



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.

FACULTAD 5.

Título: “Sistema Basado en el Conocimiento para asistir el diagnóstico y sugerir tratamientos en pacientes hemipléjicos con alteraciones motoras”

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

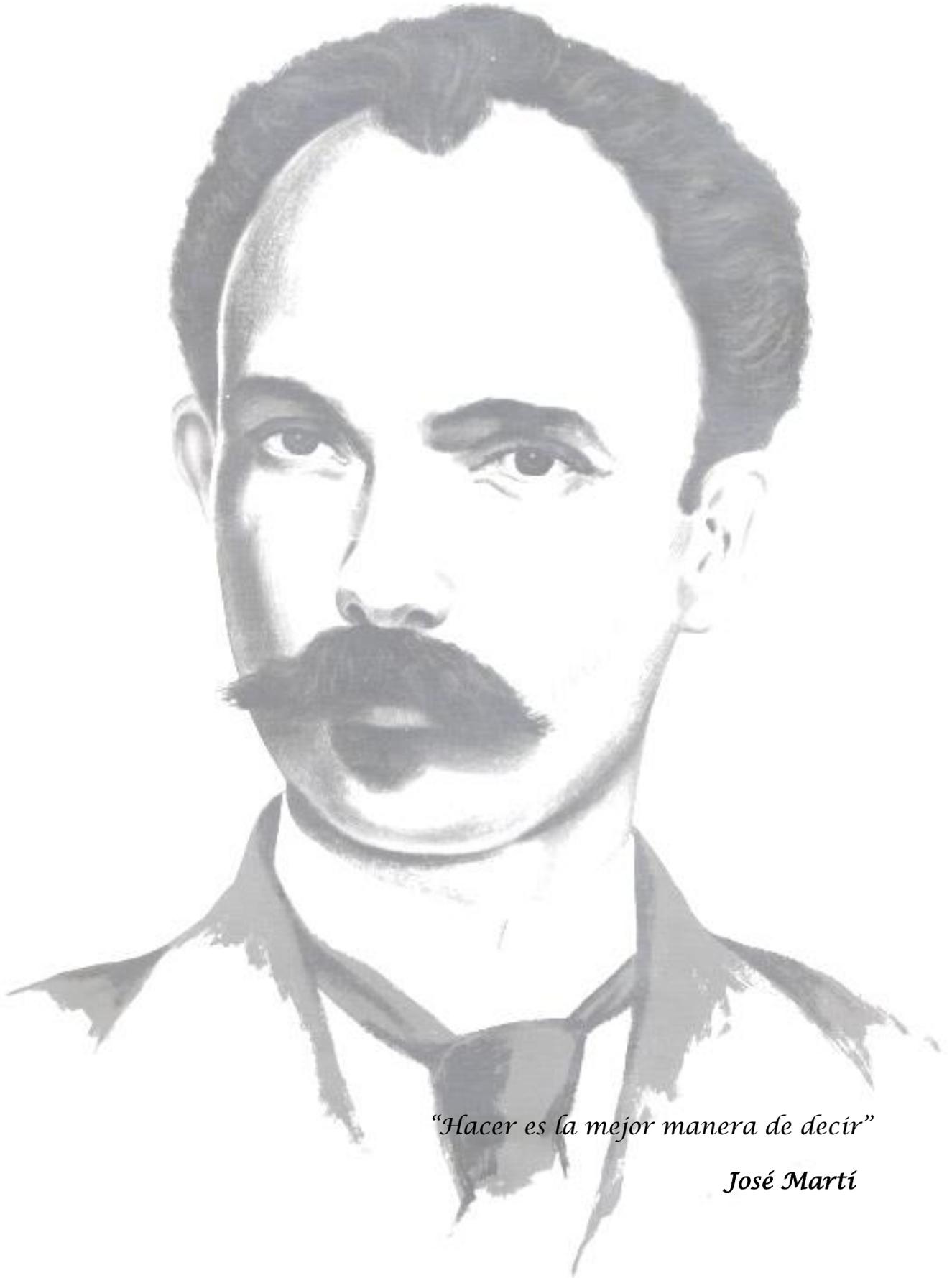
Autor: Eliuvis Matos Matos

Tutores: Ing. Andy Trujillo Rivero

Msc. Yunieski Coca Bergolla

Dr. Julio Zamarreño Hernández

**La Habana, Cuba 2016
“Año 58 de la Revolución”**



“Hacer es la mejor manera de decir”

José Martí

Declaración de autoría

Declaro ser el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Eliuvis Matos Matos

Autor

Ing. Andy Trujillo Rivero.

Tutor

Msc. Yunieski Coca Bergolla

Tutor

Dr. Julio Zamarreño Hernández

Tutor

Datos de contacto

Tutor: Ing. Andy Trujillo Rivero

Email: arivero@uci.cu

Graduado de Ingeniería en Ciencias Informáticas (UCI) en 2008 por la Universidad de las Ciencias Informáticas (UCI). Ha trabajado en varios proyectos e investigaciones relacionadas con el área de la Inteligencia Artificial (IA) en los centros de Informática Industrial (CEDIN) y de Entornos Interactivos 3D (Vertex) de la Facultad 5 de la UCI. Actualmente dirige el trabajo de la línea de IA en este último.

Tutor: MSc. Yuniesky Coca Bergolla.

Email: ycoca@uci.cu

Graduado de Licenciatura en Ciencias de la Computación en la Universidad Central de Las Villas (UCLV) 2003. Defendió su maestría en Informática Aplicada en la Universidad de las Ciencias Informáticas (UCI) en 2007. Ha tutorado más de 15 tesis de grado relacionadas con temas de Inteligencia Artificial (IA). Actualmente se desempeña como jefe de departamento de técnicas de programación de la facultad 5 y Jefe de disciplina de IA en la UCI. Además, es árbitro de la Revista Cubana de las Ciencias Informáticas (RCCI).

Tutor: Dr. Julio Zamarreño Hernández

Email: jzama@infomed.sld.cu

Especialista en Medicina Física y Rehabilitación. Profesor Asistente de la Facultad de Medicina "Dr. Enrique Cabrera" de la Universidad de La Habana. Máster en Ciencias en Procederes Fisioterapéuticos en Rehabilitación Neurológica.

Jefe de la Sección de Estimulación Eléctrica Funcional del Centro Nacional de Rehabilitación Julio Díaz. Responsable del Área de Eventos, Ciencia y Técnica del Centro Hospitalario. Presidente del Comité Organizador de "NeuroRehabana", evento científico internacional bianual sobre Neuro-rehabilitación.

Dedicatoria

Dedico este trabajo muy especialmente a mi querida mamá, a mi papá, mis hermanos y mis abuelitas; a toda mi familia que tanto me apoyó, y a los amigos que me acompañaron a lo largo de este camino, sientan este resultado como suyo.

Agradecimientos

¡A mi mamá!!! la mujer más maravillosa que existe, y mi mayor inspiración para llegar hasta aquí. Gracias por siempre estar ahí, por no rendirte nunca. ¡Te amo mamá!!

A mi papá, por la confianza depositada en mí, gracias por todos tus consejos y por cómo me apoyaste siempre. ¡Felicidades viejo!!!, tienes un hijo ingeniero.

A mi hermano mayor Erloidis por ser mi mejor amigo desde siempre e inspirarme a tomar este camino y llevarlo hasta el final, por ser como un segundo padre. Gracias papo.

A mi hermana Aidalis y mi hermanito Frank, gracias por su inmenso cariño. Espero ser un buen ejemplo para ustedes.

A mis dos bellas abuelas, por su cariño, por siempre tenerme presente en sus oraciones y por preocuparse por mí como si aún fuera un niño. Las quiero mucho. Al resto de mi familia, mis tíos y tías, a todos mis primos.

A mis tutores, al Profe Andy Trujillo, gracias por venir donde mí con este tema de tesis, por tu ayuda y la confianza depositada, al Profe Yunieski Coca, por no dudar cuando te pedimos que nos apoyaras con este trabajo, por tus correcciones siempre oportunas y por cómo me hiciste estudiar y superarme, al Dr Julio Zamarreño, por el empeño dedicado, por los largos días que aportó para que el trabajo pudiera salir adelante, por plantearme tantos retos. Gracias a los tres.

A mis amigos, a Odette y Lilitiana, quienes me hicieron prometerles que su nombre sería el primero que diría de entre todas mis amistades si lográbamos llegar a este día, ¡lo logramos Ode y Lili!! y cumplí mi promesa.

Al Feló con quien siempre compartía un saludo de choque de puños antes de empezar cualquier examen, ya nos dolían los puños, pero lo logramos brother. A Leduan a quien cariñosamente renombre “el niño” porque su nombre no lograba aprendérmelo y al final resultó ser como un hermano para mí.

A Keimis, Brenda, Juan Grabiel y el Lachy, por ser parte también de esa familia que tengo aquí. A mis compañeros de grupo, desde aquel 5101, hasta el 5503. A mis compañeros de Apartamento, los más escandalosos de toda la UCI, gracias por los buenos ratos compartidos.

A todos mis maestros y profesores a lo largo de mi vida como estudiante, de ellos aprendí mucho, muy especialmente a mi maestra Nieve, quien me enseñó a leer y escribir y a partir de ahí surgió todo. A mis amigos, los que ya hoy no estamos juntos, a todos muchas gracias.

Resumen

El diagnóstico y tratamiento de pacientes hemipléjicos con alteraciones motoras, se encuentra entre los servicios más demandados en los centros de rehabilitación. Para realizar esta tarea, se requiere la presencia de especialistas frente al paciente, pues son los que portan el conocimiento necesario para desempeñar la labor de consulta. No obstante, estos centros no siempre cuentan con la cantidad de personal suficiente, lo que incrementa la carga de trabajo para los que prestan el servicio, y propicia que no se documente correctamente la información del paciente. La eficiencia del diagnóstico que se genere y del tratamiento que se asigne, está en dependencia del grado de experticia con que cuente el especialista y de su capacidad para relacionar correctamente los síntomas. Además, durante el diagnóstico, existe la posibilidad de que se omitan elementos potencialmente relevantes. Por estos motivos se desarrolló SBC_Hemiplejia, un Sistema Basado en el Conocimiento para asistir el diagnóstico de los pacientes, permitiendo gestionar su historial clínico y sugerir tratamientos para los mismos. Este utiliza un módulo inteligente implementado como un Sistema Basado en Casos y una base de conocimientos de pacientes previamente tratados para confeccionar una propuesta de tratamiento. Además, un módulo de explicación detalla el proceso seguido por el sistema para confeccionar dicho tratamiento. El desarrollo fue guiado por la metodología XP, se utilizó lenguaje de programación Java mediante la plataforma JavaFX, el framework Hibernate y las herramientas: Netbeans IDE 8.1, PostgreSQL 9.2 y Scene Builder 2.0.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.2 Hemiplejía	6
1.3 Inteligencia Artificial y Sistemas Basados en el Conocimiento	8
1.3.1 Tipos de SBC.....	11
1.3.2 Sistemas Expertos	17
1.4 Sistemas Basados en el Conocimiento usados en el diagnóstico médico	20
1.5 Herramientas y Tecnologías	22
1.5.1 Desarrollo mediante XP:	25
Capítulo 2: Propuesta de Solución	30
2.1 Sistema Propuesto	30
2.2 Adquisición del conocimiento	32
2.2.1 Técnicas de Adquisición del Conocimiento	33
2.3 Base de Casos	33
2.4 Proceso de extracción del conocimiento	38
2.4.1 Etapa de Recuperación	38
2.4.2 Etapa de Reutilización.....	40
2.4.3 Etapa de Adaptación	40
2.4.4 Almacenamiento o Aprendizaje.....	41
2.4.5 Módulo de Explicación	41
2.5 Control, acceso y gestión de la información	41
2.6 Desarrollo guiado por XP.....	42
2.6.1 Fase de Exploración	42
2.6.2 Fase de Planificación.....	46
2.6.3 Fase de Iteraciones	50
2.7 Características y Diseño del Sistema	50
2.7.1 Arquitectura del Sistema	50
2.7.2 Tarjetas Clase-Responsabilidad-Colaboración (CRC)	51
2.7.3 Patrones de diseño	52

Capítulo 3: Implementación y pruebas del sistema	55
3.1 Implementación	55
3.1.1 Estructura y descripción de la aplicación.....	55
3.2 Pruebas.....	57
3.2.1 Pruebas Unitarias.....	57
3.2.2 Pruebas Aceptación	59
3.2.3 Pruebas de Rendimiento.....	62
3.3 Validación	64
Conclusiones	66
Recomendaciones.....	67
Referencias Bibliográficas	68

Introducción

Introducción

La gestión y organización de los grandes volúmenes de datos que se generan en la sociedad actual, ha sido uno de los aspectos en los cuales las Tecnologías de la Informática y las Comunicaciones (TIC) han desempeñado un papel protagónico. Para facilitar esta labor, se han creado un conjunto de herramientas informáticas que facilitan el manejo de estos datos. Muchas de estas incorporan técnicas de Inteligencia Artificial (IA), una rama de la informática que se especializa en el estudio y creación de métodos para dotar a los ordenadores de habilidades que son propias de los seres humanos. Estas aplicaciones no solo sirven para organizar y gestionar, sino que son capaces de extraer información o arribar a conclusiones sobre ese dominio de datos. A este tipo de sistemas se le conoce como Sistemas Basados en el Conocimiento (SBC). Esta técnica “representa un paso delante de los sistemas de información convencionales, al pretender representar funciones cognitivas del ser humano como el aprendizaje y el razonamiento” (1). Surgen a mediados de los años 60 con el desarrollo del sistema DENDRAL.

Dependiendo del tipo de conocimiento con que se desee operar existen diferentes técnicas de SBC, entre ellos: los Sistemas Basados en Reglas (SBR), los Sistemas Basados en Probabilidades (SBP), Sistemas Expertos Conexionistas o Redes Expertas (RE) y los Sistemas Basados en Casos (SBCasos). Entre los primeros sistemas desarrollados se encuentra MYCIN un SBC para determinar enfermedades infecciosas de la sangre. A partir de MYCIN, el desarrollo de este tipo de aplicaciones se generalizó en todo el mundo, siendo actualmente uno de los tipos de sistemas más aplicados a la gestión de procesos en disímiles áreas como la industria, las finanzas, la electrónica, la agricultura, en el área militar, la medicina entre otras. Actualmente se emplean en lugares donde existe escasez de personal calificado o es muy difícil el acceso de especialistas humanos, como misiones espaciales o la exploración de volcanes activos. También se usan en áreas donde se desee sacar el máximo provecho a los conocimientos existentes, ya sea para incrementar ventas o mejorar determinados procesos. Existe una gran variedad de este tipo de sistemas, dependiendo de la finalidad y el uso que se les dé, pueden ser: sistemas de diagnóstico, diseño, planificación, instrucción, predicción, enseñanza y muchos otros.

Introducción

En Cuba, como parte del proceso de informatización de la sociedad ha habido un incremento en el desarrollo de aplicaciones informáticas. Este desarrollo lo encabezan las casas de altos estudios como la Universidad Central de Las Villas, el Instituto Politécnico Superior José Antonio Echevarría (ISPJAE) y la Universidad de las Ciencias Informáticas (UCI), además de los centros de desarrollo de software existentes en el país. Estos llevan a cabo investigaciones y desarrollan productos de las más disímiles esferas, la IA, por el alcance y el desarrollo que ha logrado en todo el mundo, se ha convertido en centro de atención para los investigadores y desarrolladores de software. El estudio de esta rama ya ha dado sus frutos y el desarrollo de SBC no ha quedado exento a tal proceso de investigación. Los beneficios de la aplicación de estos sistemas han llegado a varias esferas como son: los procesos de penitenciaría, la mejora de los procesos de desarrollo de software, la industria petrolera, la minería de datos en determinadas empresas y la medicina. Particularmente la medicina es una de las más beneficiadas pues ya son varios los sistemas implementados para mejorar los procesos de determinadas áreas entre las que se encuentran: las enfermedades genéticas, interacciones medicamentosas, diagnóstico y tratamiento de determinadas enfermedades, y rehabilitación de pacientes con afecciones físico-motoras. Precisamente sobre esta última, existe un intercambio de trabajo entre el centro Entornos Interactivos 3D (Vertex) de la Facultad 5 en la UCI, y el Centro Nacional de Rehabilitación Julio Díaz, para lograr un incremento en la calidad y agilidad de los procesos que se llevan a cabo en dicho hospital mediante la informatización. Una de las áreas donde se desea trabajar es el Departamento de Estimulación Eléctrica Funcional, precisamente el proceso de diagnóstico y tratamiento a pacientes hemipléjicos que sufren alteraciones motoras producto de un accidente cerebrovascular (ACV).

En dicho departamento existe un alto nivel de profesionalidad y el personal se esfuerza al máximo para que los pacientes se rehabiliten de la mejor manera en el menor tiempo posible. No obstante, el proceso presenta las siguientes **limitaciones**: El número de especialistas capacitados para diagnosticar y confeccionar las terapias de los pacientes es limitado y no logran cubrir toda la demanda de este servicio, además la consulta requiere de la presencia obligatoria del médico que es quien porta el conocimiento

Introducción

necesario para dar el tratamiento. Al existir déficit de personal calificado el mayor esfuerzo es dirigido a la atención inmediata de los pacientes y no se documenta correctamente el seguimiento de estos, dificultando el estudio de la evolución del mismo en un futuro. Existe la posibilidad de que el médico omita aspectos relevantes dentro del proceso de diagnóstico o que simplemente no logre asociar correctamente los síntomas con el tratamiento indicado, esto depende de los conocimientos individuales de cada especialista y puede provocar una deficiente o tardía recuperación del paciente o un incremento de la lesión.

Tales insuficiencias justifican la necesidad de plantear el siguiente **problema científico**: ¿Cómo asistir a los médicos en el diagnóstico y tratamiento de alteraciones motoras en pacientes con hemiplejía?

Para dar solución al problema científico planteado anteriormente se define como **objeto de estudio**: Los Sistemas Basados en el Conocimiento y como **campo de acción**: Sistemas Basados en Casos para el diagnóstico y tratamiento de pacientes hemipléjicos con alteraciones motoras.

En relación con el problema científico, se formuló el siguiente **objetivo de investigación**: Desarrollar un Sistema Basado en el Conocimiento de apoyo a la toma de decisiones en el diagnóstico y tratamiento de pacientes hemipléjicos con alteraciones motoras.

Para dar solución al objetivo expuesto se desarrollaron las siguientes **tareas de investigación**:

1. Elaboración del marco teórico de la investigación; para identificar métodos y herramientas utilizados en el desarrollo de Sistemas Basados en el Conocimiento de apoyo a procesos de diagnóstico y tratamiento médico.
2. Caracterización de los Sistemas Basados en el Conocimiento y los Sistemas Expertos; para identificar aspectos relevantes que puedan ser utilizados en la solución del problema.

Introducción

3. Caracterización del dominio del problema (conceptos, relaciones, restricciones, reglas, flujo de información) a través del estudio de la bibliografía, así como de entrevistas y encuestas a especialistas del Centro Nacional de Rehabilitación Julio Díaz; para comprender el funcionamiento de los procesos involucrados.
4. Selección y configuración de las herramientas para el desarrollo del Sistema Basado en el Conocimiento.
5. Diseño, implementación, integración y pruebas del sistema para registrar los datos de los pacientes durante las consultas; para elaborar la base de conocimientos.
6. Análisis, depuración y validación los datos obtenidos, atendiendo a su consistencia, completitud, suficiencia, redundancia, validez, veracidad y generalidad; para garantizar su calidad y la del sistema.
7. Diseño, implementación, integración y pruebas de un Sistema Basado en el Conocimiento; para asistir a los médicos en la orientación de tratamientos a los pacientes, basándose en la experiencia adquirida de casos previos.
8. Comprobación del desempeño y depuración del Sistema Basado en el Conocimiento; para verificar su efectividad, validez y aceptación.

Para el desarrollo de estas tareas, se hizo uso de los siguientes **métodos de investigación**:

Métodos de Nivel Teórico:

Modelación: Para realizar una representación del proceso estudiado que sirva de guía en el desarrollo del sistema, y mediante este, identificar las características y relaciones fundamentales. También se usó en el proceso de construcción de la base de conocimiento y del módulo inteligente que se integra al sistema.

Introducción

Histórico-Lógico: Se emplea para identificar las tendencias actuales de los Sistemas Basados en el Conocimiento y la evolución de estos, además en la recopilación de información sobre soluciones similares al tema en cuestión.

Analítico-Sintético: Para el análisis, evaluación y selección de las técnicas a emplear en el desarrollo del SBC. Se empleó en sintetizar la información que se obtuvo mediante el intercambio con los expertos para que pudiera ser usada en la construcción del sistema, además, en la identificación de los elementos del marco teórico de la investigación.

Inductivo-Deductivo: Se utilizó en la definición de conceptos y criterios a partir de los conocimientos generales tratados en la investigación y para arribar a conclusiones sobre las respuestas del sistema en la etapa de prueba.

Métodos de Nivel Empírico:

Entrevista: Para la adquisición de los datos utilizados para crear la base de conocimiento del Sistema Basado en el Conocimiento y la comprensión del proceso de diagnóstico y tratamiento a pacientes con hemiplejía.

Matemáticos-Estadísticos: Para cuantificar los resultados de las pruebas realizadas al sistema. En la etapa de planificación se utilizó en el cálculo de las unidades de tiempo necesarias para el desarrollo del sistema.

Capítulo 1: Fundamentación Teórica

Capítulo 1: Fundamentación Teórica

En el presente capítulo se exponen los conceptos y definiciones fundamentales relacionados con la hemiplejía, causas que la provocan y su tratamiento médico. Se caracterizan además los Sistemas Basados en el Conocimiento y los Sistemas Expertos como parte de estos, sus ventajas, desventajas y áreas donde se aplican, así como la actualidad de este tipo de sistemas dentro del área de las ciencias médicas.

1.2 Hemiplejía

La Hemiplejía “es un síndrome neurológico donde se produce una lesión del tejido nervioso que se caracteriza por la pérdida de la movilidad voluntaria en una mitad vertical del cuerpo, debido a una alteración funcional u orgánica en la vía piramidal; los síntomas invalidantes más frecuentes son: parálisis, trastornos de la sensibilidad, desfase motor, alteración del esquema corporal, trastornos de la percepción de la vertical, trastornos posturales, del lenguaje, psicológicos y de las funciones mentales” (2). Puede ser un trastorno presente en el nacimiento, o puede adquirirse después. La causa subyacente de la hemiplejía es el daño al cerebro como resultado del flujo sanguíneo alterado. Esta interrupción puede estar causada por varios factores. La causa más común es el accidente cerebrovascular. Un derrame cerebral ocurre cuando se forma un coágulo y se obstruye el flujo normal de la sangre o cuando un vaso sanguíneo se rompe, corta o interrumpe el flujo de sangre. Los accidentes cerebrovasculares perinatales, se producen en los niños después de los tres días del nacimiento, y puede causar parálisis cerebral en niños. Los límites de la parálisis cerebral no funcionan específicamente como la parálisis total, sino más bien como los espasmos incontrolables. La *American Heart Association-Stroke Outcome Classification (AHA-SOC)* sistematiza los déficits neurológicos en seis dominios o áreas: motora, sensitiva, comunicación, visual, cognitiva y emocional. (3)

Alteraciones Motoras: Deficiencias orgánicas del aparato motor o de su funcionamiento, que afectan al sistema óseo, articulaciones, nervios y/o músculos. Las personas afectadas por ellas presentan una clara desventaja en su aparato locomotor,

Capítulo 1: Fundamentación Teórica

determinado por limitaciones posturales, de desplazamiento, coordinación y manipulación, pudiendo integrar dos o más de éstas. (4)

Déficit Sensitivo: Es un “defecto en la función de uno o más sentidos” (5).

Trastorno de la Comunicación: Está relacionado con alteraciones en el habla o la ausencia de este producto de algún daño cerebral.

Déficit Visual: Puede ir desde la ceguera total hasta cualquier otra afectación que dificulte la visión, no necesariamente tiene que ser por afectaciones en el ojo, sino producto de una lesión cerebral.

Trastorno Cognitivo: No es más que la manifestación de un deterioro en el rendimiento intelectual de la persona.

Déficit Emocional: Las personas pueden presentar problemas para controlar sus emociones o llegar a expresar algunas inapropiadas en determinadas situaciones. La depresión es un problema muy frecuente, esta puede dificultar la recuperación y la rehabilitación. (6)

El tratamiento de la hemiplejia está dirigido a la recuperación de las funciones corporales, así como de la prevención de complicaciones que puedan surgir. Algunas de estas opciones de tratamiento incluyen: (7)

- Medicamentos recetados.
- Terapia física y ocupacional.
- Dispositivos médicos.
- Intervención quirúrgica.

La correcta recuperación de un paciente hemipléjico depende de la mezcla de varios de los procedimientos anteriores. Cada uno de estos constituye un área bastante amplia en el proceso de diagnóstico y tratamiento de pacientes con este padecimiento. En el presente trabajo se aborda sobre las alteraciones motoras y las técnicas que se emplean

Capítulo 1: Fundamentación Teórica

en su tratamiento, que están relacionadas en su mayoría con la terapia física y ocupacional.

Para apoyar a los especialistas encargados de realizar esta labor se pudiera hacer uso de las potencialidades que brindan las nuevas tecnologías. Un sistema que sea capaz de asesorar a los doctores en el momento de asignar a un paciente determinado tratamiento sería de gran utilidad. Este sistema utilizaría los datos de pacientes que hayan sido previamente tratados, y tendría la capacidad de analizar la información para proponer una terapia al nuevo paciente. Este tipo de sistemas se construyen empleando técnicas de IA y son los denominados SBC.

1.3 Inteligencia Artificial y Sistemas Basados en el Conocimiento

El término Inteligencia Artificial se refiere de forma general a la capacidad de las máquinas y/o programas de ejecutar acciones y realizar razonamientos similares al de los seres humanos, esta actúa sobre problemas poco estructurados y de los cuales no se conoce a priori cual es la solución más eficaz o es muy difícil de definir.

Existen distintas definiciones para este término una de estas plantea que “es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento inteligente” (8).

Otra definición describe que es “la capacidad de emular el pensamiento humano” (9) o sea lograr una representación algorítmica de las ideas y el razonamiento de los seres humanos ante determinado problema. Otros autores la definen como “la capacidad de computadoras o programas de operar en concordancia con el pensamiento humano ante determinados procesos” (10).

De forma general todos los conceptos que abordan sobre el tema defienden la idea de instruir a las máquinas con inteligencia humana para que sean capaces de realizar labores complejas. No obstante, la programación tradicional presenta algunos inconvenientes que imposibilitan que un sistema pueda imitar características humanas. Esto se evidencia en las ocasiones en que es necesario trabajar con datos incompletos o borrosos, también hay que destacar que los programas tradicionales basan su

Capítulo 1: Fundamentación Teórica

funcionamiento en lenguajes procedurales o secuenciales y no todos los problemas se resuelven del mismo modo. Para lograr solucionar esto surgen un conjunto de técnicas y métodos dentro de la IA, entre ellos los SBC.

Un Sistema Basado en el Conocimiento trabaja sobre un dominio o área en particular e imita el desempeño de un especialista en esa área del conocimiento. Una definición de SBC explica que es un “sistema computarizado que usa conocimiento sobre un dominio para arribar a una solución de un problema de ese dominio. Esta solución es esencialmente la misma que la obtenida por una persona experimentada en el dominio cuando se enfrenta al mismo problema” (11). Una definición más técnica plantea que “son un modelo computacional de más alto nivel que el paradigma de la programación convencional en el cual los sistemas están formados por tres componentes: la base de conocimientos (BC), la máquina de inferencia (MI) y la interfaz de usuario” (12).

Estos sistemas se implementan mayormente para apoyar la toma de decisiones pues reducen en gran medida el error humano. Lo anterior se debe a que son capaces de extraer información oculta entre los datos, pueden operar con grandes cantidades de estos y mejorar los tiempos de respuesta con respecto a los humanos. Además, facilitan la corrección del conocimiento puesto que este está almacenado en una base de conocimientos y resulta sencillo modificar los datos.

Entre las características que distinguen a los SBC de otros sistemas están (11):

- La separación del conocimiento de cómo este es usado (distinción entre conocimiento y estrategia de control).
- El uso de conocimiento muy específico del dominio.
- Naturaleza heurística, del conocimiento empleado.
- No requieren analizar completitud, son capaces de aportar una solución a pesar de la ausencia de algunos datos.
- Pueden dar múltiples soluciones.

Dentro del amplio campo donde operan los SBC se destacan las siguientes áreas (10):

Juegos: Para diseñar estrategias, de manera que la dificultad vaya aumentando a medida que el jugador mejora el desempeño.

Capítulo 1: Fundamentación Teórica

Robótica: Es uno de los campos donde más se emplean. Dan al robot la habilidad de interactuar de manera inteligente con el medio donde se desempeña.

Sistemas de planeación: Se emplean para realizar planificaciones que ahorren presupuesto en proyectos o procesos complejos.

Reconocimiento de imágenes: En imágenes médicas se aplican para identificar células cancerígenas o para evaluar otro padecimiento.

Traductores: Se emplean en aplicaciones de reconocimiento de voz, también en sistemas para traducir o identificar el contenido de textos antiguos.

Sistemas tutoriales: En aplicaciones educativas o de instrucción sobre determinada temática.

Medicina: En el diagnóstico y tratamiento de diferentes enfermedades.

Economía: Para realizar planes económicos estratégicos, hacer predicciones e identificar clientes potenciales.

Sistemas Expertos: Sistemas que son capaces de resolver problemas sobre determinado dominio de forma parecida a un ser humano.

Los SBC pueden presentar diferentes estilos de trabajo según la finalidad con que son desarrollados o el ambiente donde operan: (11)

- Asistente: Invocado por el usuario para resolver una tarea específica como parte de un ejercicio más amplio.
- Crítico: Revisa trabajos realizados por otros y comenta su precisión y consistencia.
- Segunda opinión: Ejecuta una tarea y compara sus resultados con los del usuario.
- Consultante: Advierte o previene al usuario.
- Tutor: Entrena al usuario en ejecutar una tarea experta.
- Autómata: Completa tareas automáticamente e independientemente del usuario.

Capítulo 1: Fundamentación Teórica

Existen también un conjunto de elementos que son los que justifican el desarrollo de un SBC sobre determinada área de trabajo, estos son: (11)

- Sitios donde la experticia humana puede perderse.
- Donde la disponibilidad de expertos es escasa y se necesitan en muchos lugares.
- Ambientes hostiles donde se necesiten expertos para trabajar.
- Donde se requiera mejorar la calidad del conocimiento de los expertos humanos.

1.3.1 Tipos de SBC

La arquitectura y desarrollo de un SBC varía en dependencia de la forma de representación del conocimiento que más se ajuste a la situación. La siguiente tabla muestra distintos tipos de SBC y sus principales características.

Nombre	Forma de Representación del Conocimiento	Explicación	Método de Solución del Problema	Fuentes de Conocimiento
Sistemas Basados en Reglas (SBR)	Reglas de Producción	Reglas Activas	Usualmente busca primero en profundidad con dirección hacia delante o hacia atrás	Expertos, publicaciones, ejemplos
Sistemas Basados en Frames (SBF)	Frames		Herencia y procedimientos adjuntos	Expertos, publicaciones, ejemplos
Sistemas Basados en	Casos	Casos semejantes	Razonamiento	Ejemplos

Capítulo 1: Fundamentación Teórica

Casos (SBCasos)			basado en casos (búsqueda por semejanza y adaptación de las soluciones)	
Sistemas basados en probabilidades (SBP)	Probabilidades o frecuencias	Valores de probabilidades condicionales	Teorema de Bayes y otras técnicas de inferencia estadística	Ejemplos
Redes expertas	Pesos y alguna otra FRC	Según el modelo simbólico	Cálculo de niveles de activación de las neuronas	Ejemplos
Sistemas basados en modelos	Modelo del artefacto		Razonamiento basado en modelos	Esquemas estructurales y funcionales del artefacto

Tabla 1.1. Tipos de Sistemas Expertos. (13)

De los tipos de SBC descritos anteriormente, destacan en el área de las ciencias médicas dos: los SBR y los SBCasos. (13)

Sistemas Basados en Reglas (SBR)

Los SBR operan mediante reglas de producción de forma “SI ENTONCES”, donde la primera parte es llamada condición o antecedente y la otra conclusión o consecuente, lo que se expresa como “SI condición ENTONCES conclusión”. Cuando se cumple la condición de una regla la conclusión pasa a formar parte del conocimiento y es empleada por la máquina de inferencia para verificar el cumplimiento de nuevas reglas. Para dar solución a un problema se relacionan las reglas mediante una cadena de inferencias lo

Capítulo 1: Fundamentación Teórica

cual no es más que un camino desde la definición del problema hasta la solución que brinda el sistema, este camino puede construirse de 2 maneras: (14)

1. Seleccionar una conclusión posible y tratar de probar su validez buscando evidencias que la soporten (encadenamiento hacia atrás).
2. Comenzar con todos los datos conocidos y progresar hacia la conclusión (encadenamiento hacia delante).

Encadenamiento hacia atrás o *backward*

Este método consiste en verificar la existencia de un conjunto de reglas que prueben el cumplimiento de una hipótesis planteada, este opera de la siguiente forma: (14)

1. Se define una hipótesis, es decir, un valor a alcanzar en una variable de salida del sistema.
2. Si la hipótesis se basa en un hecho comprendido en la base de conocimiento, se finaliza el proceso. En caso contrario se continúa al paso 3.
3. Se busca el subconjunto de reglas de la base de conocimiento cuyos consecuentes coincidan con la hipótesis.
4. Se establecen los antecedentes de las reglas del subconjunto seleccionado como nuevas hipótesis a demostrar y se vuelve al paso 2.

Encadenamiento hacia adelante o *forward*

Este método utiliza los valores de las variables de entrada para ir infiriendo reglas hasta arribar a una solución deseada, su funcionamiento es el siguiente: (14)

1. Se define el hecho a alcanzar, es decir, las variables de salida del sistema cuyo valor se desea inferir.
2. Un conjunto de hechos relativo a las variables de entrada es observado por el sistema, es decir, llega una entrada al sistema.
3. Se busca el subconjunto de reglas de la base de conocimiento cuyos antecedentes son satisfechos por los hechos observados.

Capítulo 1: Fundamentación Teórica

4. Si el subconjunto está vacío, se finaliza el proceso. En otro caso se continúa al paso 5
5. El subconjunto de reglas seleccionado se activa y da lugar a un número de hechos nuevos igual al tamaño del subconjunto.
6. La base de hechos se actualiza con los nuevos hechos.
7. Si se ha alcanzado el hecho de salida deseado se finaliza el proceso, en otro caso, se vuelve al paso 3.

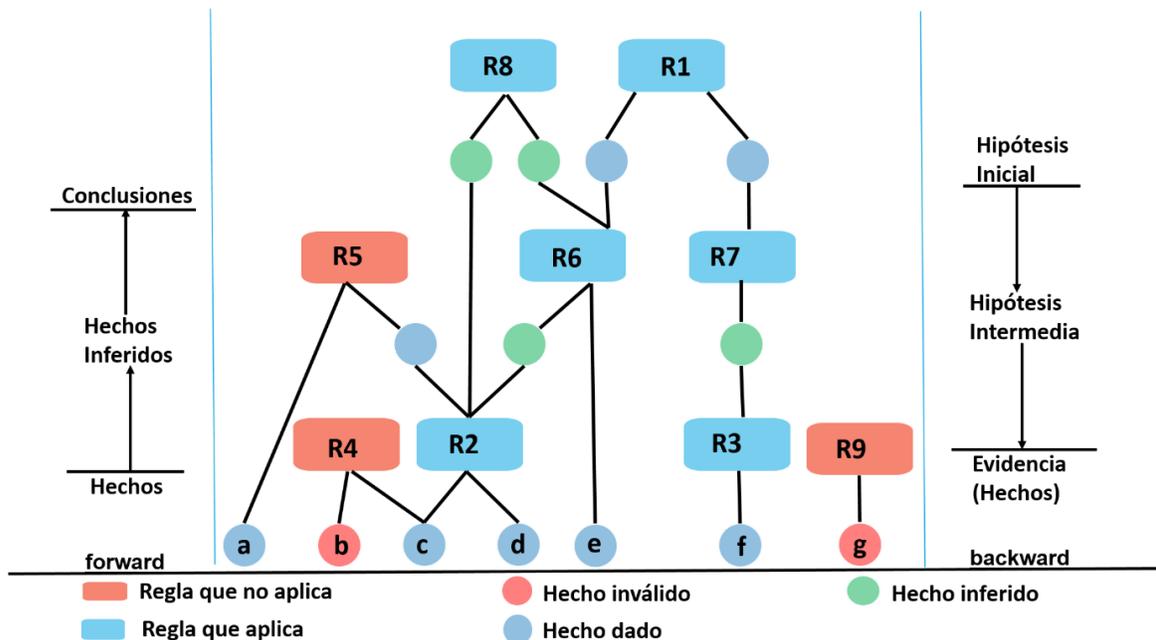


Figura 1.1. Estrategias de búsqueda de un SBR. (13)

Sistemas Basados en Casos (SBCasos)

Los SBCasos imitan la característica que tienen los seres humanos de buscar solución o arribar a conclusiones de un problema a través de experiencias previas en situaciones similares. Los elementos de mayor importancia en este tipo de sistemas son la base de conocimientos o casos y el motor de inferencia. Existen dos clasificaciones para los SBCasos, sistemas interpretativos y sistemas que solucionan problemas. Los primeros toman una solución de un problema que no está bien comprendido y brindan como salida un argumento o crítica sobre esta solución, normalmente se emplean en problemas de

Capítulo 1: Fundamentación Teórica

clasificación, argumentación o predicción. Los solucionadores son los que mediante la información que obtienen del problema construyen su solución analizando la base de casos.

Para que un SBCasos encuentre una solución a un nuevo problema debe ejecutar las siguientes etapas:

Recuperar: Obtiene los casos similares almacenados en la base de casos.

Reutilizar: Utiliza los casos recuperados para confeccionar la solución de una nueva entrada. Esta nueva solución no necesariamente tiene que ser la misma que en algún caso recuperado puesto que pueden adaptarse.

Revisar: Verifica si la solución encontrada es correcta o si contiene algún aspecto específico que deba ser transformado. El experto en el dominio valida finalmente si la solución es viable o no.

Recordar: Guardar la nueva solución en la base de casos para que pueda ser utilizada en futuras consultas.



Figura 1.2. Estrategias de búsqueda de un SBCasos. (13)

Capítulo 1: Fundamentación Teórica

Los casos se representan mediante un conjunto de características que definen al problema en cuestión, estas características se dividen en: rasgos predictores, y la solución brindada según los valores de dichos rasgos. De acuerdo a la situación que se quiera representar en una base de conocimientos se define la estructura de los casos. Para esto se tiene en cuenta cuáles son sus características más relevantes o las de mayor influencia para la solución. Se deben definir además los algoritmos que van a operar en el sistema en cada una de las etapas, desde la recuperación hasta la validación de la solución, para determinar si el nuevo caso pasa o no a formar parte de la base de conocimientos. Para poder identificar casos similares al que se trata de resolver, es necesario valerse de un método de comparación eficaz:

Funciones de semejanza para los casos donde las características que los representan son numéricas. (13)

- Distancia Euclidiana
- Distancia de Manhattan
- Distancia de Minkowsky
- Coeficiente de Duran y Odell

Funciones de semejanza para los casos donde las características que los representan son simbólicas. (13)

- Distancia de Hamming
- *Value Difference Metric* (VDM)

Funciones para calcular la semejanza en dominios de problemas donde las características son numéricas y no numéricas incluyendo información incompleta. Se considera como información incompleta, los valores desconocidos en un momento dado para un atributo. (13)

- Función de Gower
- Función de Argelio

Capítulo 1: Fundamentación Teórica

Los SBC se desempeñan de diferentes maneras, no obstante, cuando estos están encaminados específicamente a resolver problemas en un entorno de la misma forma que lo haría un especialista, empleando para ello conocimiento de otros especialistas, entonces a este tipo de sistema se le denomina Sistema Experto.

1.3.2 Sistemas Expertos

Un Sistema Experto (SE) es un tipo particular de Sistema Basado en el Conocimiento, de hecho, son considerados “la élite de los Sistema Basado en el Conocimiento” (1) . El nombre deriva de “Sistema Experto Basado en el Conocimiento” (9), estos se emplean para realizar tareas que solo pueden ejecutar un limitado número de personas expertas. Para esto, captan los conocimientos de los expertos mediante técnicas de IA y los utilizan para brindar soluciones más completas a los problemas donde se aplican. A continuación, se brindan 3 definiciones sobre este tipo de sistemas:

“Un Sistema Experto es un sistema computacional que adquiere conocimiento especializado en un campo específico para explotarlo mediante métodos de razonamiento que emulan el desempeño del experto humano en la solución de problemas” (1).

Según Giarratano Riley autor del libro “Sistemas Expertos: Principios y Programación” los SE son “un sistema de cómputo que emula la habilidad de tomar decisiones de un especialista humano” (15).

Para dar una definición más clara y entendible a personas que no tengan dominio del lenguaje técnico se puede definir a los SE como “un sistema que emplea conocimiento humano capturado en una computadora para resolver problemas que normalmente requieran de expertos humanos” (9).

Todas las definiciones anteriormente expuestas reflejan que la esencia de este tipo de sistemas consiste en imitar a las personas denominadas expertas en cierta área del conocimiento. La persona encargada de proveer este conocimiento al SE es el ingeniero del conocimiento y su labor consiste en extraer la información del especialista humano, representarla de forma tal que pueda ser usada por equipos de cómputo e incluirla en el SE.

Capítulo 1: Fundamentación Teórica

Estructura de un Sistema Experto

Los siguientes son componentes básicos de un SE: (9)

- **Subsistema de adquisición de conocimiento**

Es la acumulación, transferencia y transformación de la experiencia en la resolución de problemas a un programa de computadora para construir o expandir una base de conocimientos. El estado del arte actual requiere un ingeniero en conocimiento que interactúe con uno o más expertos humanos para construir la base de conocimiento.

- **Base de conocimientos**

Contiene el conocimiento necesario para comprender, formular y resolver problemas. Incluye dos elementos básicos: heurística especial y reglas que dirigen el uso del conocimiento para resolver problemas específicos en un dominio particular.

- **Base de hechos**

Es una memoria de trabajo que contiene los hechos sobre un problema específico que se resuelve y que han sido probados o inferidos.

- **Motor de inferencia**

Es el cerebro del SE, también conocido como estructura de control o interpretador de reglas. Es esencialmente un programa de computadora que provee metodologías para razonamiento de información en la base de conocimiento. Este componente provee las estrategias sobre cómo usar el conocimiento y controla los pasos para resolver el problema cuando se realiza una consulta.

- **Subsistema de justificación**

Se encarga de explicar el comportamiento del SE al encontrar una solución. Permite al usuario hacer preguntas al sistema para poder entender las líneas de razonamiento que este siguió. Resulta especialmente beneficioso para usuarios no expertos que buscan aprender a realizar algún tipo de tarea.

Capítulo 1: Fundamentación Teórica

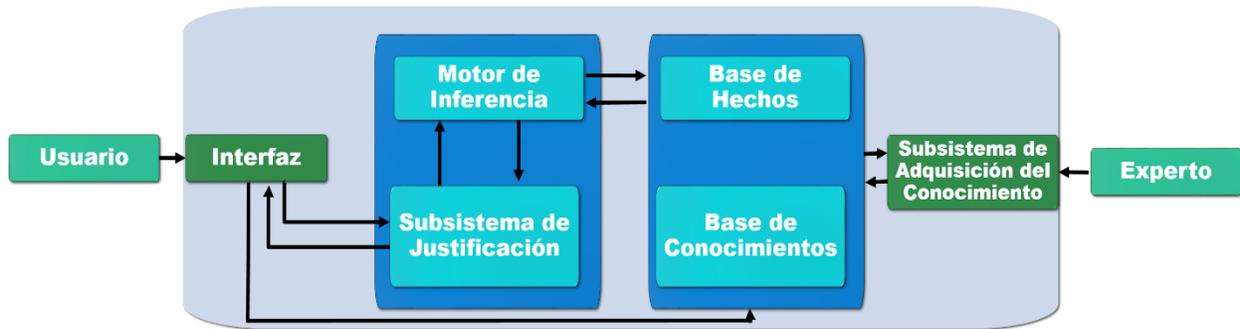


Figura 1.3. Estructura de los Sistemas Expertos. (9)

A la hora de implementar o utilizar algún sistema de este tipo es importante conocer las ventajas y limitaciones que pueden presentar, esto da una medida de cuán fiable puede ser. De modo general, las más significativas son: (9) (11)

Ventajas

- Son herramientas estables y fiables (no se enferman o envejecen).
- Sus actividades son completamente replicables (siempre contesta de la misma manera a menos que se le cambie el diseño).
- Solucionan problemas que incluyen datos incompletos.
- Velocidad de procesamiento mayor a la de un ser humano.
- Pueden ser utilizados por personas no especializadas para resolver problemas.
- Las personas que lo utilizan con frecuencia pueden aprender de ellos.
- Resultan una inversión rentable, aunque los costos iniciales sean altos.

Limitaciones

- Para actualizarlos se necesita la reprogramación de estos, lo que implica modificar su código fuente para incluir o renovar funciones.

Capítulo 1: Fundamentación Teórica

- Las respuestas no siempre son correctas, la base de conocimientos puede contener casos con soluciones erróneas que consigan alterar los resultados.
- No logran abarcar todo el dominio.
- Poco flexibles a cambios y de difícil acceso a información no estructurada.
- Carecen de sentido común.
- No son capaces de distinguir cuáles son las cuestiones relevantes de un problema y separarlas de cuestiones secundarias.

1.4 Sistemas Basados en el Conocimiento usados en el diagnóstico médico

El área de ciencias médicas ha sido una de las más beneficiadas con el desarrollo de los SBC. De hecho, los primeros fueron implementados para apoyar a los médicos en los procesos de diagnóstico de enfermedades. Entre estos se encuentran:

Sistema Experto para el Diagnóstico de Enfermedades Infecciosas en la Sangre (MYCIN)

Es un sistema desarrollado por *Edgar Short Liffe* de la Universidad de Stanford, escrito en lenguaje de programación *ListProcessing* (LISP) e inspirado en el SE Dendral. El mismo es capaz de imitar el proceso de diagnóstico de los especialistas para arribar a conclusiones y recetar medicamentos personalizados a cada paciente. “El sistema diagnostica la causa de la infección usando el conocimiento relativo a la infección de los microorganismos, a partir de historiales de pacientes, síntomas y los resultados de los test de laboratorio” (16). MYCIN es muy eficaz en el trabajo con información incompleta o incorrecta, es capaz de explicar sus respuestas y puede interactuar con los humanos mediante lenguaje natural.

Sistema Experto de Ayuda Asistencial (SEAA)

Es un sistema inteligente destinado al médico de cabecera y cuyo propósito es auxiliarlo en el diagnóstico clínico de alrededor de 700 enfermedades y síndromes que abarcan

Capítulo 1: Fundamentación Teórica

los diferentes aparatos comprendidos en la anatomía humana (digestivo, respiratorio, cardiovascular, locomotor, endocrino, neurológico, etc.) (17). Desarrollado por la Empresa Productora de Software para la Técnica Electrónica (SOFTTEL), empresa cubana pionera en el desarrollo de aplicaciones informáticas. Se confeccionó mediante la extracción de conocimientos a especialistas en medicina interna y aunque no pretende suplantar a estos, sí es empleado por muchos doctores del país como sistema de ayuda diagnóstica. Opera mediante los síntomas iniciales referidos por el paciente como motivo de la consulta, los antecedentes personales y familiares extraídos de la Historia Clínica y del interrogatorio correspondiente (17). El sistema es capaz de imitar la habilidad de los médicos y brindar un tratamiento efectivo para el diagnóstico en cuestión. Está implementado como un SBR y permite inferir conocimiento partiendo de los hechos para arribar a conclusiones (*forward*) o a partir de hipótesis determinar hechos (*backward*).

Diagnos

Es una potente herramienta de utilidad continua en la consulta, que ayuda al diagnóstico al combinar un conjunto de datos (síntomas, signos, resultados analíticos anormales, etc.) con el país, sexo y edad del paciente, ofreciendo con criterio un listado de enfermedades posibles, con potentes herramientas para afinar en el diagnóstico. Tiene también importantes utilidades para ayudar a evitar errores en la medicación (contraindicaciones, interacciones entre fármacos, doble terapia, entre otros). Esto resulta de gran utilidad dada la frecuente poli-medicación, y en pacientes con un tratamiento mayor de seis fármacos es muy probable encontrar interacciones, y Diagnos las presenta en orden de importancia, además permite consultar rápidamente enfermedades, dietas para enfermedades, imágenes, protocolos en atención primaria y en urgencias (con los criterios de ingreso hospitalario), pruebas de laboratorio, fármacos, interacciones, contraindicaciones, plantas medicinales, etc. (18) Esta aplicación no realiza diagnósticos en sí, sino que provee al médico de una herramienta potente para obtener respuestas ante determinadas dudas que puedan surgir en el proceso de diagnóstico.

Capítulo 1: Fundamentación Teórica

Arquitectura de planificación, ejecución y aprendizaje (PELEA)

Es una arquitectura genérica para la integración de planificación, ejecución, monitorización, re-planificación y aprendizaje. Está específicamente pensada para utilizarse en robótica, aunque también sirve en otros ámbitos (19). Permite la confección de terapias para la rehabilitación mediante técnicas de planificación y utiliza un robot como medio para interactuar con el paciente. Cuenta con algoritmos que le permiten llevar un seguimiento de los pacientes y tomar decisiones según su trayectoria, se apoya de distintos dispositivos instalados en el robot como cámaras y sensores.

Los sistemas descritos ejemplifican el aporte significativo que brindan los SBC a procesos de toma de decisiones en la medicina. En estos están presente varias técnicas que han demostrado ser eficaces en la implementación de SE, como son los SBCasos y los SBR. No obstante, en el estudio del arte realizado no se encontró evidencia de SBC encaminados a la rehabilitación de pacientes hemipléjicos con alteraciones motoras. La arquitectura (PELEA) está dirigida a la rehabilitación de pacientes, pero su campo de acción son lesiones que pueden producirse en los hombros de los niños durante el proceso del parto (19).

1.5 Herramientas y Tecnologías

Metodología de desarrollo de software

La mayoría de los autores que abordan los temas de metodologías de desarrollo de software coinciden en que el éxito de determinado producto depende en gran medida de la estrategia seguida durante el proceso de desarrollo. “Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software, indicando quién debe hacer cada actividad, cuándo hacerla y qué debe hacer” (20). Estas metodologías tienen dos clasificaciones principales, metodologías ágiles y metodologías tradicionales o pesadas.

Las principales características presentes en las metodologías de desarrollo ágiles son las siguientes: (21)

- Valoran al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un

Capítulo 1: Fundamentación Teórica

proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido.

- Se empeñan en que el software funcione más que en conseguir una buena documentación. Aunque se parte de la base de que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- Se centran en la colaboración con el cliente más que en la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- Responden a los cambios más que a seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir.

Entre las metodologías de desarrollo ágil que más destacan se encuentran las siguientes:

Proceso Ágil Unificado (AUP)

La metodología de desarrollo de software, AUP (del inglés *Agile Unified Process*) es la versión ágil del Proceso Unificado de Rational (RUP). Esta describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. (22)

Capítulo 1: Fundamentación Teórica

Programación Extrema (XP)

La Programación Extrema (del inglés *Extreme Programming, XP*) es una de las metodologías de desarrollo de software más exitosas para proyectos de corto plazo y equipos pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (23)

Las metodologías tradicionales o pesadas se caracterizan por estar orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar herramientas a utilizar y notaciones que se usarán. (20)

Entre las metodologías tradicionales más empleadas se encuentran:

Rational Unified Process (RUP)

RUP es un proceso formal, provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales respetando cronograma y presupuesto. (24)

Microsoft Solution Framework (MSF)

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información. (24)

En el presente trabajo se opta por emplear la metodología ágil Programación Extrema (XP), debido a que el equipo de trabajo es pequeño y no se pueden abarcar todos los roles descritos por otras metodologías como RUP o MSF. El tiempo para entregar la aplicación es limitado y debe existir la disponibilidad de responder de manera ágil a cambios bruscos que puedan surgir en la lógica del negocio. Se cuenta además con la facilidad de que el cliente esté involucrado en el equipo de desarrollo, lo que facilita el

Capítulo 1: Fundamentación Teórica

intercambio continuo de este con el equipo de desarrollo y una mejor comprensión del negocio.

1.5.1 Desarrollo mediante XP:

El desarrollo de aplicaciones informáticas utilizando la metodología XP puede ser muy dinámico pero se puede separar en un conjunto de pasos o etapas a seguir para la construcción del software, estas son: (25)

1. **Exploración:** Los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto y los casos de prueba de las mismas. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.
2. **Planificación:** Se priorizan las historias de usuario, los programadores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma. Las pruebas funcionales creadas por el cliente se ejecutan al final de cada iteración. Al final de la última iteración el sistema está listo para producción.
3. **Iteraciones:** Se implementan las funcionalidades del sistema, al final de cada una de estas iteraciones se realiza una entrega funcional del producto. Como las historias de usuario no ofrecen una descripción detallada de la funcionalidad en esta etapa se redactan las tareas de ingeniería necesarias para explicar concretamente qué se desea hacer.
4. **Puesta en Producción:** Al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste.

Roles que intervienen: (26)

- **Programador:** Es una pieza clave en XP. Su responsabilidad no se limita a implementar cierta funcionalidad del sistema; él también debe comunicarse, ya

Capítulo 1: Fundamentación Teórica

sea con otros miembros del equipo de desarrollo o con el cliente, elaborar pruebas unitarias y llevar a cabo las integraciones del sistema.

- **Cliente:** El cliente es la otra mitad de la importante dualidad de XP. El programador sabe cómo programar. El cliente sabe qué programar. El cliente debe escribir las historias de usuario y las pruebas funcionales del sistema, asignarles prioridad a las historias de usuarios y tomar decisiones acerca de cuál se debe implementar en cada iteración.
- **Encargado de pruebas:** Verifica que el sistema esté funcionando correctamente. Debe ejecutar regularmente todos los casos de prueba, informar al equipo los resultados obtenidos y ayudar al cliente a escribir las pruebas funcionales del sistema.
- **Entrenador:** Es quien advierte si ocurre una desviación en el proceso, guía al equipo de desarrollo para que se siga el proceso XP correctamente.
- **Encargado de seguimiento:** Debe verificar el cumplimiento del plan de entrega y del plan de iteración, e informar si las estimaciones realizadas fueron correctas, se subestimó o sobrestimó; con el objetivo de que el equipo sea más preciso en futuras estimaciones.
- **Consultor:** Es responsable de guiar al equipo de desarrollo para resolver los problemas que se presenten en un tema específico. Es un miembro externo al equipo con conocimientos específicos en algún tema necesario para la construcción del sistema.

Artefactos que se generan: (27)

- **Historias de Usuario:** Representan una breve descripción del comportamiento del sistema y emplea terminología del cliente sin lenguaje técnico. Se realiza una por cada característica principal del sistema y se emplean para hacer estimaciones de tiempo para el plan de entregas. Difieren de los casos de uso porque son escritas empleando terminología del cliente, son más amigables que los casos de uso formales.

Capítulo 1: Fundamentación Teórica

- **Pruebas de Aceptación:** Permite confirmar que la historia ha sido implementada correctamente.
- **Tarjetas CRC:** Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

Esta metodología también define un conjunto de prácticas que son esenciales para el éxito de los proyectos y no deben pasarse por alto en el proceso de desarrollo, estas son:

Recodificación: Consiste en escribir nuevamente parte del código del programa sin alterar su función, esto provoca que la función sea más simple y entendible y ayuda a mantener el código lo más simple posible en caso de que se requiera hacer cambios.

Pruebas Unitarias: Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben pasar las pruebas unitarias antes de ser liberados o publicados. (25)

Reuniones diarias de seguimiento: El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones.

Lenguaje de Programación

Los lenguajes de programación son los que permiten crear aplicaciones informáticas. Mediante estos se codifica la lógica de funcionamiento del producto final. Para el desarrollo del Sistema Basado en el Conocimiento se emplea como lenguaje de programación Java, este se adapta a las necesidades requeridas para el proceso de implementación. “Es un lenguaje de propósito general, concurrente, basado en clases y orientado a objetos. Está diseñado para ser lo suficientemente simple para que los programadores puedan lograr la fluidez en el mismo.” (28)

JavaFX

“JavaFX es un conjunto de paquetes de gráficos y multimedia que permite a los desarrolladores diseñar, crear, probar, depurar y desplegar aplicaciones de cliente, que operan constantemente a través de diversas plataformas.” (29) Facilita notablemente la

Capítulo 1: Fundamentación Teórica

construcción de las interfaces de usuario de la aplicación. Son construidas con la herramienta *Scene Builder*.

Scene Builder

“Es una herramienta de diseño visual que permite a los usuarios diseñar rápidamente interfaces de usuario de las aplicaciones JavaFX, sin necesidad de programación. Los usuarios pueden arrastrar y soltar los componentes de interfaz de usuario a un área de trabajo y modificar sus propiedades. Se aplican las hojas de estilo (ccs), y el código FXML se genera automáticamente en segundo plano según los elementos que se vayan creando. El resultado es un archivo FXML que a continuación se puede combinar con un proyecto de Java mediante la unión de la interfaz de usuario a la lógica de la aplicación.”

(30)

Entorno de Desarrollo Integrado (IDE)

Como entorno de desarrollo se emplea Netbeans IDE versión 8.1, “es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación” (31). Provee facilidades para tener el control de lo que se está implementando. Proporciona herramientas para la realizar pruebas y permite la interacción con otros programas.

JUnit

Es una Herramienta especialmente diseñada para implementar y automatizar la realización de pruebas unitarias en Java (32). Contiene un conjunto de librerías que facilitan la realización de dichas pruebas, necesarias para verificar que los métodos implementados se desempeñen de forma correcta.

Framework

La palabra *framework* define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. Los objetivos principales que persigue un *framework* para el desarrollo de software son:

Capítulo 1: Fundamentación Teórica

acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones (20). En este trabajo se utiliza el *framework Hibernate*, el cual facilita el trabajo con bases de datos.

Hibernate

Es una herramienta para hacer corresponder objetos de una aplicación con las entidades que los hacen persistentes en una base de datos relacional. Se emplea en plataformas Java y según el sitio oficial del mismo es “una herramienta que permite a los desarrolladores escribir más fácilmente aplicaciones cuyos datos sobreviven al proceso de solicitud” (33). *Hibernate* tiene que ver con la persistencia de datos que se aplica a las bases de datos relacionales.

Gestor de Bases de Datos

Las herramientas encargadas de operar con las Bases de Datos (BD) son las que se conocen como Sistemas de Gestión de Bases de Datos (SGBD). Estos llevan el control de los datos que se guardan una vez que las aplicaciones que se estaban ejecutando se han cerrado. Permiten crear copias de seguridad de la Base de Datos y restaurar otras ya existentes, además, añaden seguridad a las mismas mediante el establecimiento de usuarios y contraseñas que limitan el acceso. Para la gestión de la Base de Datos del SBC desarrollado en esta investigación, se emplea PostgreSQL en su versión 9.2.

PostgresSQL

Es un Sistema de Gestión de Bases de Datos de código abierto, “utiliza un modelo cliente-servidor y usa multi-procesos en lugar de multi-hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará al resto y el sistema continuará funcionando” (34). Es capaz de soportar grandes cantidades de datos sin que se afecte su rendimiento, además puede operar sobre la mayoría de los sistemas operativos. Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos almacenados (20).

Capítulo 2: Propuesta de Solución

En este capítulo se describe un sistema propuesto para dar solución al problema planteado en la presente investigación. Se especifican las técnicas empleadas para la adquisición del conocimiento, se define la estructura de los casos y la forma en que están organizados en la base de conocimientos. Además, se detallan las técnicas empleadas para el procesamiento de los datos, funciones de recuperación, comparación, adaptación, y la estrategia de aprendizaje del sistema. También se exponen las características del módulo de recolección y gestión de datos, sus funcionalidades y características. De la metodología empleada se describen sus fases, así como las historias de usuario (HU) más importantes. Por último, se abordan los patrones arquitectónicos y de diseño que fueron empleados en la construcción del SBC.

2.1 Sistema Propuesto

Para dar solución a la problemática planteada se propone el desarrollo de un SE basado en casos. Estos ofrecen la posibilidad de trabajar mediante experiencias previas de especialistas y son capaces de confeccionar propuestas de solución para casos similares a los que contiene en su base de conocimientos. A pesar de las limitaciones que presentan estos de que pueden arrojar soluciones erróneas, es factible su aplicación en esta investigación ya que se cuenta con datos registrados durante un estudio de casos publicado en (35). Lo anterior garantiza que se cuente con información fiable referente a la atención de pacientes hemipléjicos con alteraciones motoras y reduce el riesgo de que el sistema desarrollado aporte soluciones incorrectas.

El proceso de consulta a los pacientes requiere de la recopilación de sus datos personales y clínicos, estos últimos son los que manejará el sistema para indicar una propuesta tratamiento. La aplicación está conformada por tres módulos que permiten llevar este proceso (figura 2.1). El primero de estos es el de gestión de datos, permite al usuario registrar y manipular la información de sus pacientes y el respectivo historial clínico de estos. Un conjunto de interfaces de usuario facilita esta tarea. El segundo módulo corresponde al agente inteligente, contiene los algoritmos empleados para

Capítulo 2: Propuesta de Solución

realizar inferencia y confeccionar la propuesta de tratamiento. Este proceso está dividido a su vez en tres etapas, Recuperación, Reutilización y Adaptación, que aparecen descritas más adelante en este documento. El último módulo es el de explicación, su función es mostrar al usuario los resultados obtenidos por el agente inteligente en cada una de sus etapas.

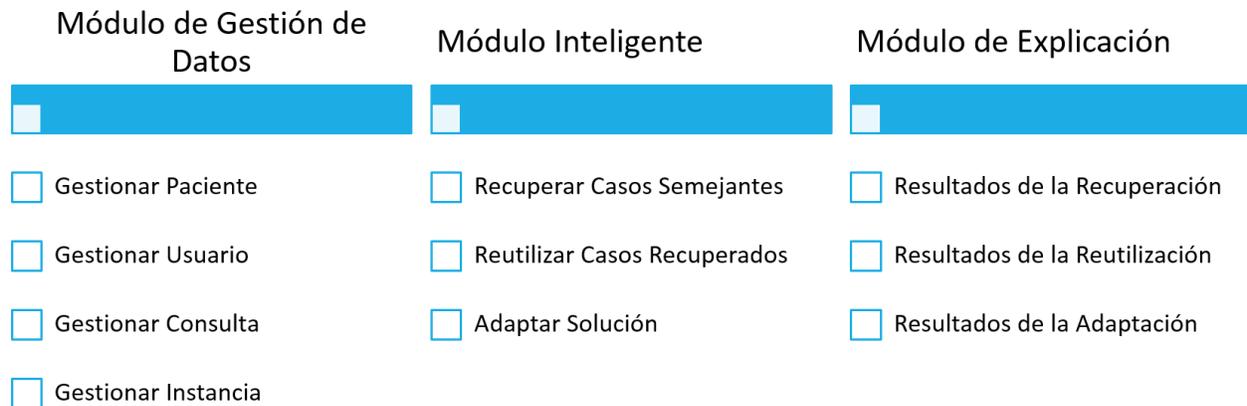


Figura 2.1 Módulos que componen la solución propuesta.

La figura 2.2 muestra como resultaría el proceso de consulta y tratamiento de pacientes hemipléjicos con alteraciones motoras con ayuda del SBC; el paciente llega a la consulta médica, el médico le realiza la entrevista y los test correspondientes e introduce los datos en el sistema, este los analiza, realiza una propuesta de tratamiento para el caso recién llegado y finalmente el médico decide si le da ese tratamiento o no al paciente.



Figura 2.2 Flujo del proceso de diagnóstico y tratamiento de pacientes mediante el SBC.

Capítulo 2: Propuesta de Solución

2.2 Adquisición del conocimiento

El proceso de adquisición del conocimiento es una de las tareas fundamentales en el desarrollo de un SBC, mediante este proceso, el ingeniero de conocimiento confecciona y organiza la información necesaria para que el sistema pueda operar, este proceso transita por varias etapas que se muestran en la figura 2.3.

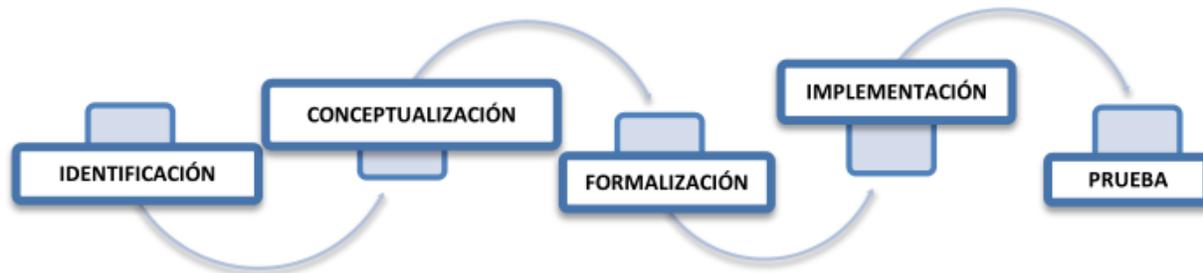


Figura 2.3 Etapas del proceso de adquisición del conocimiento. (13)

A continuación, se detallan cada una de las etapas anteriores:

Identificación: En esta etapa se identifican los participantes, tanto expertos como los ingenieros del conocimiento que intervendrán en la creación de la base de datos, se especifican además los aspectos fundamentales que se trabajarán para ubicar a los especialistas en lo que se desea hacer.

Conceptualización: Se especifican y relacionan los aspectos fundamentales que se trabajarán. Se identifica el flujo de información y los tipos de datos de que se dispone.

Formalización: Se define la forma en que se pudiese representarse la información, lo que llevaría también a identificar la herramienta necesaria para construir la base de conocimiento.

Implementación: Se codifica la información extraída de los expertos según la estrategia definida y la herramienta seleccionada.

Capítulo 2: Propuesta de Solución

Prueba: Se realizan pruebas a la base de conocimiento para identificar sus potencialidades y debilidades y con ello determinar si es necesario realizarles cambios al modelo o la forma de representación empleada.

2.2.1 Técnicas de Adquisición del Conocimiento

Las técnicas de adquisición del conocimiento son aquellas que permiten obtener el conocimiento necesario para conformar la base de datos con que el sistema va a funcionar. En esta investigación se emplearon las siguientes:

Consulta a Expertos: Este método consiste en realizar encuestas y entrevistas a especialistas en el tema para extraer el conocimiento de que disponen para incluirlo en el sistema. También para verificar la veracidad de los datos una vez conformada la base de conocimiento. En la conformación y validación de la base de casos del SBC se realizaron consultas y entrevistas a especialistas del Centro Nacional de Rehabilitación Julio Díaz de La Habana.

Conocimiento Documentado: El conocimiento es extraído de distintas fuentes tales como libros, revistas y otros documentos que contengan información sobre el tema. En la investigación en curso se emplearon datos de historias clínicas de pacientes hemipléjicos con alteraciones motoras y documentos e informes de investigaciones sobre el tema realizadas por especialistas del hospital Julio Díaz.

Mediante la consulta con especialistas y los documentos estudiados (35) se identificaron un conjunto de 36 variables (tabla 2.1) que son medidas en el diagnóstico a pacientes hemipléjicos con alteraciones motoras y que reflejan elementos indispensables a tener en cuenta para indicar un tratamiento. Estas variables se separan en cuatro grupos, las relacionadas con las características de las manos, las del paciente de pie, cuando está sentado y durante la marcha.

2.3 Base de Casos

Los casos son una de las formas en que puede ser representado el conocimiento para su posterior utilización. El Razonamiento Basado en Casos se basa en emplear datos de casos previos para extraer conocimientos y construir a partir de estos una solución a

Capítulo 2: Propuesta de Solución

nuevas situaciones. Para construir la base de casos es necesario inicialmente identificar los atributos que describirán cada uno de los casos. Estos se dividen en rasgos predictores y rasgos objetivo. Los primeros corresponden a los elementos que el especialista utiliza para caracterizar el estado del paciente y los segundos al tratamiento que se asigna a estos. De cada uno se debe especificar el dominio de definición y el tipo de dato correspondiente.

En el sistema que se implementa cada caso está compuesto por un número fijo de rasgos que se describen a continuación:

Rasgo Predictor	Dominio	Tipo de Dato
Postura de Pie		
Apoyo Externo	Ninguno, Bastón, Muleta, Asistente	Nominal
Nivel de Pelvis	Nivelado, Descenso Izquierdo, Descenso Derecho	Ordinal
Nivel de Hombros	Nivelado, Descenso Izquierdo, Descenso Derecho	Ordinal
Equilibrio Estático	Estable, Inestable con Micro-Oscilaciones, Inestable con Macro-Oscilaciones, Inestable con Movimientos Compensatorios	Ordinal
Equilibrio Dinámico	Estable, Inestable con Micro-Oscilaciones, Inestable con Macro-Oscilaciones, Inestable con Movimientos Compensatorios	Ordinal
Base de Sustentación Ampliada	Sí, No	Binario

Capítulo 2: Propuesta de Solución

Rotación de Miembro Inferior	Sí, No	Binario
Genus Flexus	Sí, No	Binario
Genus Valgo	Sí, No	Binario
Genus Varo	Sí, No	Binario
Genus Recurvatum	Sí, No	Binario
Postura Sentado		
Nivel de Hombros	Nivelado, Descenso Izquierdo, Descenso Derecho	Ordinal
Equilibrio Estático	Estable, Inestable con Micro-Oscilaciones, Inestable con Macro-Oscilaciones, Inestable con Movimientos Compensatorios	Ordinal
Equilibrio Dinámico	Estable, Inestable con Micro-Oscilaciones, Inestable con Macro-Oscilaciones, Inestable con Movimientos Compensatorios	Ordinal
Marcha		
Fase de Balanceo		
Caída del Pie	Sí, No	Binario
Inversión del Pie	Sí, No	Binario
Eversión del Pie	Sí, No	Binario
Tobillo Inestable	Sí, No	Binario
Rodilla Extendida	Sí, No	Binario

Capítulo 2: Propuesta de Solución

Hiperflexión de Caderas	Sí, No	Binario
Descripción de Arco	Sí, No	Binario
Fase de Apoyo		
Golpeteo del antepie	Sí, No	Binario
Inversión del Pie	Sí, No	Binario
Eversión del Pie	Sí, No	Binario
Genus Flexus	Sí, No	Binario
Genus Valgo	Sí, No	Binario
Genus Varo	Sí, No	Binario
Genus Recurvatum	Sí, No	Binario
Fase General		
Rodilla Inestable	Sí, No	Binario
Inclinación al lado sano	Sí, No	Binario
Inclinación al lado afectado	Sí, No	Binario
Trendelemburg Positivo	Sí, No	Binario
Rotación Interna	Sí, No	Binario

Capítulo 2: Propuesta de Solución

Rotación Externa	Sí, No	Binario
Retraso MI Afectado	Sí, No	Binario
Retraso Mi Sano	Sí, No	Binario
Mano		
Función de Apertura	Presente, Interferida, Parcial, Ausente	Ordinal
Función de Cierre	Activa, Parcial, Ausente	Ordinal
Tipo de Agarre	Simio, Oposición del Pulgar, Aducción del Pulgar, Pinza Bidigital, Pinza Trípode, Pinza Pentadigital, Gancho	Nominal

Tabla 2.1 Descripción de los rasgos predictores de la base de casos del SBC.

En correspondencia con los rasgos predictores identificados los especialistas seleccionaron un conjunto de ejercicios y tratamientos que permiten recuperar las funciones corporales de pacientes hemipléjicos con alteraciones motoras. La siguiente tabla muestra los tratamientos seleccionados.

Rasgo Objetivo	Posibles Valores
Electro-Estimulación Funcional (EEF)	EEF-Manos, EEF-Hombros, EEF-Pie, EEF-Rodilla
Técnicas de Biofeedback	Sí, No
Mecano-Suspensión Terapia	Sí, No
Bipedestador	Sí, No
Danzo-Terapia	Sí, No
Terapia con cuña y rodillos	Sí, No

Capítulo 2: Propuesta de Solución

Espejo Panorámico	Sí, No
Troller de techo	Sí, No
Estera Rodante	Sí, No

Tabla 2.2 Descripción de los rasgos objetivo de la base de casos del SBC.

2.4 Proceso de extracción del conocimiento

Tal como se describió en el capítulo, el proceso de extracción del conocimiento en los SBC está compuesto por cuatro etapas que deben ser ejecutadas para obtener una solución factible al problema en cuestión, estas etapas son: Recuperación, Reutilización, Adaptación y Almacenamiento o Aprendizaje. A continuación, se explican los algoritmos y estrategias empleadas en cada una de estas etapas:

2.4.1 Etapa de Recuperación

La primera etapa en el proceso de extracción del conocimiento generalmente es antecedida por labores de pre-procesamiento de datos que garantizan completitud y organización. Esto se garantiza en el proceso de creación de la base de conocimientos pues las variables con que se trabaja son de tipo booleanas y Ordinales, su dominio está definido y se valida que no se almacenen valores incorrectos o nulos en la aplicación.

Esta etapa es la encargada de recuperar instancias o casos con características similares al caso analizado, siguiendo una estrategia de umbral para comparar parámetros de los rasgos predictores. El usuario tiene la posibilidad de definir el grado de similitud mínimo para considerar dos casos semejantes. Para ello se emplean funciones de semejanza entre instancias que dependen de la comparación entre cada uno de sus atributos. Los atributos que conforman una instancia están separados en cuatro grupos que representan diferentes tipos de exámenes, estos son: Postura de Pie (PP), Postura Sentado (PS), Marcha (M), Manos (Mn). Para cada una de estas clasificaciones se calcula individualmente el valor de disimilitud entre sus atributos y se realiza la suma de estos. Esta estrategia es similar a la Distancia Euclidiana donde se efectúa una sumatoria de las diferencias entre atributos numéricos, el valor de disimilitud calculado en este caso

Capítulo 2: Propuesta de Solución

es el equivalente a esa distancia. Teniendo en cuenta que todas las variables analizadas son de tipo booleanas, nominales u ordinales las funciones utilizadas para calcular la disimilitud son las siguientes:

Para atributos booleanos se emplea la Función de Hamming (36):

$$d(x,y) = \begin{cases} 0 & \text{si } x = y \\ 1 & \text{CC} \end{cases}$$

Donde (x,y) representan los valores del atributo analizado en las instancias que estén siendo evaluadas y *CC* indica “caso contrario”.

En atributos nominales y/u ordinales se ponderan los valores de disimilitud (36) según criterio de expertos. Estos definen qué tanto se asemejan o no los distintos valores que puede asumir un atributo, quedando como el ejemplo siguiente:

$$\text{Atributo Nivel de Hombros: } d(x,y) = \begin{cases} 0 & \text{si } x = y \\ & \text{si } x = Desc_{Izq}, y = Desc_{Der} \\ & \text{ó} \\ & \text{si } y = Desc_{Izq}, x = Desc_{Der} \\ 0.25 & \\ 1 & \text{en otro caso} \end{cases}$$

Donde (x,y) son los valores del atributo analizado en las instancias que estén siendo evaluadas.

Luego para calcular la disimilitud existente entre cada uno de los criterios en que se agrupan los rasgos predictores (PP, PS, M y Mn) se procede de la siguiente forma. Si la suma obtenida como resultado de la comparación entre atributos es cero, se devuelve ese valor, de lo contrario se divide dicha suma entre la cantidad de atributos evaluados, esto último para garantizar que el resultado quede normalizado (valores entre 0 y 1). La cantidad de atributos evaluados puede variar debido a que la mayoría de los rasgos son booleanos. Si alguna característica no aparece reflejada en ninguna de las dos instancias que están siendo comparadas, entonces este rasgo no se tiene en cuenta pues puede aportar información falsa de similitud o disimilitud entre las instancias. Lo anterior se expresa de la siguiente forma:

Capítulo 2: Propuesta de Solución

$d(C1, C2) = \frac{\sum_{k=0}^n (d(x,y))}{n}$ donde $C1$ y $C2$ representan la característica que está siendo comparada (PP, PS, M o Mn) y n el número de parámetros evaluados.

Entonces la disimilitud entre dos instancias viene dada por:

$d(I1, I2) = \sum_{k=0}^n (d(C1, C2))$ esta función tiene dominio de 0 a 4, 0 en caso de que las instancias no presenten diferencias y 4 en el caso de que no tengan ningún elemento en común.

2.4.2 Etapa de Reutilización

Una vez que se han recuperado de la base de conocimientos los casos más semejantes al nuevo caso analizado, se procede a reutilizar ese conocimiento para generar una propuesta de ejercicios de rehabilitación para el paciente. Este es un problema de clasificación multi-objetivo, puesto que existen varias variables dependientes (varias combinaciones de ejercicios). Para resolverlo se transformó en varios problemas de clasificación para una sola variable dependiente. Básicamente el procedimiento que se sigue es el de clasificar cada ejercicio individualmente. En el conjunto de casos recuperados, se contabiliza la aparición de cada uno de los posibles tratamientos. Los que sobrepasen un valor (definido mediante criterio de expertos) son incluidos en la nueva solución. El conjunto de ejercicios candidatos conforman una propuesta como resultado de esta etapa.

2.4.3 Etapa de Adaptación

La solución obtenida en el paso anterior es analizada y ajustada en esta etapa. Para realizar esta operación se emplea un sistema de reglas que permite revisar los tratamientos propuestos, para identificar elementos que pueden haberse obviado o rasgos incorrectos que se hayan incluido en la solución. Las reglas fueron definidas por los expertos y describen características y procedimientos del negocio que están bien definidos. Incluyen relaciones entre síntomas y tratamientos, dependencia o solapamiento de ejercicios, prohibiciones y contraindicaciones, entre otras. Un ejemplo

Capítulo 2: Propuesta de Solución

de aplicación de las reglas es: “Si el paciente presenta Caída del Pie se debe aplicar Electro Estimulación Funcional”.

2.4.4 Almacenamiento o Aprendizaje

Para garantizar que en un futuro el sistema pueda brindar soluciones a problemas más complejos este debe ir aprendiendo. Esto se traduce en incorporar nuevos casos a su base de conocimientos para poder reutilizarlos más adelante. Esta tarea requiere mayor participación del usuario del sistema, pues debe valorar si el caso que se acaba de analizar es suficientemente interesante como para adicionarlo. El sistema le ofrece la posibilidad de hacerlo, previa verificación de que no existan ya casos demasiado similares en la base de datos, en cuyo caso alertará a usuario. Finalmente, este tiene la última palabra para decidir si continúa con la operación o no.

2.4.5 Módulo de Explicación

Cuando el usuario hace uso de la funcionalidad “Sugerir Tratamiento” y el sistema le brinda una respuesta, mediante el módulo de explicación este puede observar el resultado de cada una de las etapas involucradas en el proceso de extracción del conocimiento. Como resultado de la etapa de recuperación, se muestran las instancias recuperadas, el nivel de semejanza que tienen respecto al nuevo caso y los valores que toman cada uno de sus atributos. En la etapa de reutilización se muestra la propuesta de solución confeccionada a partir de los casos recuperados. Finalmente, en relación con la adaptación se incorpora la propuesta de solución definitiva y si se aplicaron reglas se muestran cuales fueron estas.

2.5 Control, acceso y gestión de la información

Para garantizar el control, acceso y gestión de toda la información que se maneja en el sistema se cuenta con un módulo de gestión para los siguientes fines:

Gestión de usuarios del sistema: Permite gestionar los usuarios que harán uso del sistema, especificando dos roles que pueden ser asignados (Administrador y Usuario). Brinda además la posibilidad de cambiar los datos personales o la cuenta de estos.

Capítulo 2: Propuesta de Solución

Gestión de Pacientes: Permite llevar el control de los pacientes en tratamiento, registrando sus datos personales y clínicos.

Gestión del Historial Clínico del Paciente: Estas funciones están relacionadas con el control y seguimiento que se realiza en la historia clínica de los pacientes. Se permite además de modificar, actualizar y eliminar datos de las historias, mostrar al usuario los datos de dos consultas realizadas a un mismo paciente para que se puedan establecer criterios de comparación sobre su evolución.

Módulo de explicación: A esta funcionalidad se accede cuando se ha hecho uso de los algoritmos del SBC para obtener una propuesta de tratamiento para el paciente. La función de este módulo es ofrecer al usuario la descripción de cómo se llevó a cabo el proceso de confección de la solución en cada una de las etapas que compone al SBC.

2.6 Desarrollo guiado por XP

2.6.1 Fase de Exploración

En esta etapa se confeccionan las historias de usuario que describen las funcionalidades que el cliente desea estén presentes en la aplicación; el equipo de trabajo se familiariza con las herramientas que se emplearán en el desarrollo; y se define el alcance del proyecto.

Historias de Usuario

El método empleado en XP para describir el funcionamiento de la aplicación son las historias de usuario (HU). Estas son el equivalente a los casos de uso empleados en otras metodologías. “Deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia” (25). Las HU correspondientes al SBC no fueron redactadas por el usuario, pero su confección sí fue guiada por este, especificando la prioridad de cada una de ellas. Están representadas mediante tablas con los siguientes elementos:

- **Número:** Número de la HU, incremental.

Capítulo 2: Propuesta de Solución

- **Nombre:** Nombre que identifica la HU.
- **Referencia:** Es el conjunto de HU de las cuales depende la actual.
- **Prioridad:** Esta característica es dada por el cliente con los valores: alta, media o baja en dependencia de la importancia y orden en que desean que sean implementadas.
- **Iteración Asignada:** Número de la iteración en la cual se desarrollará la HU.
- **Puntos Estimados:** Tiempo estimado en semanas que se le asignará.
- **Descripción:** Breve descripción del proceso que define la historia.
- **Observaciones:** Alguna acotación importante a señalar sobre la historia.
- **Prototipo de interfaz:** Prototipo de interfaz, si aplica.

Para asignar los valores de prioridad se tiene en cuenta lo siguiente:

Alta: Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, las que el cliente define como principales para el control del sistema.

Media: Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

Baja: Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura, y no tienen que ver directamente con el sistema en desarrollo.

Se redactaron 9 HU, de ellas 7 se clasificaron con prioridad alta y 2 con prioridad media. A continuación, aparecen representadas 3 de las principales HU especificadas (las demás aparecen en los anexos).

Historia de Usuario	
Número: 4	Nombre de HU: Gestionar Consulta
Referencia: 2,3	Prioridad: Alta

Capítulo 2: Propuesta de Solución

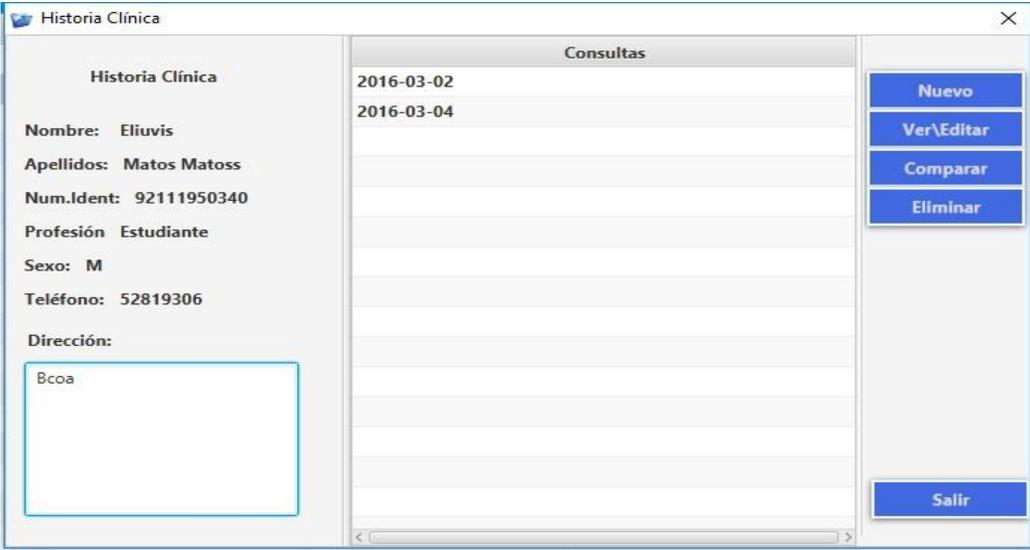
Iteración Asignada: 2	Puntos Estimados: 2
Descripción: El sistema debe permitir registrar, mostrar, modificar y eliminar las consultas del paciente seleccionado.	
Observaciones: La consulta se identifica por la fecha en que se realizó; consta de dos partes: el diagnóstico realizado por el especialista y el tratamiento orientado (sugerido por el propio sistema o definido por el médico).	
Prototipo de Interfaz:	
	

Tabla 2.3 Descripción de la HU 4 Gestionar Consulta.

Historia de Usuario	
Número: 5	Nombre de HU: Gestionar Diagnóstico
Referencia: 2,3,4	Prioridad: Alta
Iteración Asignada: 2	Puntos Estimados: 2
Descripción: El sistema debe permitir registrar, mostrar, modificar y eliminar el diagnóstico realizado en una consulta, validando en cada caso que los datos sean correctos.	

Capítulo 2: Propuesta de Solución

Observaciones: El proceso de diagnóstico cubre varias etapas: datos generales, postura de pie, postura sentado, marcha, miocinética y otros trastornos.

Prototipo de Interfaz:

Tabla 2.4 Descripción de la HU 5 Gestionar Diagnóstico.

Historia de Usuario	
Número: 6	Nombre de HU: Sugerir Tratamiento
Referencia: 5	Prioridad: Alta
Iteración Asignada: 3	Puntos Estimados: 3
Descripción: El sistema debe confeccionar una propuesta de tratamiento, teniendo en cuenta la base de conocimiento de pacientes tratados.	
Observaciones: Se debe implementar un módulo inteligente que emplee técnicas de IA capaces de establecer similitudes entre los casos registrados y el nuevo caso para proponer un tratamiento lo más cercano posible al que daría un médico especialista.	
Prototipo de Interfaz:	

Capítulo 2: Propuesta de Solución

Ver y Editar Diagnóstico

Datos Generales Postura de Pie Postura Sentado Marcha Miocinética Otros Trastornos **Tratamiento**

Analizar casos con semejanza mayor o igual al: 80 %

Sugerir Tratamiento Detalles Tratamiento Guardar Instancia

*(EEF) Electroestimulación Funcional

- EEF Mano
- EEF Pie
- Técnicas de Biofeedback
- Mecano Suspensión Terapia
- Bipedestador
- Danzo-Terapia
- Terapia con Cuña y Rodillos
- Terapia con Aro
- Paralela
- Espejo Panorámico
- Troller de Techo
- Estera Rodante
- Dispositivos Ortésicos
- Dispositivos de Bandaje

Editar

Registrar Cerrar

Tabla 2.5 Descripción de la HU 6 Sugerir Tratamiento.

2.6.2 Fase de Planificación

En la fase de planificación se establece prioridad a las HU más importantes. Los desarrolladores estiman el esfuerzo necesario para cada HU y a partir de esto se define el plan de desarrollo de las iteraciones. Durante la primera iteración debe construirse la arquitectura del sistema.

Estimación del esfuerzo por HU

La estimación del esfuerzo necesario para el desarrollo de la aplicación mediante las HU consiste en la suma del tiempo establecido en cada una de las iteraciones donde se distribuyeron esas HU. La siguiente tabla muestra las HU con los puntos de estimación en semanas.

Historias de Usuario	Puntos de Estimación(semanas)
Gestionar Paciente	2.5
Gestionar Lista de Consultas	1
Gestionar Diagnóstico	2.5

Capítulo 2: Propuesta de Solución

Gestionar Tratamiento	1
Gestionar Base de Conocimiento	3
Sugerir Tratamiento	3
Gestionar Usuario	1
Comparar Consultas	1
Módulo de Explicación	1

Tabla 2.6 Estimación del esfuerzo por HU.

Al realizar la estimación, el tiempo de desarrollo obtenido fue de 16 semanas. Los tiempos estimados de las HU variaron de 1 a 3 semanas.

Plan de Iteraciones

Luego de estimar el tiempo de desarrollo de las HU se procede a realizar el plan de iteraciones. Se seleccionan las HU a desarrollar en cada una de las entregas pactadas. Por cada iteración se definen las pruebas que se realizarán para verificar si se cumplió con lo especificado en la HU correspondiente. A las pruebas fallidas se les realiza un análisis y se corrigen las dificultades para la próxima iteración.

El plan de iteraciones del SBC está distribuido como se describe a continuación:

- **Iteración 1:** En esta iteración se confecciona la arquitectura del SBC; se definen las clases del modelo; se realiza la conexión a la base de datos mediante el *framework hibernate*; y se diseñan las interfaces. Se desarrollan las HU 1,2 y 7 que permiten la gestión de los pacientes y de los usuarios que harán uso de la aplicación. Estas HU son de gran importancia para las futuras iteraciones pues contienen los elementos generales de los que se derivan las demás funcionalidades. Se realizan las pruebas de aceptación, y al finalizar se obtiene una entrega funcional de esta iteración.

Capítulo 2: Propuesta de Solución

- **Iteración 2:** Se desarrollan las HU 3, 4 y 8 correspondientes a la gestión del diagnóstico y tratamiento de los pacientes. En esta etapa se identifican todas las variables que se registran de cada paciente y que son imprescindibles para decidir un diagnóstico. Se definen los elementos necesarios para crear la base de conocimientos que empleará el sistema para sugerir un tratamiento automáticamente. Se implementan también las funcionalidades que le permiten al usuario realizar una comparación visual de los datos de 2 consultas. Para concluir con la iteración se realiza una entrega de la aplicación con las nuevas funcionalidades.
- **Iteración 3:** Se le da solución a las HU 5, 6 y 10 que son las encargadas de operar con la base de conocimientos para realizar una predicción del tratamiento que necesita el paciente; así como de ofrecer una explicación de cómo se arribó a esa solución. Al finalizar se desarrollan pruebas de aceptación y se realiza la entrega final del producto.

El cálculo del tiempo se realizó teniendo en cuenta que una semana de trabajo consta de 5 días con 6 horas dedicadas en cada uno. La siguiente tabla muestra el plan de iteraciones con el tiempo empleado en cada una de ellas y las HU que las componen.

Iteración	Historias de Usuario	Puntos de Estimación
1	Gestionar Paciente	4.5
	Gestionar Lista de Consulta	
	Gestionar Usuario	
2	Gestionar Diagnóstico	4.5
	Gestionar Tratamiento	
	Comparar Consultas	

Capítulo 2: Propuesta de Solución

3	Gestionar Base de Conocimiento	7
	Sugerir Tratamiento	
	Módulo de Explicación	

Tabla 2.7 Plan de Iteraciones.

Plan de Entregas

El plan de entregas se realiza para definir las fechas en que se deberán desplegar las versiones de la aplicación que resultan de cada iteración durante el proceso de desarrollo. Su planificación se basa en la estimación realizada de los tiempos de cada una de las iteraciones. Según el cálculo de tiempo de las iteraciones se confeccionó el siguiente plan de entregas, en el cual se tiene como fecha definitiva el 15 de mayo de 2016.

Historias de Usuario	1ra Iteración 17 de febrero de 2016	2da Iteración 15 de marzo de 2016	3ra Iteración 15 de mayo de 2016
Gestionar Paciente	V-1.0		
Gestionar Lista de Consulta			
Gestionar Usuario			
Gestionar Diagnóstico		V-2.0	
Gestionar Tratamiento			
Comparar Consultas			
Gestionar base de Conocimiento			V-3.0
Sugerir Tratamiento			

Capítulo 2: Propuesta de Solución

Módulo de Explicación			
-----------------------	--	--	--

Tabla 2.8 Plan de Entregas.

2.6.3 Fase de Iteraciones

En esta fase se implementan las funcionalidades del SBC según la iteración que esté siendo desarrollada se realizan las pruebas de aceptación para comprobar si se le dio cumplimiento a lo especificado en la HU. Si las pruebas fallan se procede a re-implementar las funcionalidades deficientes para solucionar el error. Cuando la aplicación de las pruebas es satisfactoria entonces se le entrega al cliente la versión recién implementada para que haga uso de las funcionalidades disponibles y se familiarice con la aplicación. En esta etapa el cliente se mantiene relacionado con el proceso de implementación para esclarecer posibles dudas en la comprensión del negocio.

2.7 Características y Diseño del Sistema

2.7.1 Arquitectura del Sistema

El sistema que se implementa está diseñado mediante una arquitectura Modelo Vista Controlador (MVC), esta permite mantener separada la lógica de la aplicación, los datos que se manejan y los archivos correspondientes a las vistas. “Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.” (37)

- **El Modelo**, contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- **La Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con éste.
- **El Controlador**, actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Capítulo 2: Propuesta de Solución

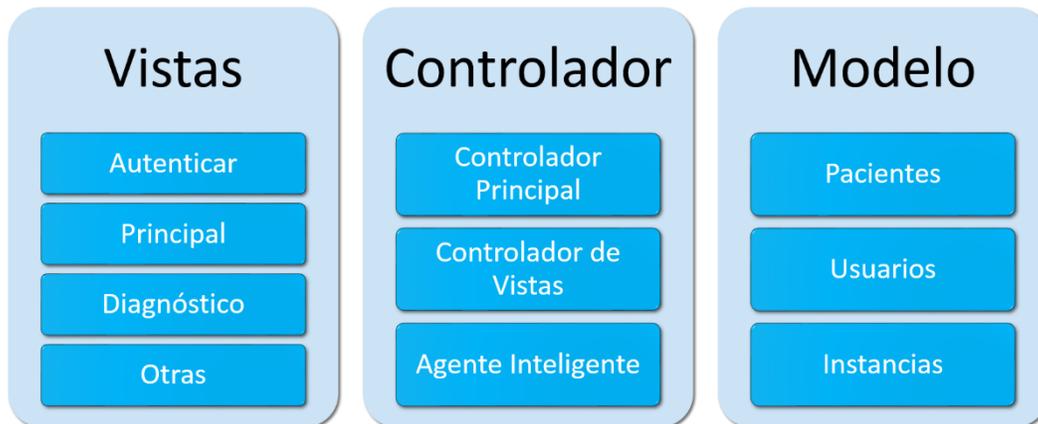


Figura 2.2 Arquitectura MVC del SBC.

2.7.2 Tarjetas Clase-Responsabilidad-Colaboración (CRC)

Las tarjetas CRC se confeccionan para describir el objetivo que cumple una clase determinada dentro de la aplicación, se incluyen además aquellas clases que guardan determinada relación con la que se describe. Estas “permiten desprenderse del método basado en procedimientos y trabajar con una metodología basada en objetos, así el programador se centra y comienza a apreciar el desarrollo orientado a objetos” (38). Estas están compuestas por los siguientes campos:

- **Clase:** Nombre de la clase que se corresponde con la tarjeta.
- **Responsabilidad:** Describe las funcionalidades que contiene clase.
- **Colaboración:** Enuncia las clases que guardan relación con la que se describe en la tarjeta.

A continuación, aparecen tres tarjetas CRC que corresponden a clases de relevancia dentro del sistema:

Tarjeta CRC	
Clase: PrincipalController	
Responsabilidad	Colaboración

Capítulo 2: Propuesta de Solución

<p>Es la clase principal de la aplicación y se encarga de:</p> <ul style="list-style-type: none"> • Controlar las interfaces para la gestión de Usuarios, Pacientes e Historial Clínico del paciente en el sistema. • Crear el Agente inteligente que emplea el sistema para brindar las propuestas de soluciones. • Permitir a los Usuarios visualizar y eliminar instancias. • Permitir al usuario que está trabajando cerrar su sesión o la aplicación. 	<p>Crear_UsuarioController</p> <p>NuevoPacienteController</p> <p>GestionarUsuarioController</p> <p>Lista_DiagnosticoController</p> <p>Agente</p>
--	--

Tabla 3.1: Tarjeta CRC correspondiente a la clase PrincipalController.

Tarjeta CRC	
Clase: DiagnosticoController	
Responsabilidad	Colaboración
<p>Es la clase para la gestión de diagnóstico, se encarga de:</p> <ul style="list-style-type: none"> • Registrar los datos clínicos de los pacientes. • Brindar la entrada al Agente inteligente brindar una propuesta de solución. • Almacenar un diagnóstico como instancia si se considera relevante. • Establecer comparaciones visuales entre elementos del diagnóstico. 	<p>PrincipalController</p> <p>Lista_DiagnosticoController</p> <p>Detalles_Tratamiento</p> <p>Agente</p> <p>Paciente</p> <p>Diagnostico</p> <p>Tratamiento</p>

Tabla 3.2: Tarjeta CRC correspondiente a la clase DiagnosticoController.

2.7.3 Patrones de diseño

Los patrones de diseño son soluciones ya probadas a problemas con características similares que surgen en el proceso de desarrollo de aplicaciones informáticas. Ayudan

Capítulo 2: Propuesta de Solución

al éxito del proyecto, pues permiten la reutilización de código, garantizan la robustez y extensibilidad del sistema. A continuación, se describen los patrones empleados en la implementación del SBC.

Patrones GRASP

Bajo Acoplamiento: Este patrón vela por que no existan dependencias innecesarias entre las clases. Garantiza que cada una pueda realizar sus funciones por separado, relacionándose con las otras clases cuando sea necesario; y que, si en el futuro si se deseara realizar alguna modificación al sistema, se minimizaran las afectaciones a otros componentes. En el sistema que se implementa se ve reflejado en la estructura con que está estructurado el sistema inteligente donde este recibe peticiones únicamente del controlador e interactúa con las clases que contienen información necesaria para su funcionamiento. La tabla 2.9 muestra la tarjeta CRC correspondiente al agente inteligente donde se pone de manifiesto lo descrito anteriormente.

Tarjeta CRC	
Clase: Agente	
Responsabilidad	Colaboración
Es la clase principal del sistema inteligente y se encarga de: <ul style="list-style-type: none">• Proveer la funcionalidad para brindar una solución ante una nueva entrada al sistema.• Contener la lista de instancias con que cuenta el sistema para realizar inferencia.• Controlar el almacenamiento de instancias previendo que no se almacenen instancias sumamente semejantes.	PrincipalController Recuperacion Reutilizacion Adaptacion

Tabla 2.9 Tarjeta CRC de la clase Agente.

Capítulo 2: Propuesta de Solución

Alta Cohesión: Este está muy vinculado al anterior se encarga de que las clases solo realicen funciones estrechamente relacionadas con su contenido, lo que facilita la organización del proyecto y que estas no estén sobrecargadas de funcionalidades ajenas. La tabla 2.9 también ejemplifica el empleo de este patrón.

Controlador: Este patrón se ve reflejado por el modelo arquitectónico MVC utilizado en el desarrollo del SBC. Garantiza que esté centralizado el control de las funcionalidades del software. Esto ofrece una mejor estructuración y organización durante el desarrollo del producto.

Creador: Se emplea al momento de definir la creación de objetos, se garantiza que si una clase puede crear objetos de otra es porque guardan determinada relación, ya sea que una contiene a la otra, o que una contiene los datos que necesita la segunda. Esto se refleja en el momento de crear objetos de clases controladoras de interfaces donde estos solo pueden ser creados por el controlador de la vista que dio origen a la nueva interfaz.

Experto: Se utiliza para garantizar que determinadas clases contengan todos los elementos necesarios para realizar sus labores. Las clases encargadas de realizar las funciones en las que está separado el proceso de extracción del conocimiento presentan estas características (Recuperación, Reutilización y Adaptación).

Patrones GOF

Singleton: Se usa para indicar que de una clase se crea un único objeto, siendo este utilizado por todas las demás clases o funcionalidades que necesiten información contenida en ese objeto. La clase “Agente” representa un ejemplo de cómo se utilizó este patrón en la implementación del SBC. En el sistema se crea una sola entidad de esta clase y es a esta a la que se le efectúan todas las peticiones que involucran a las funcionalidades del agente inteligente.

Capítulo 3: Implementación y pruebas del sistema

Capítulo 3: Implementación y pruebas del sistema

En el presente capítulo se exponen los elementos relacionados con la construcción del sistema, su estructura y la descripción de las clases que intervienen. Se describen también las diferentes pruebas aplicadas y el resultado de estas. Por último, se exponen las técnicas de validación empleadas y sus resultados.

3.1 Implementación

La implementación es la tarea principal que se realiza en la etapa de iteraciones. La metodología XP describe un conjunto de elementos a tener en cuenta a la hora de codificar la lógica de lo que se desea construir. La disponibilidad del cliente es uno de los aspectos fundamentales; la metodología requiere que esté involucrado lo más posible con el equipo pues durante el proceso de desarrollo van a necesitar esclarecer aspectos del negocio que no necesariamente deben estar documentados. Estos son elementos que se discuten de manera informal entre cliente y desarrolladores durante el proceso de desarrollo. Los otros elementos que plantea la metodología como buenas prácticas para el desarrollo son: el uso de codificación estandarizada, programación dirigida por las pruebas (*Test-driven programming*), integraciones permanentes y ritmo sostenido.

3.1.1 Estructura y descripción de la aplicación

La estructura principal de la aplicación se corresponde con un proyecto JavaFX, dentro de esta estructura se organizan las clases y archivos del sistema en determinados paquetes de acuerdo a la función que cumplan dentro de la aplicación, la figura 3.1 muestra cómo está estructurado el proyecto.

Capítulo 3: Implementación y pruebas del sistema

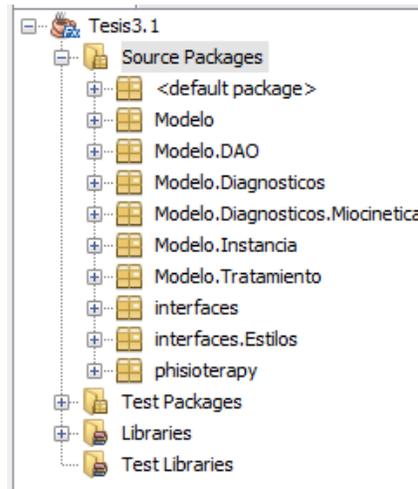


Figura 3.1 Representación de los paquetes que forman parte de la aplicación.

A continuación, se ofrece una descripción del contenido de cada paquete:

1. <default package>: Contiene el archivo de configuración del Framework Hibernate. Este debe aparecer en la raíz del proyecto.
2. Modelo: Contiene las clases que conforman el modelo de la aplicación. Dentro de este paquete aparecen separadas las clases en otros paquetes para facilitar la comprensión del modelo, estos son: Diagnósticos, encargado de contener las clases relacionadas con los datos del diagnóstico de los pacientes; DAO, para almacenar las clases que gestionan la conexión con la base de datos; Instancia y Tratamiento cada uno encargado de contener las clases de estas características.
3. Interfaces: Contiene los archivos FXML que se corresponden con las interfaces visuales del sistema, las clases Java encargadas de controlar estas interfaces y un paquete denominado “Estilos” que agrupa los archivos *Cascade Style Sheet* (css) y las imágenes que se utilizan en el diseño visual del sistema.
4. Physiotherapy: Contiene la clase de arranque del sistema y las clases involucradas en la lógica de la aplicación, el agente inteligente y sus complementos.
5. Test Package: Agrupa las clases que contienen los casos de pruebas unitarias implementadas mediante JUnit.

Capítulo 3: Implementación y pruebas del sistema

3.2 Pruebas

La realización de las pruebas permite evaluar el desempeño de la aplicación. En relación con el aspecto que se desee comprobar existen diferentes tipos de pruebas que pueden ser ejecutadas. XP describe dos tipos de pruebas fundamentales para evaluar la calidad de las aplicaciones. Estas son: las pruebas unitarias, elaboradas por el desarrollador para evaluar automáticamente el desempeño individual de las funcionalidades; y las pruebas de aceptación, que se elaboran en relación con las historias de usuario para validar que el sistema cumpla con las especificaciones realizadas por el cliente. Para la ejecución de las pruebas unitarias se emplea JUnit, un paquete de clases que permiten realizar este tipo de pruebas en Java. Se realizan además pruebas de sistema (carga) para evaluar su comportamiento en distintos escenarios.

3.2.1 Pruebas Unitarias

Se recomienda que estas pruebas sean automatizadas, pues esto facilita al programador identificar funciones que no ofrecen una salida acorde con la lógica que se deseaba implementar. La herramienta JUnit permite correr un conjunto de pruebas de forma automática e informa de aquellas que han fallado. La figura 3.3 representa una de las funcionalidades del sistema a la cual se le realiza una prueba unitaria (figura 3.4).

```
public double CompararNivel(String H1, String H2) {  
    double distancia = 0.0;  
    if ((H1.equals("Nivelados") && !H2.equals("Nivelados"))  
        || (!H1.equals("Nivelados") && H2.equals("Nivelados"))) {  
        distancia += 1;  
    } else if ((H1.equals("Descenzo_Izq") && H2.equals("Descenzo_Der"))  
        || (H1.equals("Descenzo_Der") && H2.equals("Descenzo_Izq"))) {  
        distancia += 0.25;  
    }  
    return distancia;  
}
```

Figura 3.2 Método CompararNivel.

Capítulo 3: Implementación y pruebas del sistema

```
@Test
public void testCompararNivel() {
    System.out.println("CompararNivel");
    String H1 = "Nivelados";
    String H2 = "Descenzo_Izq";
    Recuperar instance = new Recuperar();
    double expectedResult = 1;
    double result = instance.CompararNivel(H1, H2);
    assertEquals(expectedResult, result, 0.0);
}
```

Figura 3.3 Caso de prueba unitaria del método de la Fig 3.3.

Los casos de prueba fallidos fueron corregidos antes del cierre de la iteración. Se tomaron precauciones para evitar errores similares y se definieron nuevos casos de prueba para verificar el correcto desempeño de las funcionalidades. En total se implementaron 13 casos de pruebas unitarias, de ellas 3 aportaron no conformidades y se corrigieron antes de terminar la iteración. Mayormente los casos de prueba están relacionados con las funciones del agente inteligente ya que su correcto funcionamiento es vital en el desarrollo de la aplicación, los resultados aparecen en el siguiente gráfico.

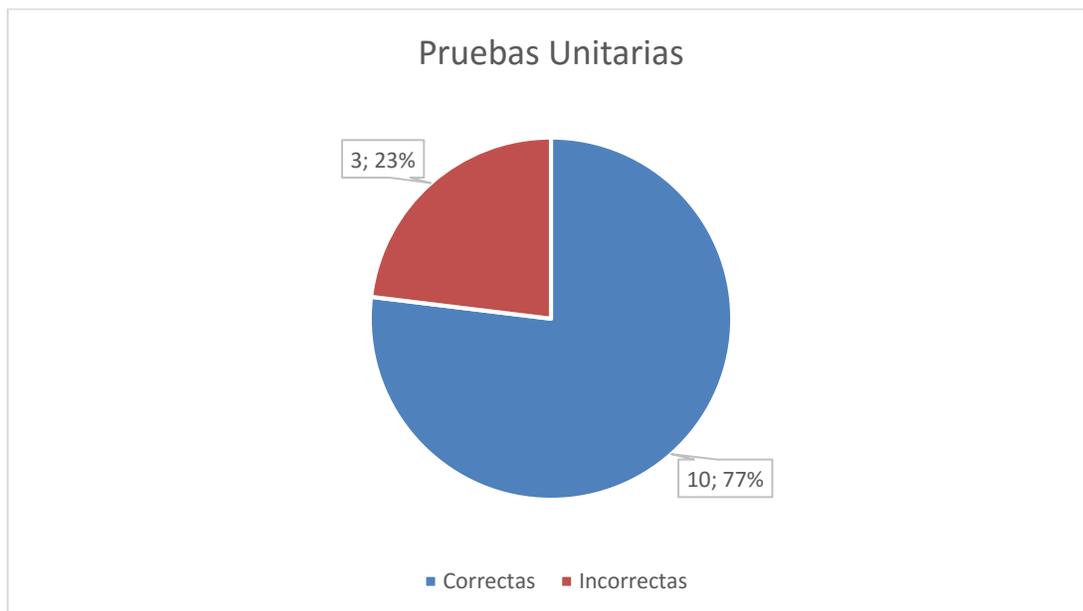


Figura 3.4 Resultados de la aplicación de los casos de pruebas unitarios

Capítulo 3: Implementación y pruebas del sistema

3.2.2 Pruebas Aceptación

Las pruebas de aceptación se elaboran a partir de las HU. Están diseñadas para verificar si el sistema cumple con los requerimientos descritos. Representan la satisfacción del cliente con el producto que se le entrega y es este el principal encargado de verificar su cumplimiento. También conocidas como “pruebas de caja negra”, se centran en probar el sistema funcionalmente, sin tener en cuenta los algoritmos internos. Se ejecuta una acción sobre la aplicación y se espera una respuesta que debe estar en correspondencia con lo esperado, de no ser así la prueba es fallida.

En la confección de estas pruebas se describen los siguientes aspectos:

- **Código:** Identifica la prueba en cuestión, incluye el número de HU, de la prueba y si posee diferentes escenarios.
- **HU:** Número de la HU a la cual pertenece.
- **Nombre:** Nombre del caso de prueba.
- **Descripción:** Acciones que debe realizar el sistema.
- **Condiciones de ejecución:** Describe las características y elementos en general que debe tener el sistema para realizar el caso de prueba.
- **Entrada/Pasos de Ejecución:** Incluye las entradas necesarias del sistema y los pasos necesarios para realizar el caso de prueba.
- **Resultados Esperados:** Respuesta que debe dar el sistema luego de aplicar el caso de prueba.
- **Resultado Obtenido:** Respuesta visual del sistema después de realizar el caso de prueba.
- **Evaluación de la prueba:** Clasificación de la prueba en satisfactoria o insatisfactoria.

A continuación, se muestran algunos de los casos de pruebas aplicados al sistema:

Prueba de Aceptación	
Código: HU1_P1_E1	HU: 1
Nombre: Adicionar Paciente	

Capítulo 3: Implementación y pruebas del sistema

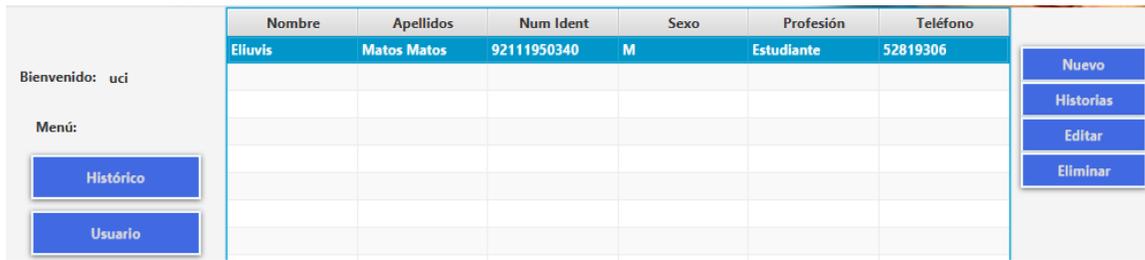
Descripción: El sistema debe permitir registrar un nuevo paciente
Condiciones de Ejecución: Estar autenticado y en la ventana principal de la aplicación
Entrada/Pasos de Ejecución: Nombre: Eliuvis Apellidos: Matos Matos Número Id: 92111950340 Sexo: M Profesión: Estudiante Teléfono: 52819306 Pasos: <ol style="list-style-type: none">1. Hacer clic en el botón Nuevo de la interfaz principal.2. Rellenar los campos con los datos anteriores.3. Hacer clic en la opción Guardar.
Resultado Esperado: La aplicación muestra su interfaz principal con el nuevo usuario registrado.
Resultado Obtenido:  <p>The screenshot shows a web application interface. On the left, there is a sidebar with a 'Bienvenido: uci' message and a 'Menú' section containing 'Histórico' and 'Usuario' buttons. The main area features a table with columns: Nombre, Apellidos, Num Ident, Sexo, Profesión, and Teléfono. The first row contains the data: Eliuvis, Matos Matos, 92111950340, M, Estudiante, 52819306. To the right of the table is a vertical menu with buttons for 'Nuevo', 'Historias', 'Editar', and 'Eliminar'.</p>
Evaluación de la Prueba: Satisfactoria.

Tabla 3.3. HU1_P1_E1 Adicionar paciente

Capítulo 3: Implementación y pruebas del sistema

Código: HU6_P1_E2	HU: 6
Nombre: Sugerir tratamiento con un porcentaje de selección de casos suficientemente alto que no logra recuperar instancias semejantes.	
Descripción: El usuario introduce un caso en el sistema y selecciona un nivel de semejanza muy elevado para el cual no hay instancia parecida en la base de conocimientos.	
Condiciones de Ejecución: Encontrarse en la ventana de diagnóstico y tratamiento y haber completado el formulario de la consulta.	
Entrada/Pasos de Ejecución: Datos clínicos del paciente Pasos: 1. Introducir los datos clínicos del paciente. 2. Introducir un valor de semejanza suficientemente alto. 3. Hacer clic en el botón Sugerir Tratamiento.	
Resultado Esperado: La aplicación muestra al usuario una notificación con el texto "No existen casos similares".	
Resultado Obtenido:  <p>The screenshot shows a software window titled "Ver y Editar Diagnóstico" with a close button (X) in the top right corner. Below the title bar is a navigation menu with tabs: "Datos Generales", "Postura de Pie", "Postura Sentado", "Marcha", "Miocinéctica", "Otros Trastornos", and "Tratamiento". The "Tratamiento" tab is active. A central notification box displays "No Existen Casos Similares". Below this, there is a field "Analizar casos con semejanza mayor o igual al:" with a value of "100" and a percentage sign. Three buttons are visible: "Sugerir Tratamiento" (highlighted in blue), "Detalles Tratamiento" (greyed out), and "Guardar Instancia" (blue). Underneath, there is a section for "*(EEF) Electroestimulación Funcional" with a grid of checkboxes for various treatments. The checked options are "Troller de Techo" and "Estera Rodante". At the bottom right, there are "Registrar" and "Cerrar" buttons.</p>	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3.4. HU6_P1_E2 Sugerir Tratamiento con umbral de selección demasiado alto.

Capítulo 3: Implementación y pruebas del sistema

Por cada iteración se realizaron los casos de prueba correspondientes a las HU desarrolladas en la misma. Se asume que la iteración ha concluido satisfactoriamente si el 100% de los casos de prueba resultan satisfactorios, los casos que no resulten satisfactorios regresan a las etapas de análisis e implementación para solucionarlos y realizarles nuevamente las pruebas.

El siguiente gráfico muestra el resultado de la aplicación de los casos de prueba en cada una de las iteraciones realizadas, para todos los casos de prueba fallidos se dio solución a las funcionalidades que se identificaron como deficientes.

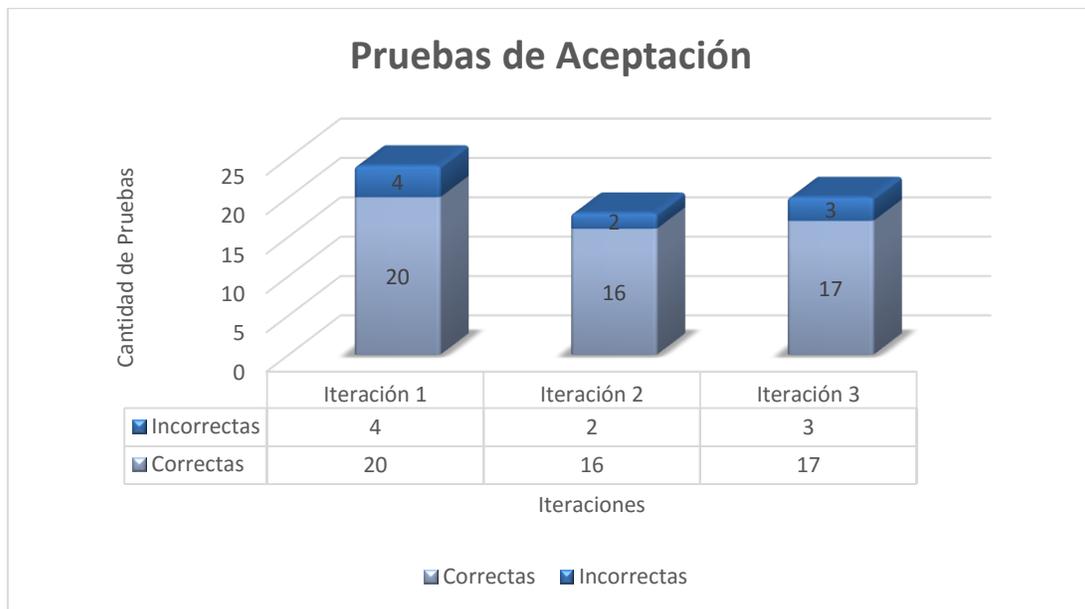


Figura 3.5 Resultados de la aplicación de los casos de pruebas de aceptación.

3.2.3 Pruebas de Rendimiento

Este tipo de pruebas se realiza para conocer cómo el desempeño del sistema en distintos escenarios de uso. También para identificar los cuellos de botella o el momento en que esta dejará de responder a las peticiones de los usuarios.

En el sistema implementado existen un número de transacciones o funciones de las cuales es necesario conocer su comportamiento en casos de carga extrema. Estas funciones son las referidas a la recuperación de los datos almacenados en la base de datos y la utilización de estos en procesos como el de asignar tratamiento. Para probar

Capítulo 3: Implementación y pruebas del sistema

estas funcionalidades se empleó una base de datos con 200 casos generados de forma automática. A esta Base de Datos se le realizaron distintas peticiones y se registraron los tiempos referidos a la demora en que el sistema arranca y recupera la información de la Base de Datos (A), la función de recuperar instancias semejantes (B) y la de asignar tratamiento (C), siendo estos los procesos más críticos en cuanto a rendimiento. El siguiente gráfico muestra cómo se comportaron estos 3 procesos (A, B, C) con diferentes tamaños de datos.

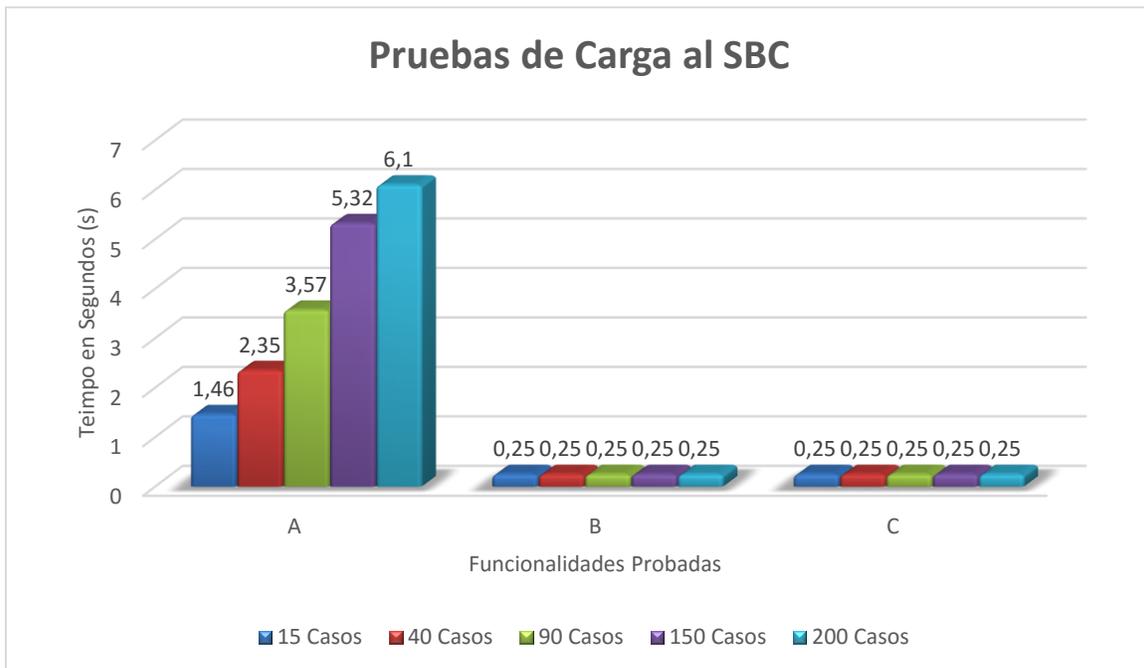


Figura 3.6 Resultados de la aplicación de los casos de pruebas carga.

La gráfica anterior muestra una estabilidad en los procesos (B) y (C) con tiempos de respuestas por debajo de un segundo, no obstante, el proceso (A) aumenta el tiempo de respuesta en dependencia de la cantidad de datos que se manejan. Lo anterior se debe a la estrategia diseñada para recopilar información de la Base de Datos. Cuando la aplicación es ejecutada, se cargan todos los datos, tanto de los pacientes como las instancias del agente inteligente, los procesos (B) y (C) utilizan esos datos ya cargados en tiempo de ejecución, por ello el tiempo de respuesta es rápido (menos de un segundo). La otra estrategia que pudiera implementarse es recuperar los datos en el momento en que serán utilizados, esto estabilizaría el proceso (A) pero provocaría un alza en el (B) y

Capítulo 3: Implementación y pruebas del sistema

(C). Esta última estrategia es recomendable si se trabaja con grandes cantidades de datos (más de mil entradas en la Base de Datos). Con esta cantidad de datos (menos de mil) el consumo de recursos del sistema no tiene grandes variaciones. Se consideró más eficaz que el proceso de arranque del sistema fuera el que más se demorara puesto que una vez que está en ejecución todas las prestaciones responden con tiempos inferiores a un segundo.

La siguiente gráfica muestra el consumo de memoria RAM de la aplicación con diferentes volúmenes de datos.

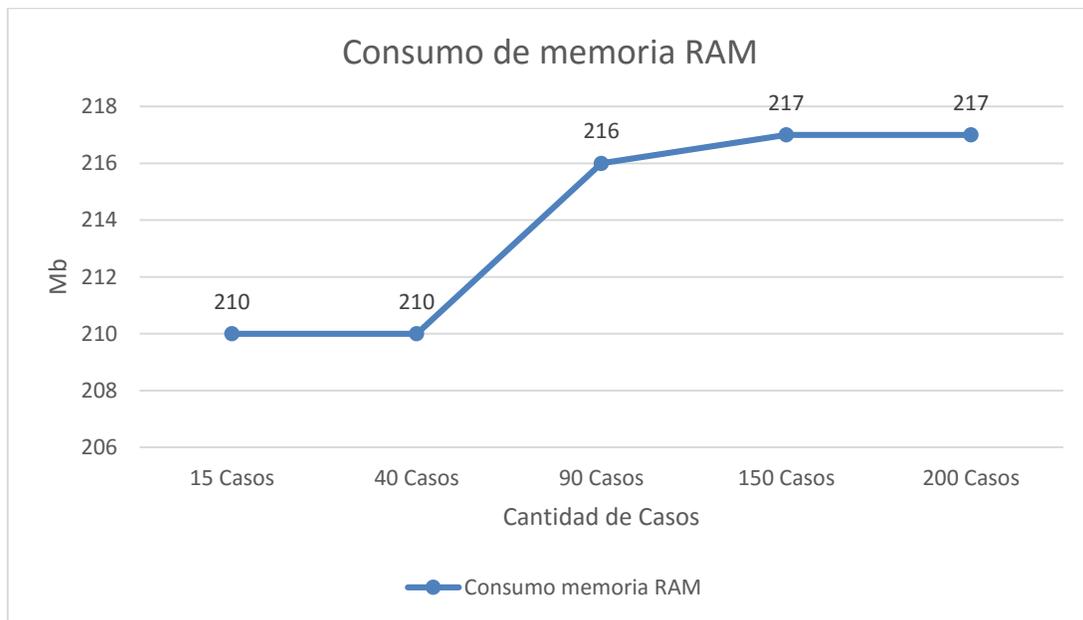


Figura 3.6 Consumo de memoria RAM de la aplicación.

3.3 Validación

Para realizar la validación del sistema se conformó una base de conocimientos con 40 casos de pacientes previamente tratados por los especialistas y con estos se les dio tratamiento a otros 20 pacientes. Se utilizó como criterio de recuperación los casos con 70% o más de semejanza debido a que el número de instancias no es abundante y un porcentaje más alto en ocasiones no arroja ningún resultado. Se sometieron las soluciones a evaluación de los expertos y se determinó que 14 de las 20 respuestas fueron totalmente correctas para un 70% de efectividad. Las restantes 6 soluciones fueron

Capítulo 3: Implementación y pruebas del sistema

modificadas por los especialistas, aunque no resultaron ser soluciones incorrectas sino incompletas, teniendo en cuenta el examen previo que se había realizado al paciente. Las respuestas identificadas como incompletas presentan la característica de ser aquellas cuyo diagnóstico contiene pocos elementos por lo que para lograr una solución correcta debe establecerse un criterio de selección mayor al 70%. El siguiente gráfico muestra los resultados de este proceso:

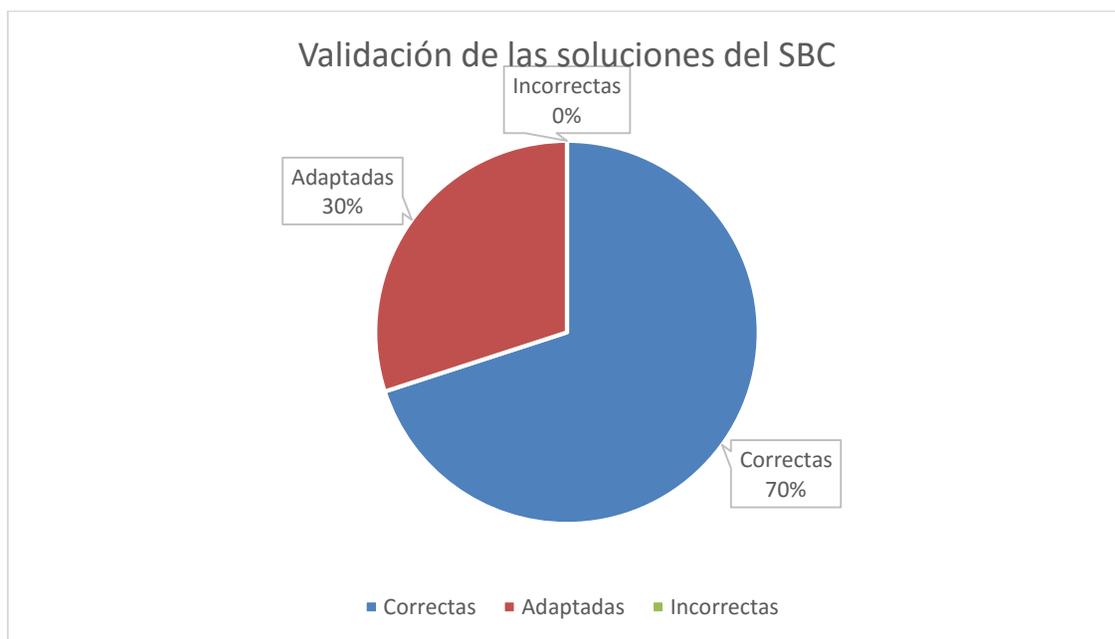


Figura 3.7 Resultados de la evaluación de las respuestas del SBC.

Conclusiones

Conclusiones

- Se comprobó la viabilidad de los SBC para proponer soluciones válidas al problema de recomendar terapias para pacientes hemipléjicos con alteraciones motoras.
- La comparación entre casos compuestos en su mayoría por atributos booleanos mediante la distancia de Hamming puede verse alterada si no se excluyen las características que no aportan información relevante a ninguno de los casos que estén siendo evaluados.

Recomendaciones

Recomendaciones

- Enriquecer la base de conocimientos del sistema mediante la incorporación de nuevos casos de pacientes tratados.
- Aplicar técnicas de Ingeniería del Conocimiento en otras áreas de la atención a pacientes hemipléjicos.
- Extender el Sistema Basado en el Conocimiento a otros procesos de diagnóstico y tratamiento de pacientes hemipléjicos.

Bibliografía

Referencias Bibliográficas

1. Ayala, Alejandro Peña. *Sistemas basados en Conocimiento: Una Base para su Concepción y Desarrollo*. México : s.n., 2006.
2. Piñero, Bismarck Martín. SciELO Cuba. [En línea] sept de 2013.
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1560-43812013000300009.
3. Cuadrado, Ángel Áreas. *Rehabilitación del ACV: evaluación, pronóstico y tratamiento*. Galicia : s.n., 2009.
4. Bases Psicológicas de la Educación Especial. [En línea] ene de 2012.
<http://www.ugr.es/~iramirez/Bases%20Psicol/Dismot/Concepto.htm>.
5. Diccionario Médico. [En línea] 1 de ene de 2008. [Citado el: 5 de feb de 2016.]
<https://diccionario.medciclopedia.com/d/deficit-sensitivo/>.
6. Salud. [En línea] 10 de oct de 2012. [Citado el: 25 de mar de 2016.]
http://salud.discapnet.es/Castellano/Salud/educacion/manejo_enfermedad/Pagina_s/lctus.aspx.
7. Hemiplejia. [En línea] 2013. <http://hemiplejia.org/tratamiento/>.
8. Romero, Juan Jesús. *Inteligencia Artificial y Computación Avanzada*. 2007.
9. Badaró, Sebastián. *Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones*. 2013.
10. Munakata, Toshinori. *Fundamentals of de new Artificial Intelligence*. Cleveland : s.n., 2008.
11. Lio, Daniel Gálvez. *Curso de Sistemas Basados en el Conocimiento*. Las Villas : s.n., 1998.
13. Pérez, Ileana Martí. *Sistema para la ayuda a la toma de desiciones y tratamiento de Disliproteinemias Basado en la Ingeniería del Conocimiento*. La Habana : s.n., 2012.
14. Rodríguez, Marianela Gutiérrez. *Sistema Experto para el diagnóstico médico de las enfermedades genéticas con dismorfias (SEGEDIS)*. La Habana : s.n., 2010.
15. Riley, Giarratano. *Sistemas Expertos: Principios y Programación*. España : International Thomson Editores, 2001.

Bibliografía

16. Prendes, Frank David Corona. *Sistema Basado en Casos para predecir la ocurrencia de reacciones adversas a medicamentos en la consulta médica*. La Habana : s.n., 2013.
17. *Aplicaciones médicas como ayuda al diagnóstico en la medicina. Experiencia SOFTEL-MINSAP*. Hernández, Mirna Cabrera. La Habana : s.n., 2012.
18. Diagnos. [En línea] [Citado el: 20 de oct de 2015.]
http://www.diagnosmd.com/que_es.php.
19. Pascual, José Carlos Pulido. *Planificación Jerárquica y Detección de Poses para el Desarrollo de Terapias de Rehabilitación con Robots Sociales*. Madrid : s.n., 2014.
20. Lazo, Niusbel Hernández. *Clasificación Penitenciaria apoyado en Razonamiento Basado en Casos (RBC)*. La Habana : s.n., 2013.
21. Canós, José H. *Metodologías Ágiles en el Desarrollo de Software*. Valencia : s.n.
22. Sánchez, Tamara Rodríguez. *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2015.
23. Sanchez, María A. Mendoza. *Metodologías De Desarrollo De Software*. Perú : s.n., 2004.
24. Figueroa, Roberth G. *Metodologías Tradicionales VS. Metodologías Ágiles*. 2010.
25. Joskowicz, Ing. José. *Reglas y Prácticas en eXtreme Programming*. España : s.n., 2008.
26. Ramírez, Ing. Danay Pérez. *Metodologías Ágiles ¿Cómo desarrollo usando XP?* La Habana : s.n., 2008.
27. Villegas, Adrian Anaya. Monografías.com. [En línea] 2009.
<http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema2.shtml#top>.
28. Gosling, James. *The Java™ Language*. 2011.
29. Oracle. Oracle Help Center. [En línea] ene de 2016.
<http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784>.

Bibliografía

30. —. Oracle Help Center. [En línea] ene de 2016. <http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html>.
31. NetBeans. [En línea] ene de 2016. https://netbeans.org/index_es.html.
32. Galiano, Fernando Berzal. *"Apuntes de programación orientada a objetos en Java: Fundamentos de programación y principios de diseño"*. 2010.
33. HIBERNATE. [En línea] feb de 2015. <http://hibernate.org/orm/>.
34. Martínez, Rafael. PostgreSQL-es. [En línea] 2013. http://www.postgresql.org.es/sobre_postgresql.
35. Hernández, Julio Zamarreño. *Estimulación eléctrica funcional en la flexión dorsal del pie en pacientes hemipléjicos*. La Habana : s.n., 2008.
36. López, Beatriz. *Case-Based Reasoning a Concise Introduction*. 2013.
37. Botella, Andrés Vallés. *ASP.NET MVC*. España : s.n., 2012.
38. Beck, Kent. *Extreme Programming Explained*. 2004.
39. Hemiplejia. [En línea] 2013. <http://hemiplejia.org/causas/>.
40. Escribano, Gerardo Fernández. *Introducción a Extreme Programming*. 2002.
41. Marin, Carmelo. Tutor de Programación. [En línea] 2015. [Citado el: 04 de feb de 2016.] <http://acodigo.blogspot.com/2014/10/javafx-8-administrar-ventanas.html>.