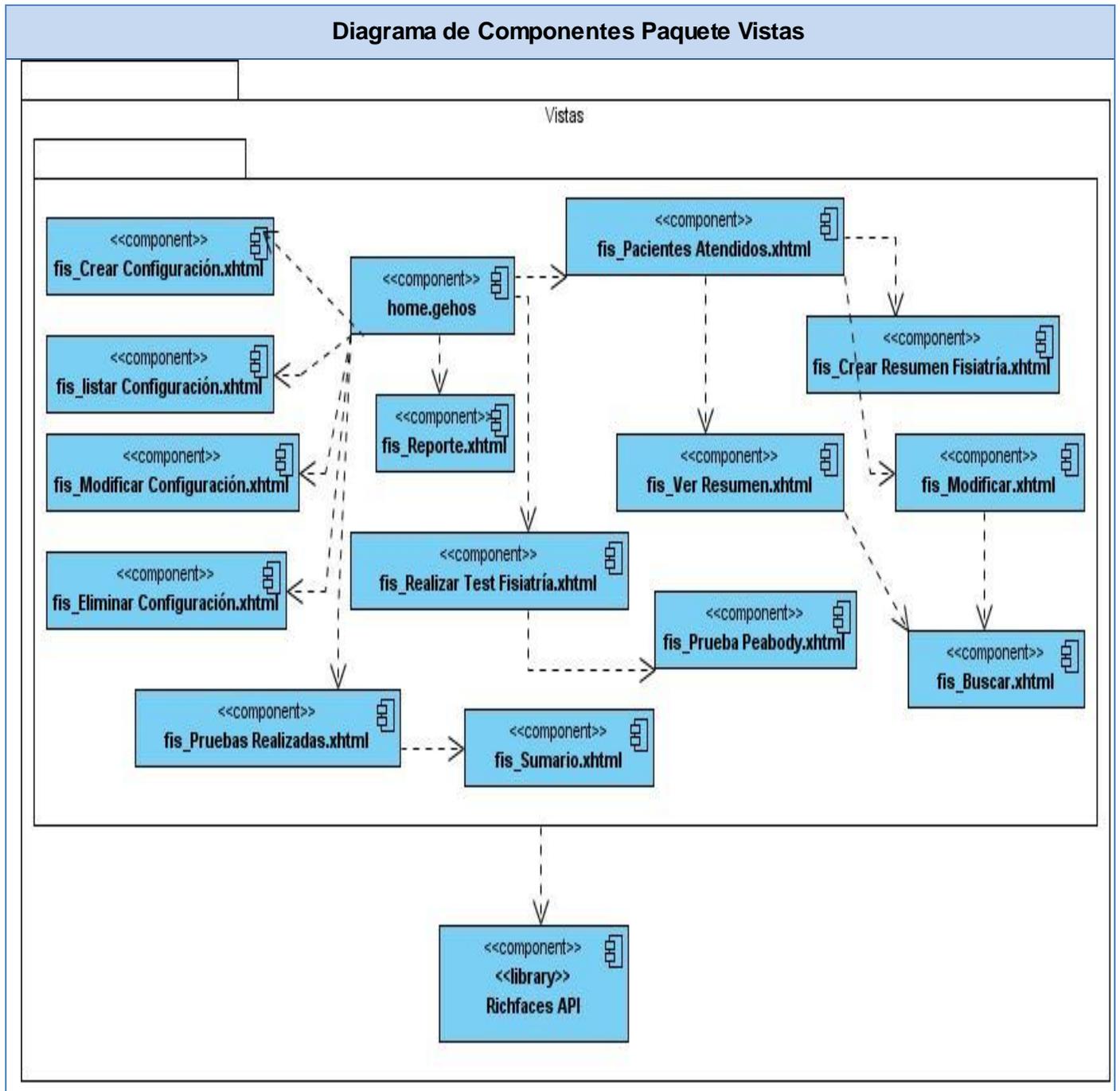
Los diagramas de despliegue y componente conforman lo que se conoce como modelo de implementación. Este describe los componentes a construir, su organización y la dependencia entre nodos físicos, en los que funcionará el sistema.

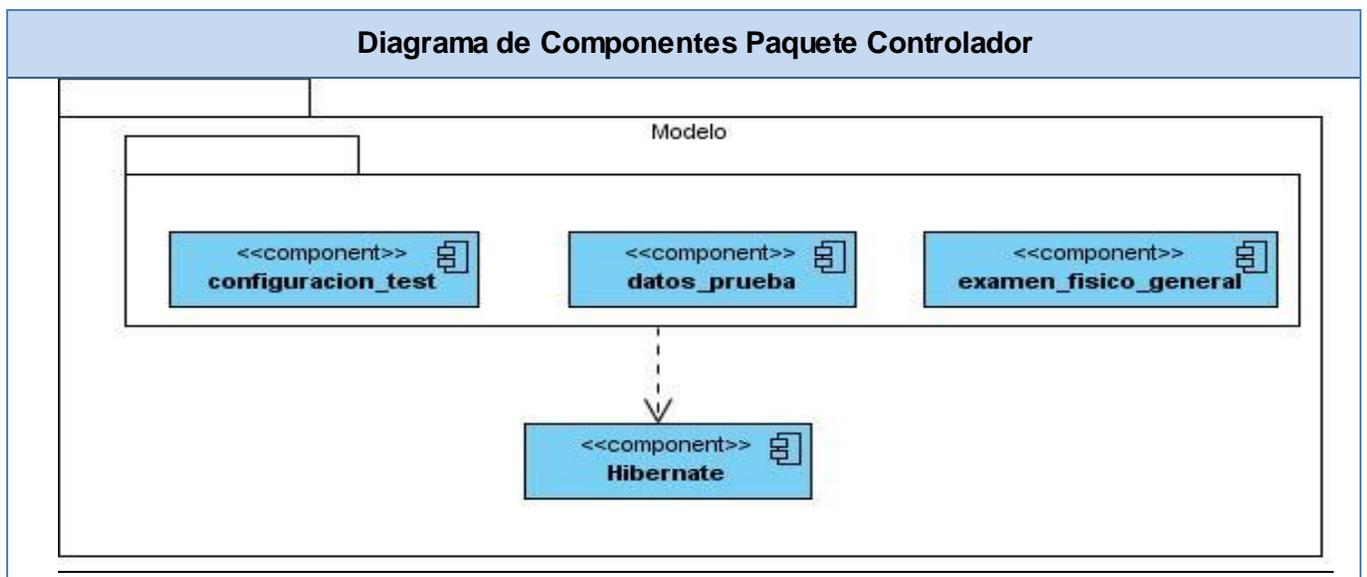
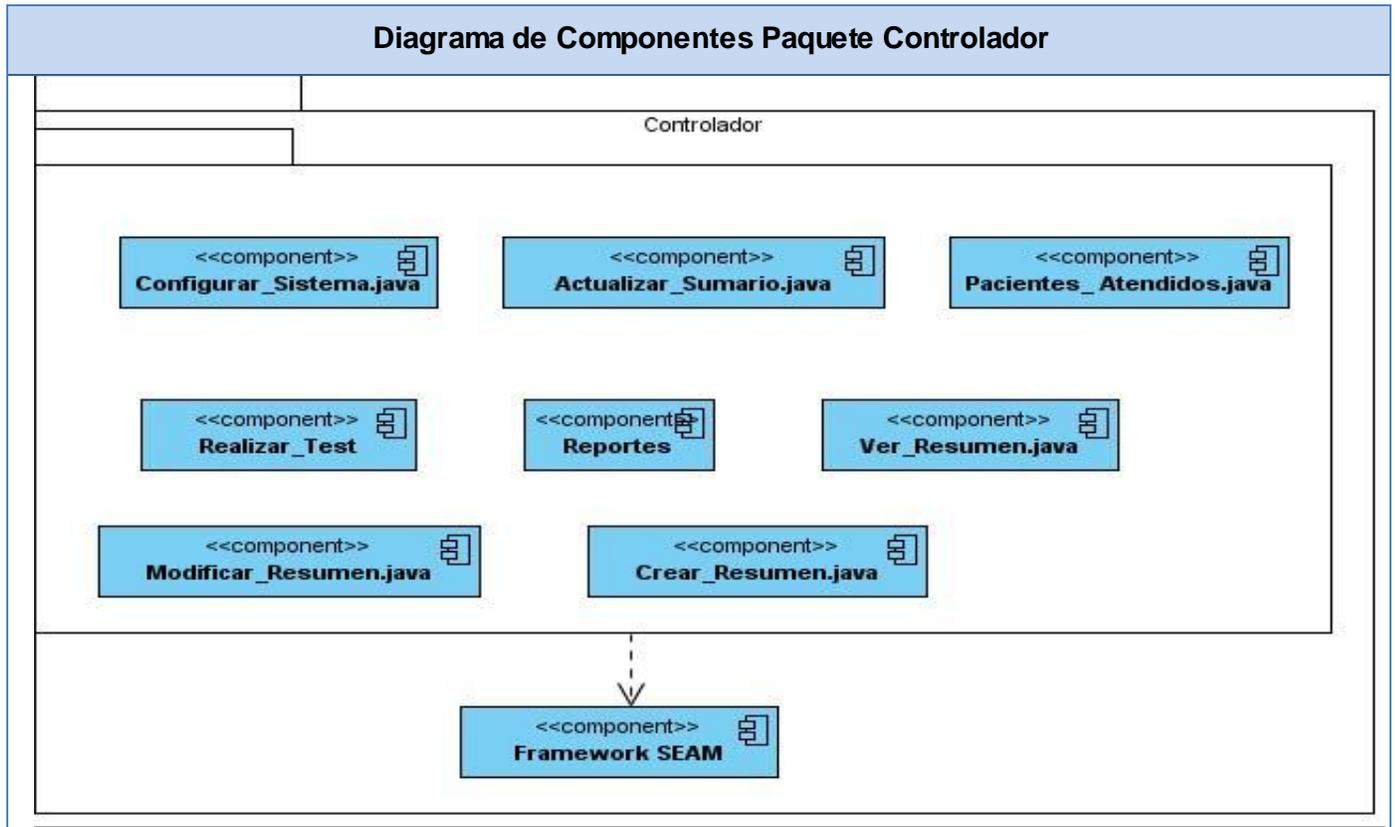
El modelo de implementación del Módulo de Fisiatría del Sistema de Evaluación del Neurodesarrollo en Niños expone una organización en capas, jerarquías de paquetes y subsistemas de implementación. Los mismos contienen componentes y sus relaciones, dividiendo el sistema en partes más manejables. Esto posibilita la reutilización, que se pueda implementar por separado y disminuye el impacto que pueda traer consigo un cambio.

4.2.1 Diagrama de componentes.

“Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.” (23)

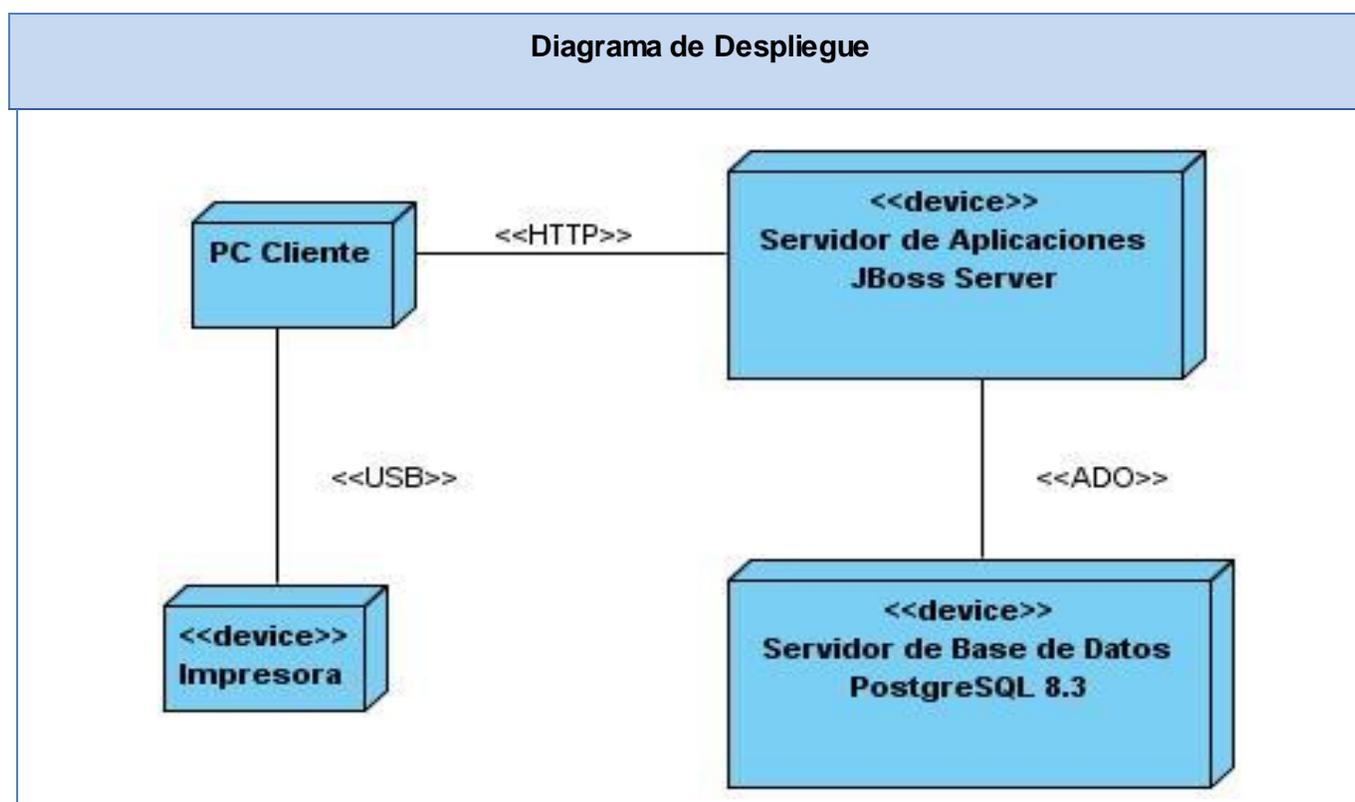






4.2.2 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución.



4.3 Estándares utilizados.

En el campo de la informática se define un estándar como: “Una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad. También significa un modelo o guía que se sigue para realizar un proceso o para no desviarse de un objetivo”. (24)

También puede decirse que estándar en informática, es un conjunto de especificaciones técnicas utilizadas para unificar el desarrollo de hardware o de software. Los estándares de computadoras se

desarrollan tradicionalmente de dos maneras. Una de ellas, la menos formal, tiene lugar cuando una compañía desarrolla un producto o una filosofía en solitario y consigue convertir el concepto en un estándar por la popularidad de la idea o por las imitaciones fabricadas por los competidores. Otra forma de crear un estándar es la redacción de las especificaciones por un grupo de expertos. Esta redacción se hace después de llevar a cabo un estudio exhaustivo de los métodos existentes, las propuestas y las tendencias o desarrollos tecnológicos.

4.3.1 Estándares de Diseño.

En el diseño de una interfaz para un sitio web es necesario tomar en cuenta la interacción de un usuario con un objetivo determinado y un espacio virtual, el sitio web, donde se encuentra el objeto buscado por el usuario. Deben evitarse problemas comunes como el exceso de entretenimiento e interactividad innecesarias, éstos influyen negativamente en el acceso del usuario a la información ofrecida y/o buscada dentro del sitio. Algunos efectos multimedia pueden ser muy atractivos, pero dentro de un sitio público traen consigo problemas de carga lenta del sitio web, poca compatibilidad en diferentes navegadores y versiones de los mismos.

El Módulo de Fisiatría 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 estos 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 am/pm, 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 test.

Capítulo 4: Implementación

- 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 este 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: “Si” y “No”. El botón “Si” 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 alinearé 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 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 píxel, la primera línea de componentes estará a 8 píxel del lado izquierdo del panel, entre la primera columna de componentes y la segunda 25 píxel)

The screenshot shows a web application window titled "Contenido" with a search bar "Buscar...". Below the title bar are two tabs: "Datos personales" and "Examen Físico". The "Examen Físico" tab is active, displaying a form titled "Examen Fisiátrico".

The form is divided into sections. The main section is "Examen Físico General - Desarrollo Sensorial", which contains the following fields:

- Facies:
- Marcha:
- Postura Decúbito:
- Postura Sedestación:
- Postura Bipedestación:
- Seguimiento Ocular:
- Reflejo Codeo-Palpebral:
- Tónico Asimétrico Del Cuello:
- Asimetría postural de los miembros:
- Actividad Motora Espontanea:
- Movimientos Anormales:

Below the main section are three sub-sections:

- Tono Pasivo
- Asimetrías - Tono Activo - Reflejos Primarios
- Observaciones

At the bottom of the form are two buttons: "Guardar" and "Salir".

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: <Seleccione>, 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.

4.3.2 Estándares de codificación

Actualmente se encuentran estándares de codificación para la mayoría de los lenguajes existentes. El uso de los mismos, partiendo de las convenciones definidas, permite una mejor comunicación entre los programadores, creando las condiciones para la reusabilidad y mantenimiento de los sistemas. Para definir el estilo de codificación a seguir en la aplicación se utilizó la notación estándar establecida para las aplicaciones desarrolladas en JAVA.

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.

4.4 Seguridad

La seguridad en un sistema, es de gran importancia, ya que de ella depende la integridad, autenticidad y confiabilidad de su información. 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.

Capítulo 4: Implementación

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. En este caso toda la autorización, desde la autorización a directorios, páginas, controles, opciones del menú, servicios del negocio, está basado en reglas, esto permite que ninguna de estas “reglas del negocio” esté hard-coded en la aplicación y que el cambio de alguna de estas reglas no requiera cambio alguno en el código, solo en la definición de alguna regla en un fichero de configuración. El Seam Security Framework permite todo esto gracias a su integración con el potente motor de reglas JBoss Rules. (25)

En este capítulo se presentaron los diagramas de Componentes que muestran la distribución de todos los paquetes para su implementación y el diagrama de Despliegue que describe la estructura de distribución de la aplicación. Además se refleja el uso de los estándares de diseño y codificación, así como de la seguridad del sistema.

CONCLUSIONES

El estudio de los sistemas que se encargan de evaluar a los niños con problemas en el neurodesarrollo más difundidos a nivel nacional e internacional, permitió determinar que estos no se ajustan al Sistema Nacional de Salud. Estos 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 constituye un subsistema del Sistema de Información Hospitalaria, en consecuencia la arquitectura del mismo está basada en la propia arquitectura de este sistema, utilizando las herramientas, tecnologías y metodología propuestas por este.

El Sistema de Evaluación del Neurodesarrollo en Niños constituye un subsistema del Sistema de Información Hospitalaria. Por tanto, se utiliza la arquitectura propuesta por el HIS, utilizando las herramientas, tecnologías y metodología.

Como resultado del estudio del funcionamiento del programa de atención y evaluación del niño en la especialidad de Fisiatría, se identificaron los principales procesos del negocio y se modelaron los flujos de trabajo propuestos por el Proceso Unificado de Desarrollo, obteniéndose los artefactos que se generan en cada uno de ellos.

Se realizó la implementación del Módulo de Fisiatría 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 pruebas del Peabody Motor Grueso y Fino y estas a su vez agilizan la evaluación de los niños en la consulta de fisiatría.

RECOMENDACIONES

Los autores del presente trabajo recomiendan:

- Incluir la prueba de la Escala Motora Modificada en el sistema.

REFERENCIAS BIBLIOGRÁFICAS

1. **Castro Ruz, Fidel.** La idea esencial es acercar los servicios primarios a los ciudadanos. *Periódico Granma*. 2003.
2. **Castro Ruz, Fidel.** Discurso pronunciado el 24 de Febrero 1960.
3. **Atención Temprana. Atención Temprana en la Provincia de Cienfuegos.** [En línea] [Citado el: 10 de noviembre del 2008.] www.sld.cu/sitios/rehabilitacion-temprana/temas.php
4. **Idem 3**
5. **Idem 3**
6. **Booch, G., Rumbaugh, J. y Jacobson, I.** *El Lenguaje Unificado de Modelado*. 2000.
7. **Jacobson, I, Booch, G y Rumbaugh, J.** “El Proceso Unificado de Desarrollo de Software”. La Habana : Félix Varela, 2004. Vol. 1.
8. **Arquitectura de programación en 3 capas.** <http://www.elcodigok.com.ar>. [En línea]. [Citado el: 25 de noviembre del 2008]. Disponible en: <http://www.elcodigok.com.ar/2007/09/arquitectura-de-programacion-en-3-capas/>
9. **Assembla. Documents. Documentación. Drools.** [En línea] [Citado el: 12 de abril del 2008.] http://www.assembla.com/spaces/FallasII/documents/c_YT_le88r3B0oab7jnrAJ/download/Drools%20Documentacion
10. **JBoss. JBoss. Dowlands. JBPM.** En línea] [Citado el: 12 de abril del 2008.] <http://www.jboss.org/jbossjbpm/downloads/>
11. **Conceptos Básicos de Computación.** [En línea] [Citado el: 20 de noviembre del 2008.] <http://haideeperez75.blogspot.com/feeds/posts/default/3466693163993704565>
12. **Owen Martin, Raj Jog.** BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. Popkin Software. 2003. p 4. Disponible en: www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf
13. **Jacobson, I. y Booch, G. y Rumbaugh, J.** El Proceso Unificado de Desarrollo de software. S.I.: Addison-Wesley, 2000.

Referencias Bibliográficas

14. **Pons Capote, Olga, y otros.** Introducción a las Bases de Datos. El Modelo Relacional. s.l. : Thomson Learning Ibero, 2005. ISBN 8497323963.
15. Peabody developmental motors scales (PDMS) an activity card. Hospital Pediátrico Universitario William Soler.
16. **Ingeniería de Software 1.** “UML y RUP.” UCI. curso 2008_2009. Clase Teórico Práctica # 1.
17. **Ingeniería de Software 1.** “UML y RUP.” UCI. curso 2008_2009. Clase Teórico Práctica # 1.
18. **Pascual y Genís.** Gestión y Reingeniería de Procesos.
19. **Kiran Garimella, Michael Lees y Bruce Williams** Introducción a BPM para Dummies.
20. **Geoffrey Sparks, Sparx Systems.** Una Introducción al UML. El Modelo de Casos de Uso. www.sparxsystems.com.ar - www.sparxsystems.cl
21. **Idem 19**
22. **Ingeniería de Software 2.** “Continuación del FT Análisis y Diseño. Modelo de Diseño.” UCI. curso 2008_2009. Conferencia # 1.
23. **Ingeniería de Software 2.** “Flujo de Implementación .” UCI. curso 2006_2007. Conferencia.
24. **Estándares de diseño web.** [En línea] [Citado el: 3 de abril del 2009.]
<http://www.slideshare.net/dwebslide/estandares-de-diseo-web>.
25. **IH-SW-DR-087 ALAS-HIS_ Elementos comunes_Modelo de casos de uso del sistema.**

BIBLIOGRAFÍA

- **Arquitectura de programación en 3 capas.** <http://www.elcodigok.com.ar>. [En línea]. [Citado el: 25 de noviembre del 2008]. 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 2008]. Disponible en: 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 2008] <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.
- **Conceptos Básicos de Computación.** [En línea]. [Citado el: 15 de enero del 2009.] <http://haideeperez75.blogspot.com/feeds/posts/default/3466693163993704565>
- **Casalánguida H., Durán J. AOWE.** Una Metodología Orientada a Aspectos para Desarrollo de Aplicaciones Web. Proceedings of ASSE 2007.
- **Desongles Corrales, Juan.** Técnicos Auxiliares de Informática. Temario parte II. s.l.: MAD-Eduforma, 2006. ISBN 8466554068.
- **Estándares de diseño web.** [Citado el: 20 de enero del 2009] <http://www.slideshare.net/dwebslide/estandares-de-diseo-web>.
- **GARRETT, J. J.** AJAX un nuevo acercamiento a Aplicaciones Web, [2007]. [Citado el: 10 de febrero del 2009] Disponible en: <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>

- **Geoffrey Sparks, Sparx Systems.** Una Introducción al UML. El Modelo de Casos de Uso. www.sparxsystems.com.ar - www.sparxsystems.cl
- **IH-SW-DR-087 ALAS-HIS_ Elementos comunes_ Modelo de casos de uso del sistema.**
- **Håkon Wium, Lie, Bert Bos. Cascading Style Sheets.** Designing for the Web, Third Edition, Addison Wesley Professional.
- **JavaHispano. [EN línea]** . [Citado el: 10 de febrero del 2009]
<http://www.javahispano.org/contenidos.type.action?type=NEWS&menuId=NEWS>
- **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>
- **Kiran Garimella, Michael Lees y Bruce Williams** Introducción a BPM para Dummies.
- **LARGMAN, C. UML Y PATRONES,** Introducción al análisis y diseño orientado a objetos.
- **María Consuelo Franky ,** Java EE 5 (sucesor de J2EE):el reto de volver a empezar. Junio de 2007
- **MARIN D. M. E.** Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. La Habana. 2006.
- **Owen Martin, Raj Jog.** BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. Popkin Software. 2003. p 4. [Citado el: 10 de marzo del 2009] Disponible en: www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf
- **Pons Capote, Olga, y otros.** Introducción a las Bases de Datos. El Modelo Relacional. s.l. : Thomson Learning Ibero, 2005. ISBN 8497323963.
- **Plataforma De Aplicaciones Jboss Enterprise.** [Citado el: 10 de marzo del 2009]
www.redhat.es/jboss
- **Peabody developmental motor scales an activity card.** Hospital Pediátrico Universitario William Soler.
- **Proyecto Esperanza.** http://www.cadenagramonte.cubaweb.cu/proyecto_esperanza

- **Potencier, Fabien y Zaninotto, François.** Symphony, la guía definitiva. [En línea] 2008. [Citado el: 10 de marzo del 2009] <http://www.librosweb.es/symfony/index.html>
- **PRESSMAN, R. S.** *Ingeniería del Software*. Un enfoque práctico. Ciudad de la Habana: 2005. Vol. 1
- **P. Stevens y R. Pooley.** Utilización de UML en ingeniería del software.
- **Revista Cubana de Informática Médica.** Revista Cubana de Informática Médica. [En línea] [Citado el: 15 de abril del 2009] http://www.cecam.sld.cu/pages/rcim/revista_15/articulos_htm
- **Software para el tratamiento terapéutico de niños con problemas cerebrales.** Grupo de investigación en Computación Gráfica y Procesamiento de Imágenes (OHWAHA).
- **Schach, Stephen R.** Análisis y diseño orientado a objetos con UML y el proceso unificado.
- **The PostgreSQL Global Development Group.** About [En línea] [Citado el: 15 de abril del 2009] <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>

ANEXO 1

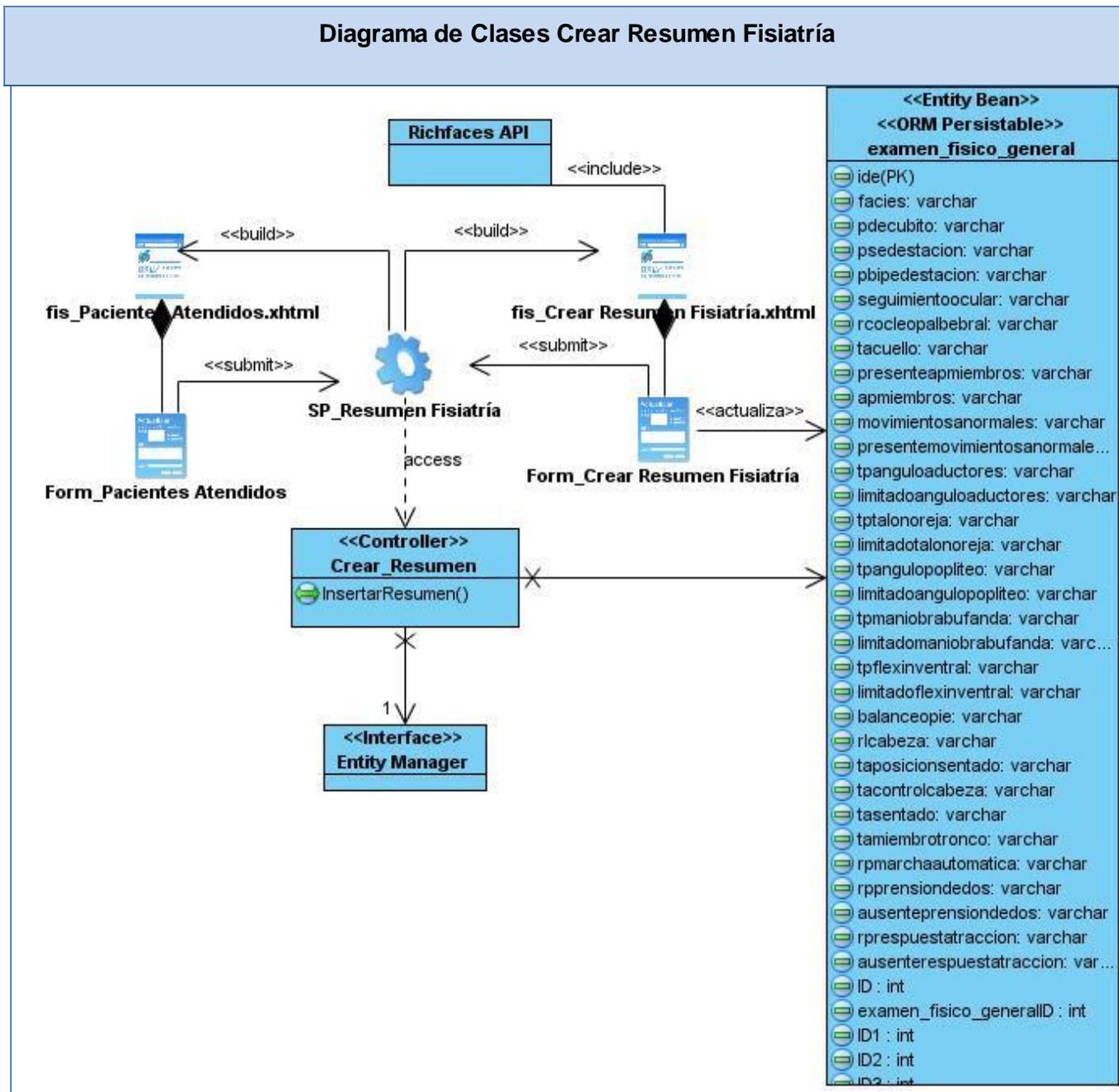


Diagrama de Clases Modificar Resumen Fisiatría

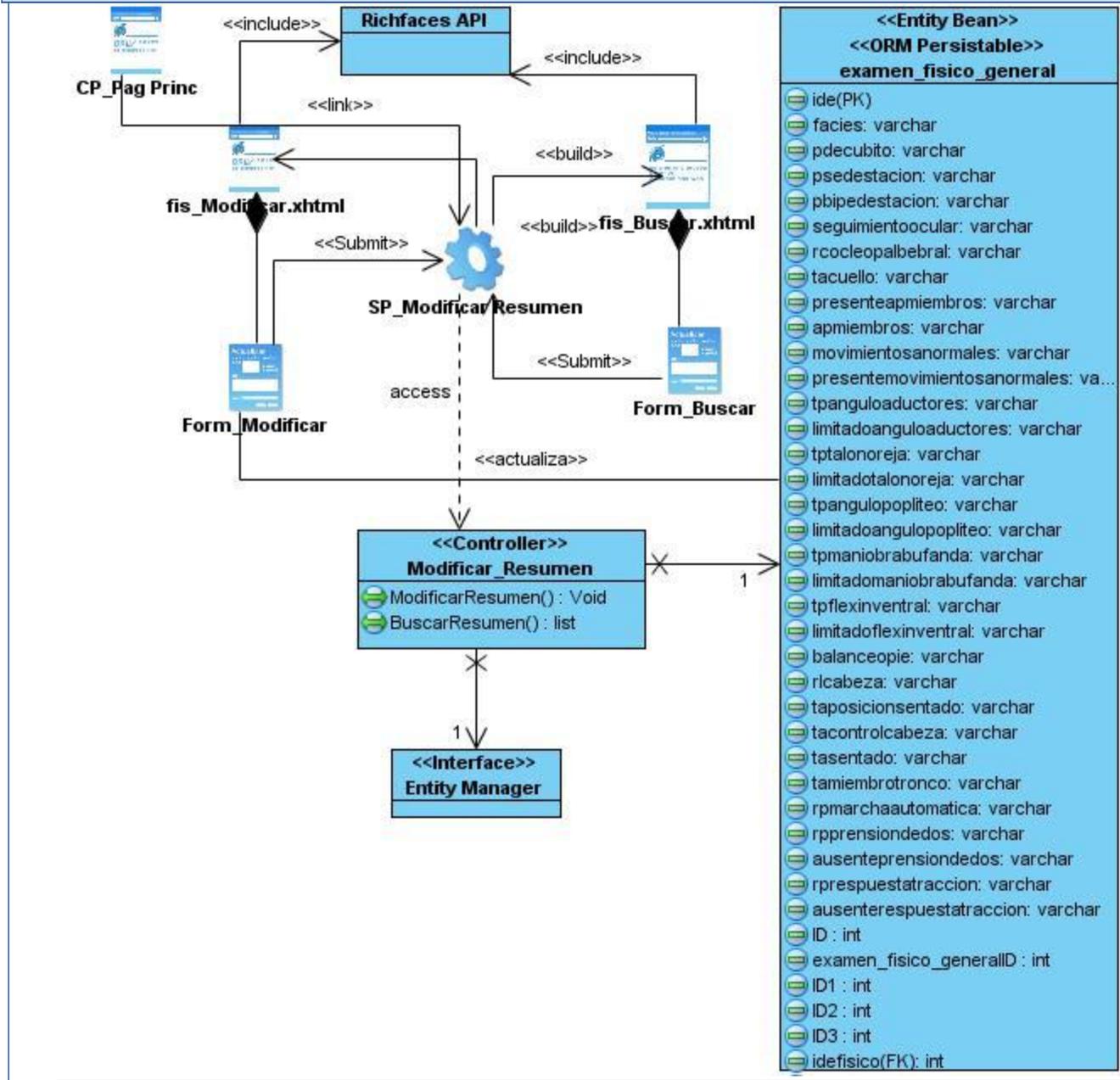


Diagrama de Clases Actualizar Sumario Fisiatría

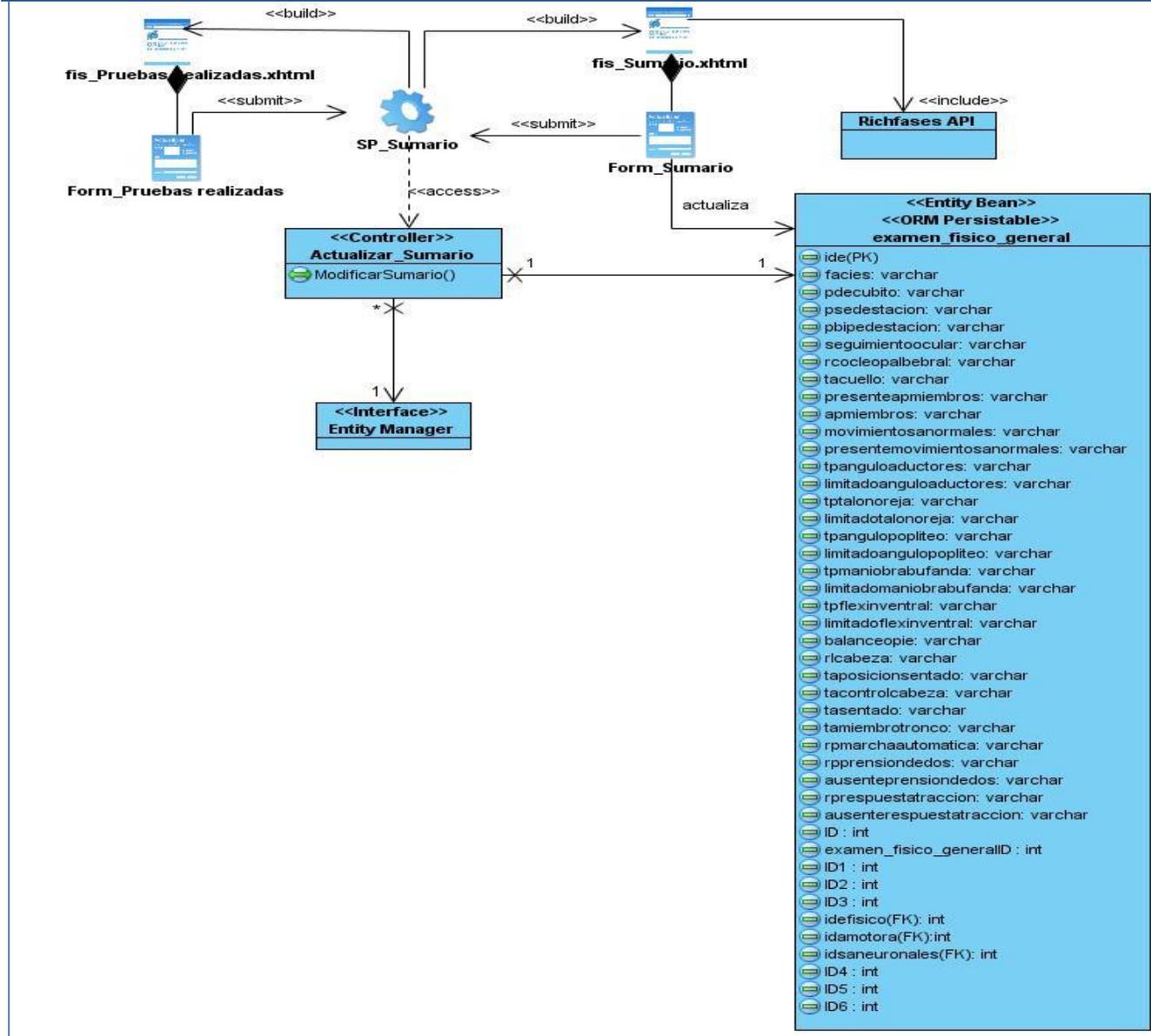


Diagrama de Clases Listar Pacientes Atendidos

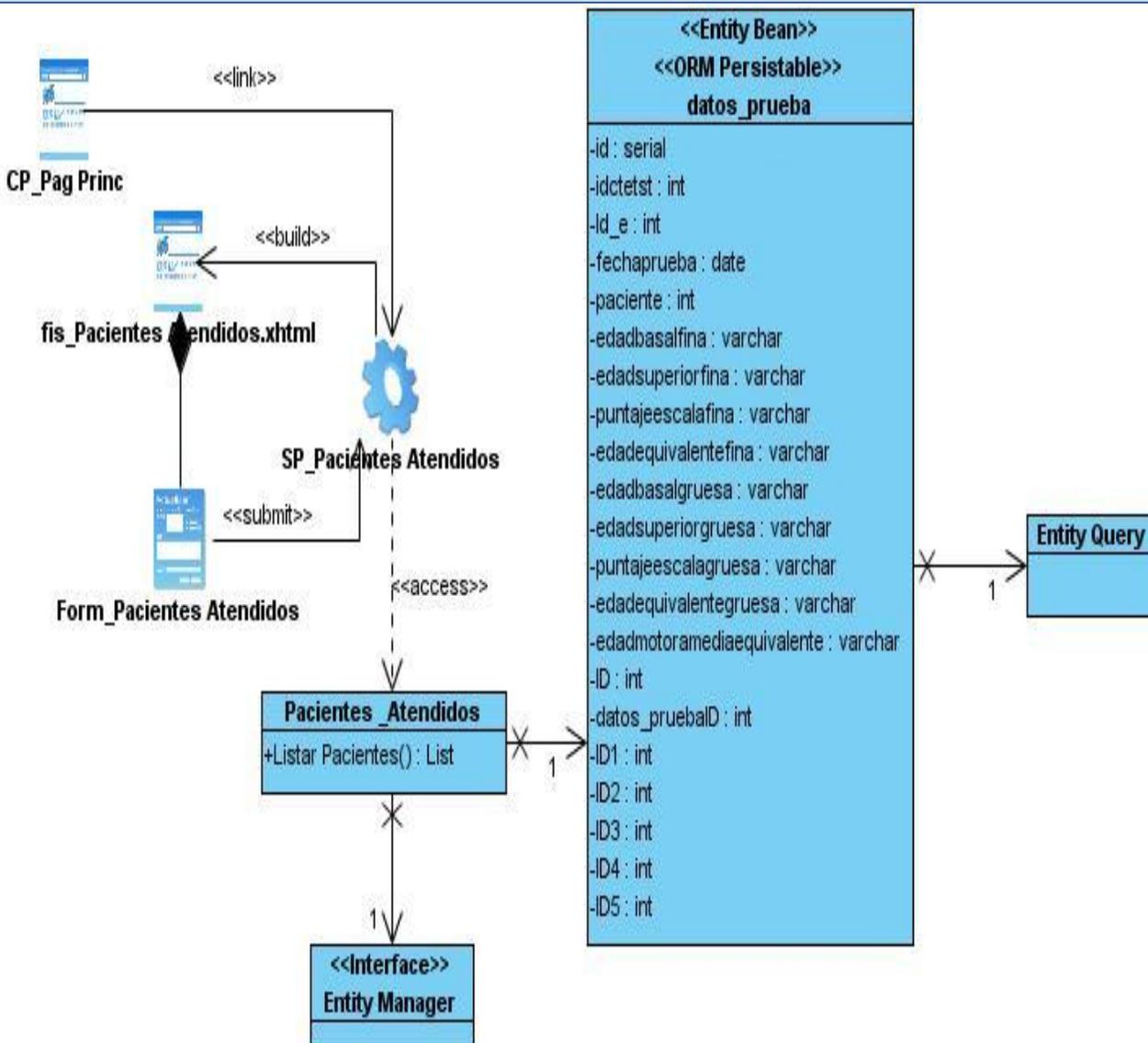
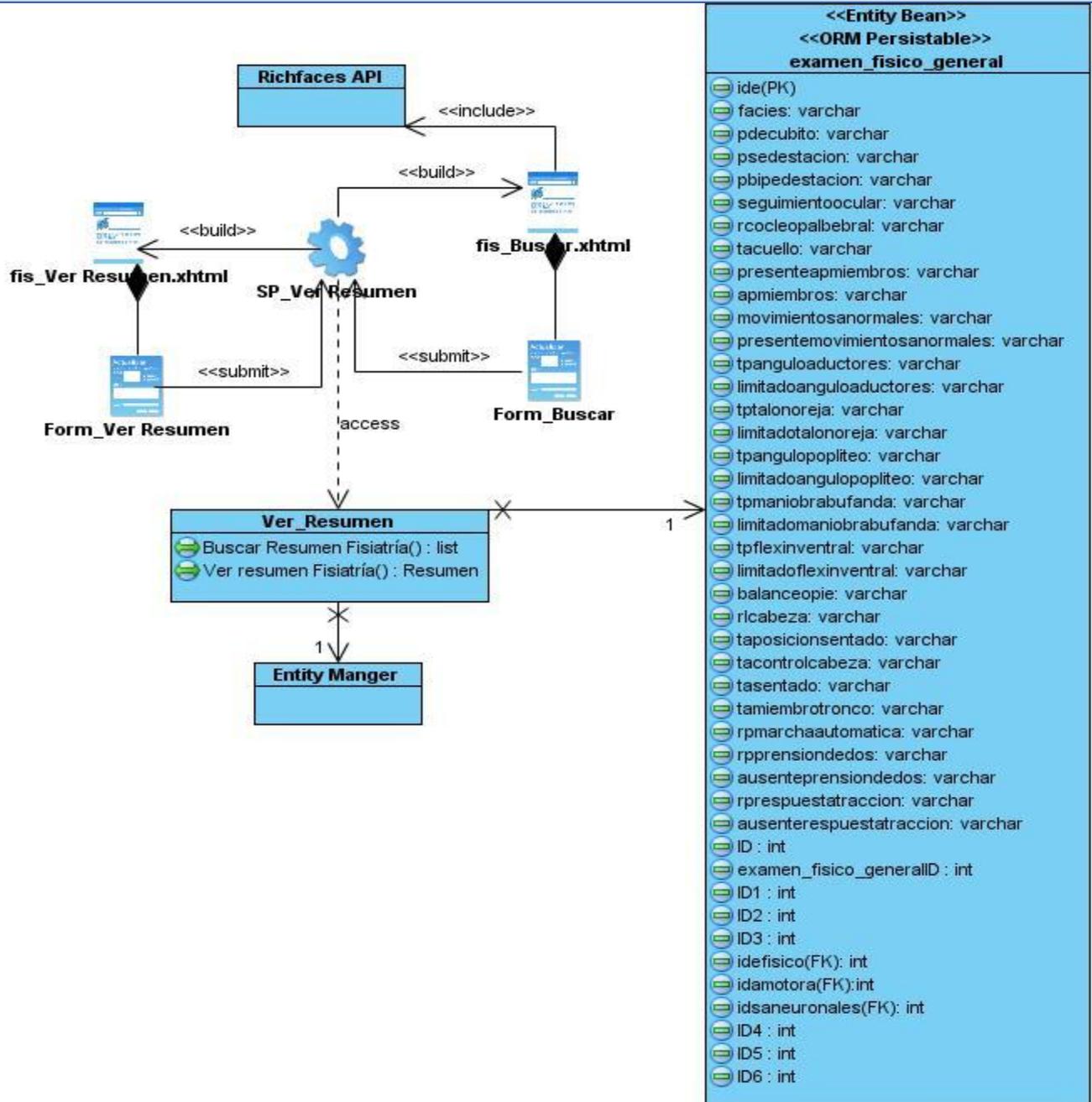


Diagrama de Clases Ver datos Resumen Fisiatría



ANEXO 2

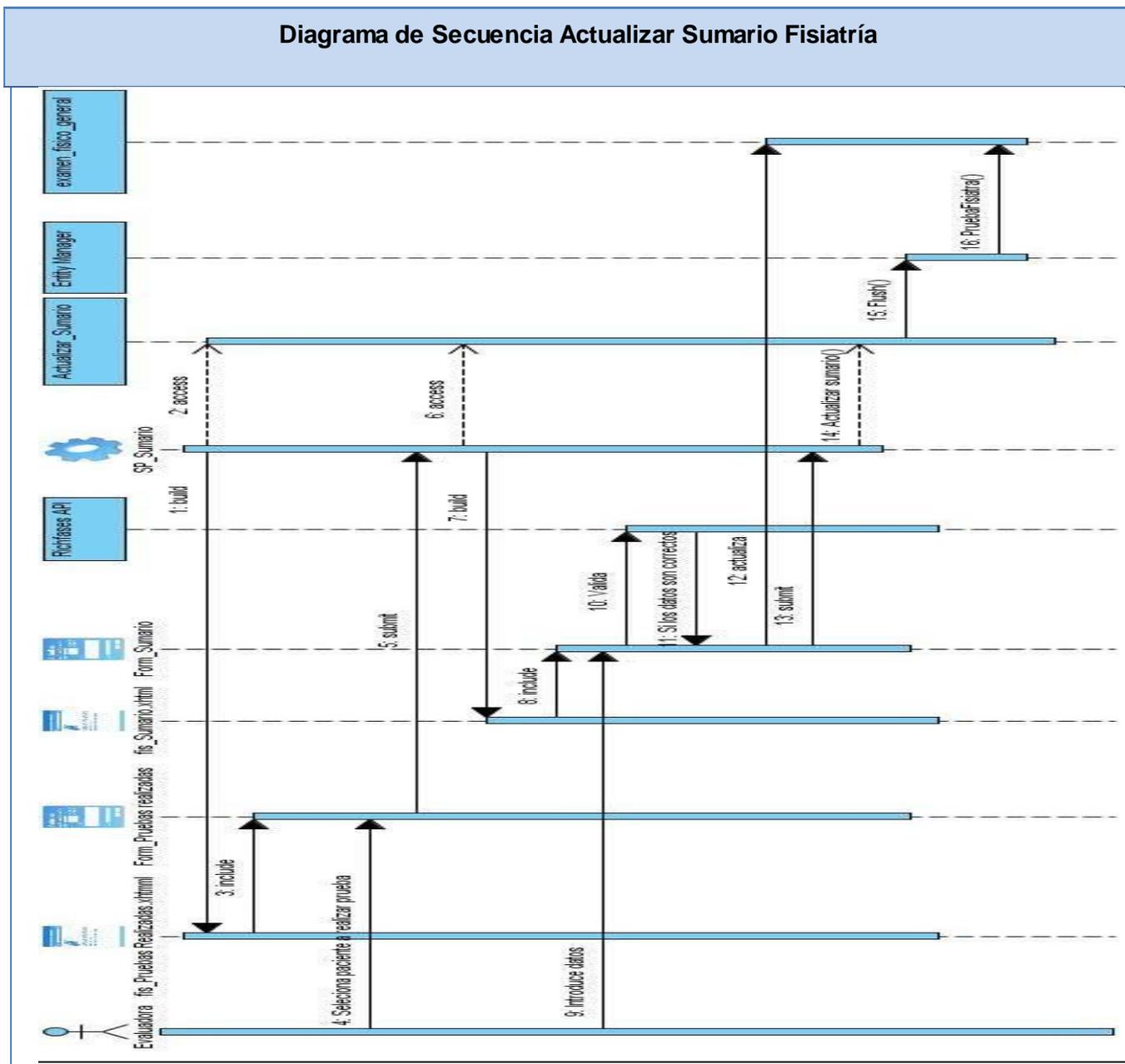


Diagrama de Secuencia Crear Resumen Fisiatría

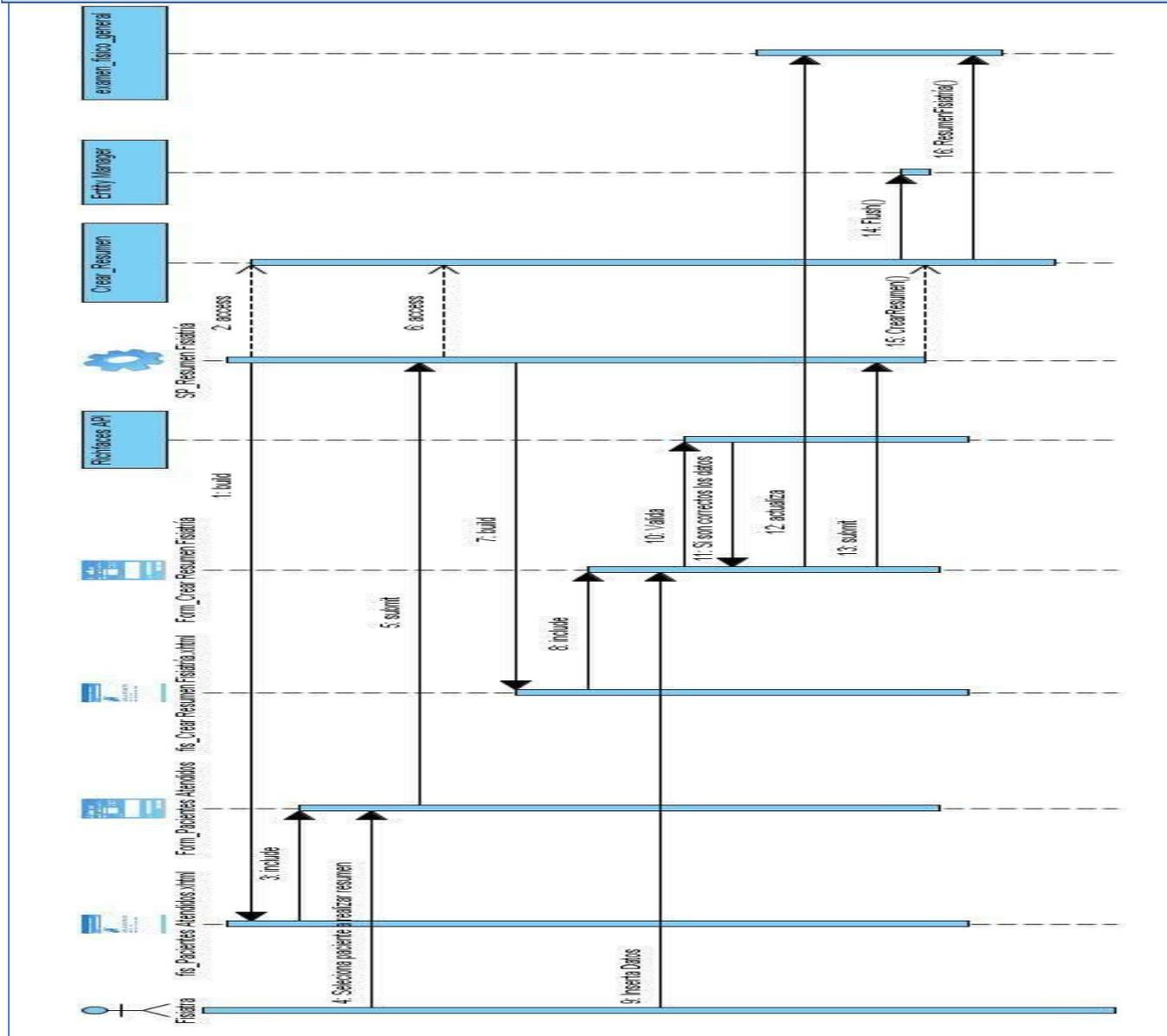


Diagrama de Secuencia Modificar Resumen Fisiatría

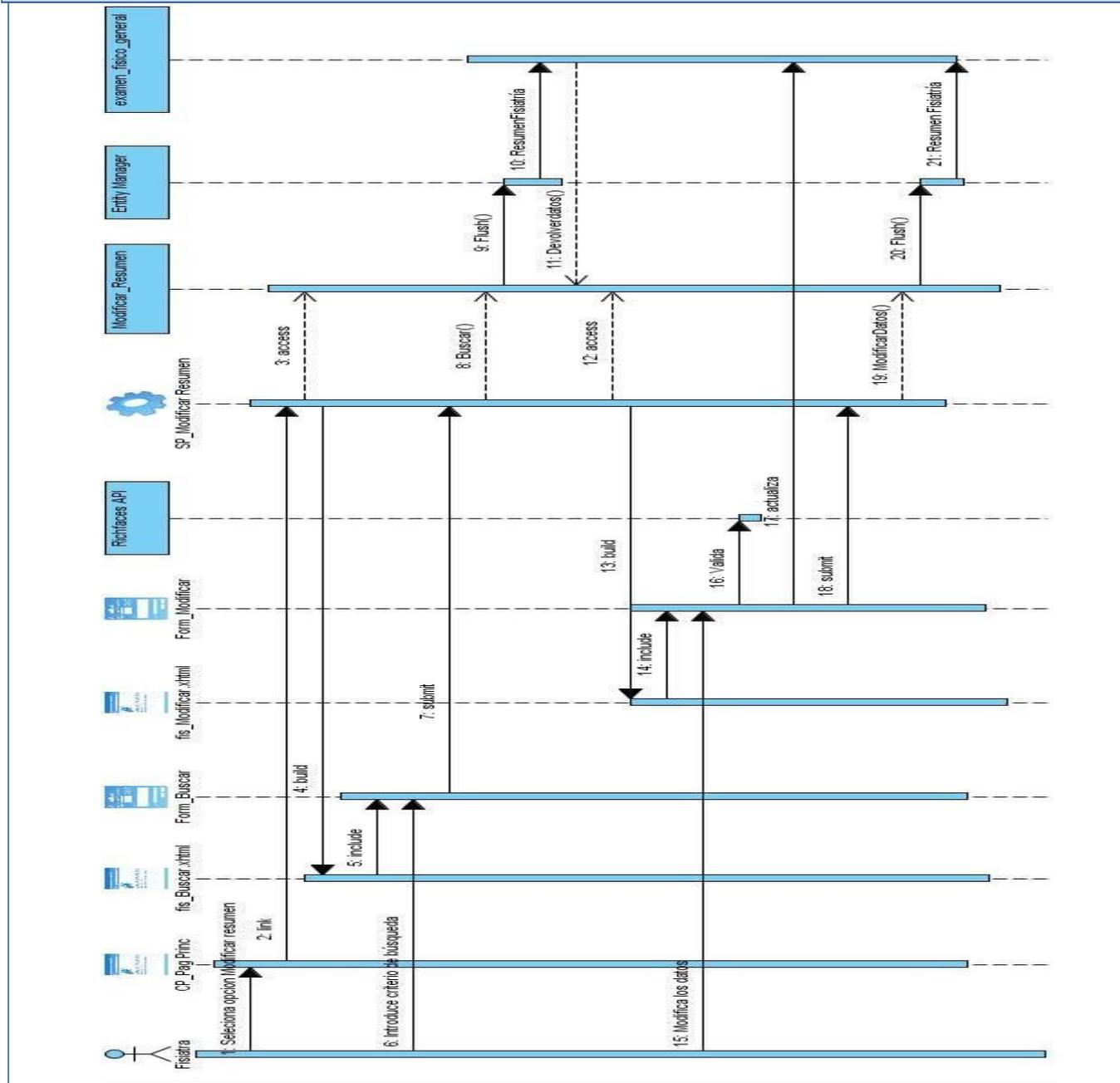


Diagrama de Secuencia Listar Pacientes Atendidos

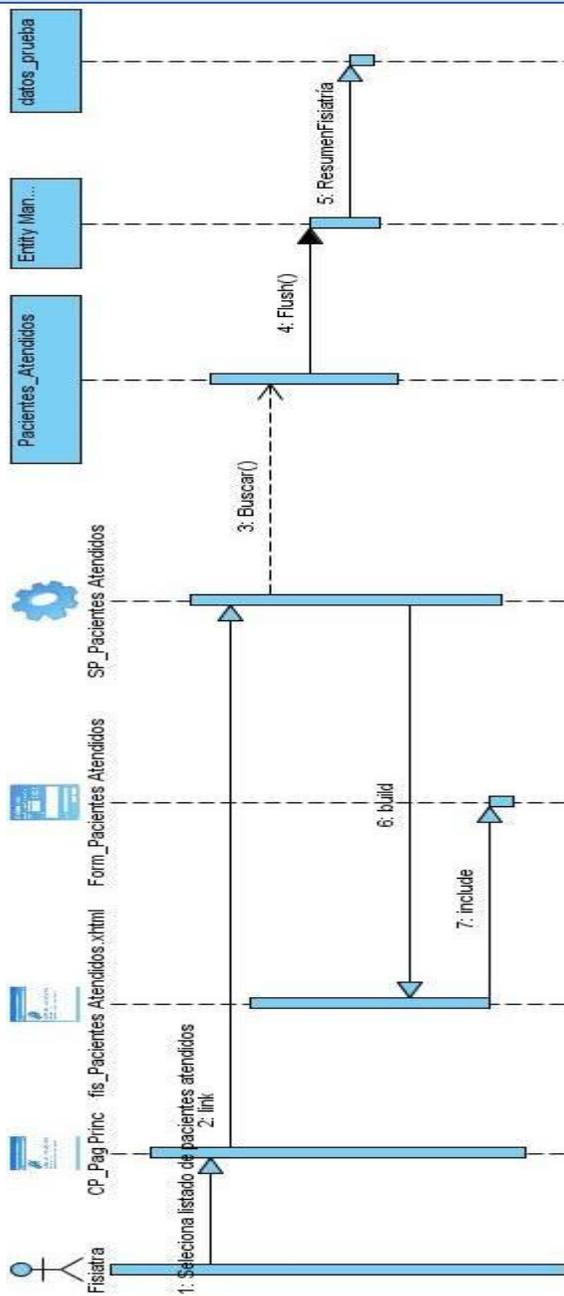
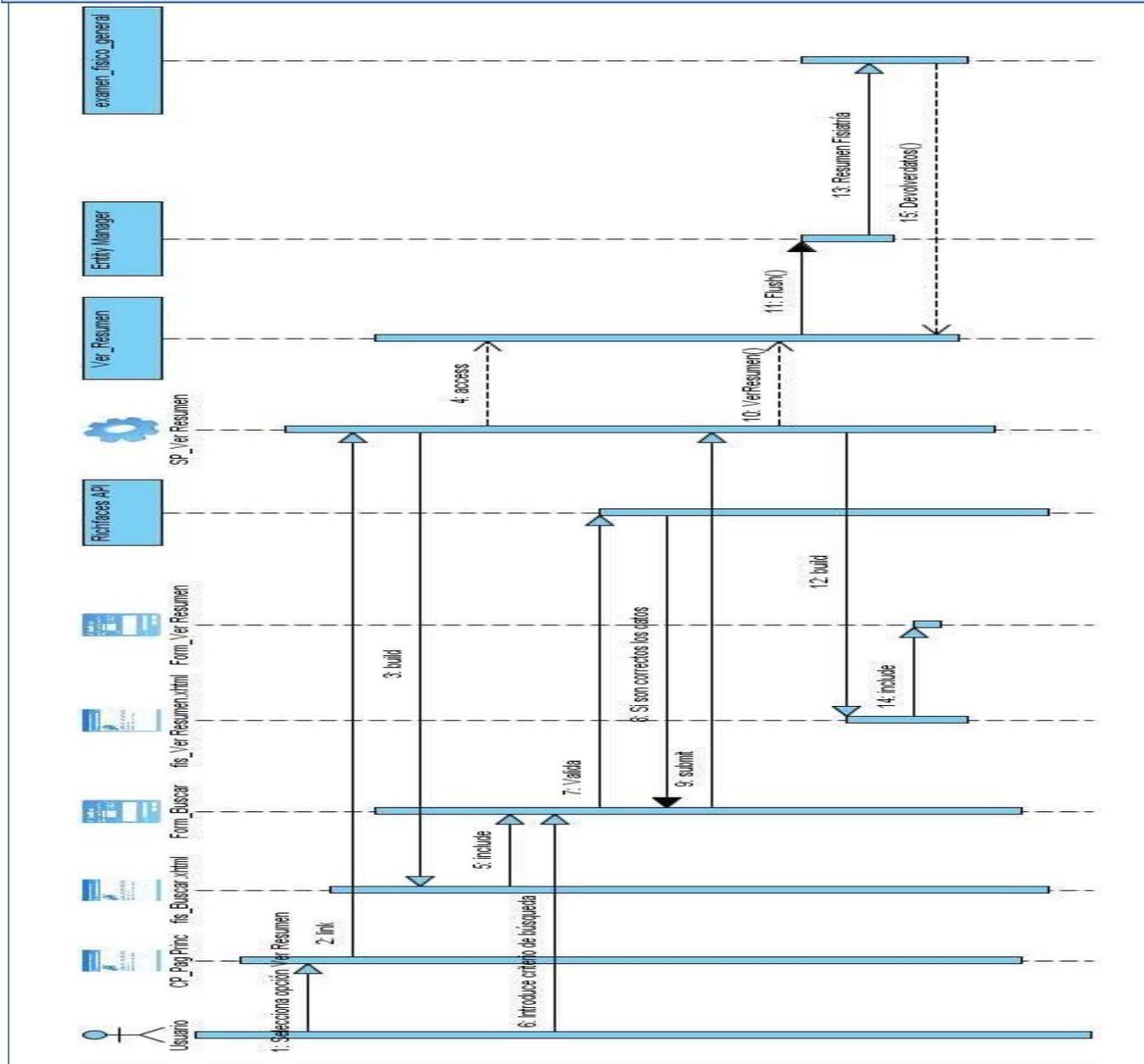


Diagrama de Secuencia Ver datos Resumen Fisiatría



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.

DMQ: Cociente de Desarrollo Motor. Es un test que se utiliza para determinar el coeficiente de desarrollo motor del niño.

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.

Fisiatría: Es la especialidad médica que se ocupa fundamentalmente de la rehabilitación de personas con patologías motoras. Para esto trabaja básicamente tres grandes áreas: La Medicina Física, la Medicina de Rehabilitación y los estudios electro-fisiológicos.

Fisiatra: El médico especializado en medicina física y rehabilitación recibe el nombre de fisiatra, es una especialidad médica que se ocupa de la restauración de las capacidades perdidas a causa de una enfermedad, trastorno o lesión.

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.

Neonatología: Especialidad que atiende al recién nacido/a en los primeras horas de vida extrauterina y se encarga de las enfermedades que pueden afectarle, incluida su vigilancia intensiva.

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 virtual, por encima de la base de datos relacional.

Glosario de Términos

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.

Subproceso: Son partes bien definidas en un proceso. Su identificación puede resultar útil para aislar los problemas que pueden presentarse y posibilitar diferentes tratamientos dentro de un mismo proceso.

Peabody Motor fino y grueso: Test estandarizado de administración individual que mide las habilidades motoras finas y gruesas de los niños de 0 a 5 años de edad.

RUP: (*Rational Unified Process*) Proceso unificado de desarrollo de software.

SENDN: Sistema de Evaluación del Neurodesarrollo en Niños.

SGBD: Sistema de Gestión de Bases de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una o varias base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

UML: (Lenguaje Unificado de Modelado) Es un lenguaje de modelado visual para especificar, visualizar, construir y documentar artefactos de un sistema de software.

UCIN: Unidades de Cuidados Intensivos Neonatales.

UCIP: Unidades de Cuidados Intensivos Pediátricos. Asistencia eficiente a las urgencias pediátricas, es el servicio del hospital dedicado a la asistencia intensiva integral y continuada al niño críticamente enfermo, independientemente de cual sea el origen de esta.

