



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7

**Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas**

Sistema Inteligente para el Diagnóstico de Hipertensión Arterial

Autor: Osvaldo Cáceres Prieto

Tutor: Ing. Yovannys Sánchez Corales

La Habana, junio 2011

“Año 53 de la Revolución”

“Educar es hacer prevalecer en la especie humana la conciencia por encima de los instintos.”

Fidel Castro

DATOS DE CONTACTO

Nombre: Ing. Yovannys Sánchez Corales: Profesor asistente graduado en el año 2005 de Ingeniero en Informática en la CUJAE. Pertenece al Centro de Desarrollo de Software para la Salud. Ha impartido las asignaturas de Inteligencia Artificial, Programación III y Práctica Profesional. Forma parte del proyecto de Atención Primaria de la Salud.

Correo electrónico: yscorales@uci.cu

A mi mamá

A mi familia en general

A la Universidad

A mis amigos

AGRADECIMIENTOS

A mi madre:

Por estar siempre ahí cuando la necesité, por ayudar a levantarme de los tropiezos de la vida, por estar siempre pendiente de mí, por quererme tal como soy.

A mi padre:

Por ser siempre mi ejemplo a seguir, por guiarme por el mejor camino, por darme el apoyo necesario a lo largo de la carrera.

A mis hermanas:

Por estar pendientes de mí durante esta etapa, por estar siempre presentes en las buenas y en las malas brindándome todo su apoyo.

A mis abuelos:

Por ayudarme tanto durante estos cinco años, por estar siempre pendientes de mí, por su amor sin medida, y por confiar siempre en mí.

A mis tías:

Por ayudarme en todo lo que necesité, por estar pendientes de mí en todo momento y complacer muchos de mis caprichos.

A toda mi familia:

Por estar al tanto de mí en todo momento y brindarme su apoyo en el momento que lo necesité.

A Albita y su familia:

Por apoyarme siempre y estar pendiente de mí.

A mi tutor:

Por su apoyo incondicional en todo el desarrollo de la tesis, por ser más que eso, por ser un buen amigo.

Al tribunal y oponente:

Por exigirme tanto para poder llegar a lograr un trabajo con la mayor calidad y por ayudar a hacer realidad mi sueño de graduarme con resultados satisfactorios.

A Zulema y Menéndez:

Por malcriarme en todos estos años y ayudarme siempre que lo necesité.

A Odaisa y Yosleivis:

Por estar siempre a mi lado en estos años, por ayudarme en lo que necesité siempre

A Milena:

Por su ayuda incondicional durante estos meses para poder alcanzar los mejores resultados, por aguantar todas mis malcriadeces, por toda su comprensión y dedicación

A Anny:

Por haberme ayudado en este último año para obtener buenos resultados.

A mis amistades:

Por estar siempre a mi lado compartiendo momentos de alegría y tristeza, en fin por estar ahí cuando los necesité.

Al colectivo de trabajadores y directivos de APS de la Facultad 7 por haberme formado como ingeniero.

Quisiera agradecer a todas aquellas personas que de una forma u otra han tenido que ver con mi desarrollo como persona; a todas las amistades que hice en esta universidad, a mis amistades de Pinar, a mis profesores de la vocacional; sepan que lo que soy hoy como ser humano se debe en gran medida al aporte que hizo cada uno de ustedes en mi vida.

RESUMEN

Uno de los problemas de salud más comunes en la Atención Primaria de Salud es la Hipertensión Arterial. Esta enfermedad afecta a un elevado número de personas a nivel mundial y en Cuba. Los países en vías de desarrollo son los más afectados pues su prevalencia ha aumentado progresivamente en los últimos años. La Hipertensión Arterial constituye un importante factor de riesgo cardiovascular y renal, identificándose como uno de los principales factores de riesgo de mortalidad y la tercera causa de incapacidad de la población general.

Teniendo en cuenta esto, el presente trabajo de diploma pretende ayudar al diagnóstico de dicha enfermedad en la Atención Primaria de la Salud. Para ello se plantea la construcción de un Sistema Inteligente para el Diagnóstico de Hipertensión Arterial. Se utilizará la metodología “Modelo de Dominio” haciendo uso de la plataforma de desarrollo Java Enterprise Edition. Se usa Java como lenguaje de programación y se implementa el patrón de arquitectura Modelo Vista Controlador. Como gestor de bases de datos se usará PostgreSQL, Hibernate como herramienta ORM para la persistencia de los datos y el framework Seam para la lógica del negocio.

Aporta como beneficio la posibilidad de contar con información real almacenada, además permite autoabastecerse con nuevos datos surgidos en la marcha, facilitando así la toma de decisiones estratégicas y tácticas para los especialistas.

Palabras Claves: sistema inteligente, diagnóstico, hipertensión arterial.

ÍNDICE

INTRODUCCIÓN.....	1
FUNDAMENTACIÓN TEÓRICA.....	5
1.3 <i>Informatización del Sistema Nacional de Salud Cubano.....</i>	5
1.4 <i>Medicina Interna e Hipertensión Arterial.....</i>	6
1.3 <i>Sistemas Expertos.....</i>	7
1.3.1 Estructura de los Sistemas Expertos.....	8
1.3.2 Elementos comunes de los Sistemas Expertos.....	10
1.3.3 Ventajas de los Sistemas Expertos.....	11
1.3.4 Limitaciones actuales de los Sistemas Expertos.....	12
1.3.5 Tipos de Sistemas Expertos.....	12
1.4 <i>Razonamiento Basado en Casos.....</i>	13
1.4.1 Ciclo del razonamiento basado en casos.....	14
1.4.2 Ventajas del uso de razonamiento basado en casos.....	16
1.4.3 Inconvenientes de uso de razonamiento basado en casos.....	17
1.4.4 Funciones de Semejanza.....	17
1.5 <i>Análisis de soluciones existentes.....</i>	19
Ámbito Nacional.....	19
Sistema Experto para el Diagnóstico y Tratamiento de Infecciones de Transmisión Sexual (DITRITS).....	19
Sistema Experto para el Diagnóstico de Anomalías Craneofaciales (DIAG).....	20
HyperWeb.....	21
Ámbito Internacional.....	22
Sistema Experto para el Diagnóstico de Enfermedades Infecciosas en la Sangre (MYCIN).....	22
Sistema Experto para el Diagnóstico en Medicina Interna (INTERNIST/CADUCEUS).....	24
DiagnosMD.....	24
Casey.....	26
1.6 <i>Tendencias y tecnologías actuales a considerar.....</i>	27
1.6.1 Sistemas Distribuidos. Modelo Cliente-Servidor.....	27
1.6.2 Patrón Arquitectónico.....	27
1.6.3 Marcos de desarrollo, bibliotecas y componentes.....	28
Java Server Faces (JSF).....	28
RichFaces 3.3.1.....	29
Ajax4JSF.....	29
Facelets.....	29
Jboss Seam 4.2.2.....	30
Hibernate 3.3.....	30
1.6.4 Lenguaje de Programación.....	30
1.6.5 Servidor de Aplicaciones.....	30
1.6.6 Sistema Gestor de Base de Datos.....	31
1.6.7 Metodologías de Desarrollo de Software.....	31
1.6.8 Lenguaje Unificado de Modelado (UML).....	33
1.6.9 Herramientas.....	33
CARACTERÍSTICAS DEL SISTEMA.....	34
2.1 <i>Conceptos Fundamentales.....</i>	34
2.1.1 Modelo de Dominio.....	35
2.2 <i>Propuesta del Sistema.....</i>	36
2.3 <i>Especificación de los requerimientos de software.....</i>	37

2.3.1	Requerimientos Funcionales	37
2.3.2	Requerimientos no Funcionales.....	38
DISEÑO DEL SISTEMA		41
3.1	<i>Descripción de la arquitectura.....</i>	<i>41</i>
3.1.1	Fundamentación del uso de patrones	41
	Patrón arquitectónico	41
	Patrones de diseño.....	43
3.2	<i>Modelo de diseño.....</i>	<i>44</i>
3.2.1	Definición de elementos del diseño.....	44
3.2.2	Diagramas de clases del diseño	45
3.2.3	Descripción de las clases.....	46
IMPLEMENTACIÓN Y PRUEBA.....		48
4.1	<i>Implementación.....</i>	<i>48</i>
4.1.1	Diagrama de despliegue	48
4.1.2	Tratamiento de errores.....	49
	Seguridad.....	49
4.1.3	Estrategias de codificación. Estándares y estilos a utilizar.....	50
4.2	<i>Pruebas de Software.....</i>	<i>51</i>
4.2.1	Niveles de Prueba.....	52
4.3	<i>Métodos de Prueba</i>	<i>53</i>
4.3.1	Pruebas de Caja Negra.....	53
4.3.2	Diseño de Casos de Prueba.....	53
4.4	<i>Resultados Obtenidos.....</i>	<i>53</i>
CONCLUSIONES.....		54
ANEXOS		64
	ANEXO #1: TABLA DE CRITERIOS DE EXPERTOS PARA VALIDAR LOS DATOS DE LA BASE DE DATOS.	64
	ANEXO #2: DIAGRAMAS DE CLASES DEL DISEÑO	65
	Descripción del diagrama de clases del diseño Adaptar Casos.	65
	Descripción del diagrama de clases del diseño Recuperar Casos.	66
	Descripción del diagrama de clases del diseño Índice de Masa Corporal.	68
	Descripción del diagrama de clases del diseño Clasificar Casos.....	69
	ANEXO #3: TABLA DE CASOS QUE INICIALIZAN LA BASE DE DATOS PARA EL DIAGNÓSTICO DE LA HIPERTENSIÓN ARTERIAL...69	

INTRODUCCIÓN

En la actualidad la toma de decisiones objetivamente fundamentada, se ha convertido en una tarea fundamental en la dirección de diferentes procesos, con énfasis en lo empresarial, sin embargo, las decisiones no son sólo tareas relativas a la dirección. Cualquier hombre ha tomado todo tipo de decisiones a lo largo de su vida. [1]

La base del proceso de toma de decisiones es la información que se tiene del dominio de aplicación. Mientras mayor y mejor sea la información, mayor calidad existirá en la definición del problema, en las propuestas de solución, en el análisis de variantes y en la selección de la acción más conveniente. [2]

La mayoría de las empresas generan información de gran valor como producto de las actividades que realizan diariamente, la cual es almacenada y manipulada a través de sistemas de bases de datos operacionales. Muchas veces, cuando el volumen de los datos crece, estos sistemas se tornan lentos e ineficientes, provocando demoras en la obtención de resultados y la pérdida de información al no contar con la capacidad lógica necesaria para analizar todos los indicadores. Es por esto que, para perfeccionar la toma de decisiones objetivamente fundamentada, surge la inteligencia de negocio, cuyo objetivo fundamental es analizar eventos que ya ocurrieron o que están sucediendo y predecir eventos futuros.

Uno de los objetivos actuales del gobierno cubano es la informatización y perfeccionamiento de los sectores fundamentales de la sociedad. El sector de la salud en Cuba es uno de los más grandes y complejos si de gestionar y actualizar información se trata. El proceso de informatización del mismo se comenzó desde hace varios años, con la ayuda de las empresas productoras de software y universidades cubanas. La Universidad de las Ciencias Informáticas (UCI) es uno de los centros educacionales cubanos que tiene dentro de sus funciones principales el desarrollo de software que, entre otros objetivos contribuyen a la informatización de la sociedad.

El Sistema Nacional de Salud (SNS) expresa su estructura dividida en los siguientes niveles de atención:

- ❖ **Atención Primaria de Salud (APS):** Es la piedra angular del SNS de Cuba, el primer contacto que realiza la población con la atención médica. Enmarca una serie de acciones de diagnóstico,

prevención, curación y rehabilitación, que deben realizarse desde un Nivel Primario en beneficio de la comunidad.

- ❖ **Atención Secundaria de Salud (ASS):** Es el segundo escalón asistencial, comprende los servicios de los hospitales a la comunidad y está dirigido a los pacientes con problemas que no son solucionados en el Nivel Primario.
- ❖ **Atención Terciaria de Salud (ATS):** Comprende los servicios médicos integrales y de alta calidad a enfermedades o padecimientos que requieren hospitalización con atención especializada y de alta tecnología. Pertenecen a este nivel los Hospitales Especializados e Institutos de Investigaciones.

Dentro de las enfermedades más comunes que atiende la APS se encuentra la Hipertensión Arterial (HTA); de aquí que este nivel sea el eslabón básico para el tratamiento de la misma. Como factor de riesgo cardiovascular, la HTA es potencialmente “modificable”, y todas las intervenciones para su control son importantes. La magnitud del problema de la HTA y su impacto sobre la morbimortalidad cardiovascular, determinan la necesidad de mejorar y actualizar el conocimiento de la misma, su tratamiento y control. [3]

En aras de buscar una respuesta a las dificultades que frenan el diagnóstico eficiente de la HTA en la APS, los esfuerzos estarán encaminados a resolver la siguiente **situación problemática**:

En ocasiones, los especialistas médicos cometen errores en el diagnóstico de la enfermedad debido a que esta se presenta de forma asintomática. En estas situaciones, normalmente los especialistas recurren a la búsqueda en su memoria del conocimiento adquirido por la experiencia, además, solicitan los criterios de especialistas con mayor experiencia. Esto ocasiona que se omita información que podría facilitar el proceso de diagnóstico de la enfermedad cuando el especialista se encuentra ante un caso donde los síntomas no son evidentes.

Además, los expertos médicos no pueden estar disponibles de forma ininterrumpida. Así pues, a pesar de que ofrezcan su máximo desempeño durante la jornada laboral es imposible obtener una disponibilidad permanente de su conocimiento.

Por otra parte, los médicos especialistas no tienen un tiempo de vida de servicio ilimitado por lo cual la preservación del conocimiento que ellos almacenan depende de un constante proceso de enseñanza y aprendizaje.

Teniendo en cuenta los elementos mencionados anteriormente se plantea como **problema a resolver**: ¿Cómo facilitar el diagnóstico de la hipertensión arterial en la Atención Primaria de Salud?

Para el desarrollo de la investigación se define como **objeto de estudio** el proceso de descubrimiento de información en la Atención Primaria de Salud.

Se define como **campo de acción** el proceso de descubrimiento de la información para el diagnóstico de la hipertensión arterial en la Atención Primaria de Salud.

Para dar solución al problema planteado se define como **objetivo general**: Implementar un sistema inteligente basado en la experiencia que tribute al diagnóstico de hipertensión arterial en la Atención Primaria de Salud.

Para lograr el objetivo propuesto se dará cumplimiento a las siguientes tareas:

- ❖ Identificar las características y síntomas de las enfermedades en edades avanzadas pertenecientes a la Atención Primaria de Salud.
- ❖ Aplicar la arquitectura definida por el Centro de Salud para el desarrollo de sus aplicaciones.
- ❖ Realizar pruebas al sistema para comprobar la veracidad del mismo.
- ❖ Implementar las funcionalidades identificadas en el ciclo de vida del Razonamiento Basado en Casos:
 - Recuperación de Casos.
 - Almacenamiento de Casos.
 - Crítica y Justificación de Casos Elegidos.
 - Adaptación de Casos.
- ❖ Confeccionar la base de casos que contenga el conocimiento inicial para el diagnóstico de la HTA.

Se plantea como **idea a defender** que: el desarrollo de esta aplicación permitirá acelerar el proceso de análisis de la información y toma de decisiones por parte de los especialistas, permitiendo así a los especialistas un mejor diagnóstico de los problemas de salud en los pacientes. Además, contribuir al aprendizaje de los médicos y especialistas basándose en la información generada por el sistema.

El documento está estructurado por cuatro capítulos:

CAPÍTULO 1. Fundamentación Teórica: Contiene los aspectos esenciales para entender el entorno del problema a resolver. Se describen los conceptos fundamentales asociados al dominio del problema además de ubicar al lector en el ambiente de desarrollo, justificándose el uso de tecnologías, metodologías y herramientas para el desarrollo de la solución propuesta

CAPÍTULO 2. Características del Sistema: Presenta los argumentos principales del dominio de aplicación en el que se enmarca el problema a resolver. En el mismo, se plasma el marco conceptual asociado a la información que será manipulada por el sistema, llegándose a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

CAPÍTULO 3. Diseño del Sistema: Muestra los elementos básicos del diseño del sistema, mediante la justificación del uso de patrones y los diagramas de clases del diseño. Se centra en la modelación detallada y la construcción de la estructura de la aplicación.

CAPÍTULO 4. Implementación y prueba: Describe los componentes fundamentales de todo el proceso de implementación, argumentando la justificación de la integración con otros módulos, estándares de diseño, codificación y excepciones, así como la representación general del diagrama de despliegue.

FUNDAMENTACIÓN TEÓRICA

El objetivo fundamental de este capítulo es abordar los aspectos más relevantes que conforman el marco teórico de la investigación y que ayudarán al diseño y la implementación de sistema. En el mismo se describen los conceptos principales para comprender el basamento de la investigación y se estudian algunos de los sistemas existentes a nivel mundial y nacional vinculados al problema y que pueden servir como referencia para la elaboración de esta solución. Además se realiza un análisis de las técnicas, tecnologías, metodologías y herramientas de software sobre las cuales se realiza el proceso de desarrollo.

1.3 Informatización del Sistema Nacional de Salud Cubano.

En los últimos años varias instituciones, entre ellas la Universidad de Ciencias Informáticas (UCI) y el propio Ministerio de Salud Pública, han desarrollado sistemas encaminados a lograr la informatización de los diferentes niveles de atención de salud. El objetivo primordial del desarrollo de estas soluciones informáticas ha sido proveer al SNS de información consistente, confiable y oportuna para los procesos de toma de decisiones y el mejoramiento de los procesos médicos asistenciales, garantizándose con esto el incremento en la calidad y seguridad de la atención médica a la población. La informatización de la APS es esencial en este sentido.

El programa de informatización de la APS tiene claramente definidos tres propósitos fundamentales: [8]

- ❖ Fortalecer la conectividad de las instituciones de salud territoriales que permita una interrelación más efectiva entre ellas, así como un flujo informativo eficiente y eficaz para la toma de decisiones a nivel local y niveles superiores.
- ❖ Fortalecer el Subsistema del Médico y la Enfermera de la Familia, para lo cual hay que centrar los esfuerzos en los policlínicos, consultorios y comunidad.
- ❖ Fortalecer el resto de los Subsistemas de salud territorial, interrelacionados con el primero que le brindan la interacción necesaria.

Es importante considerar que con la informatización del SNS, se pretende además lograr la integración y centralización de la información ya que la misma debe estar siempre disponible y accesible al personal

autorizado independientemente de donde se haya generado. En este sentido, dicha estrategia permitirá el aprendizaje basado en la experiencia y ayudará a la toma de decisiones.

1.4 Medicina Interna e Hipertensión Arterial

Con el desarrollo de la medicina hospitalaria, muy ligada a las clínicas universitarias, comenzó a perfilarse, en las postrimerías del siglo XIX, una nueva orientación en la medicina general, más ligada a las ciencias básicas biomédicas y a la experimentación, la cual recibió el nombre de **Medicina Interna** [4]. La misma se define como una especialidad médica que se dedica a la atención integral del adulto enfermo, enfocada al diagnóstico y el tratamiento no quirúrgico de las enfermedades que afectan a sus órganos y sistemas internos, y a su prevención. [5]

Esta definición hace alusión a dos características fundamentales de la medicina interna: [6]

- ❖ Separa el campo de acción del internista de las otras especialidades clínicas básicas originales: cirugía, obstetricia-ginecología y pediatría.
- ❖ Explicita que la atención del enfermo es integral, lo cual distingue la acción del internista de quienes ejercen las subespecialidades o especialidades derivadas de la Medicina Interna.

De modo que, los subespecialistas se centran en resolver problemas inherentes a su especialidad derivando al paciente a otros médicos si consideran que el problema clínico está fuera de su área de experiencia. Por su parte los internistas brindan una atención integral al paciente consultando con otros especialistas cuando el problema de salud del paciente alcanza una complejidad diagnóstica que haga razonable contar con su mayor experiencia.

La medicina interna tiene además otra característica importante que la distingue de la medicina general y es su capacidad de incorporar al conocimiento clínico de las enfermedades y a su tratamiento los progresos de las ciencias. Se hace necesario, para dar cumplimiento al concepto de atención integral al paciente, que se incorporen a las ciencias biológicas los avances en psicología y en ciencias sociales que contribuyen al propósito integrador.

Entre las enfermedades que trata la medicina interna se encuentra la Hipertensión Arterial. La misma constituye una enfermedad del sistema cardiovascular frecuente en la población adulta de Cuba, además

se considera un factor de riesgo modificable mayor, y se establece al respecto que es el principal factor de riesgo después de los 45 años de edad [7]. Este problema de salud puede ser asintomático y tiene complicaciones mortales si no es atendido de manera adecuada. La relevancia de la HTA no reside en sus características como enfermedad, sino en el incremento del riesgo de padecer enfermedades vasculares que confiere. La mortalidad por esta enfermedad ha disminuido con los años sin embargo continúa siendo una de las causas más frecuentes de muerte.

1.1 Sistemas Expertos

Normalmente puede definirse un experto como una persona con capacidad para resolver problemas específicos en un ámbito determinado. Esta habilidad proviene, generalmente, de una vasta experiencia y de un conocimiento detallado y especializado de los problemas que maneja.

El conocimiento de un experto humano presenta dos aspectos: [9]

- ❖ **Conocimiento formal o público:** Está basado en las leyes conocidas de la naturaleza y en métodos formales de razonamiento. Este conocimiento está, usualmente, incluido en los libros de texto relativos al tema y, como está formalizado, puede ser transferido a otras personas con relativa facilidad.
- ❖ **Conocimiento informal o privado:** Según el juicio, la opinión o el criterio del experto, consta de sugerencias y de cálculos aproximados. Constituye la parte más importante del conocimiento de los expertos. Se presume que el mejor experto es el que posee un conjunto más rico de este tipo de conocimiento.

La experticia se resume entonces en el conjunto de ambos tipos de conocimiento en el ámbito de un campo dado y las habilidades conocidas para resolver problemas en dicho campo.

La **Inteligencia Artificial** (IA) es la rama de la computación que se encarga de modelar la inteligencia humana en sistemas computacionales. Mediante la misma se han desarrollado los **Sistemas Expertos** (SE), constituyendo estos un firme y verdadero avance en este campo y uno de los éxitos comerciales más importantes de la IA.

El concepto de SE fue introducido por Feigenbaum como consecuencia de la experiencia del proyecto DENDRAL. En la década de 60 se produjo un cambio en la concepción de la acumulación de conocimiento y se pasó de los intentos de emular el pensamiento humano mediante potentes métodos de razonamiento a hacer mayor énfasis a los sistemas de razonamiento de propósito específico. Se define así un **Sistema Experto** como un programa o conjunto de programas informáticos concebidos para resolver problemas o situaciones de un modo similar al que utilizaría un experto humano dentro de su dominio de su competencia. En consecuencia, el sistema experto persigue la simulación de los procesos de asociación, análisis y síntesis que se realizan en el cerebro humano [10]. Los SE pueden incorporar miles de reglas y tienen la ventaja frente a otros tipos de programas de IA, de proporcionar gran flexibilidad a la hora de incorporar nuevos conocimientos. Esto es posible solo con agregar la nueva regla que se desee sin necesidad de cambiar el funcionamiento propio del programa. El comportamiento “inteligente” de los SE está basado en el uso del conocimiento informal o privado. Así pues un SE será más inteligente mientras mayor cantidad de conocimiento posea y emplee al igual que un humano.

1.3.1 Estructura de los Sistemas Expertos

Básicamente un S.E. se divide en tres módulos: [11]

- ❖ **Base de Conocimiento (BC):** En ella se almacena el conocimiento sobre algún dominio de aplicación mediante alguna Forma de Representación del Conocimiento (FRC). Es el corazón del SE.
- ❖ **Máquina de Inferencia (MI):** Implementa algún Método de Solución de Problemas (MSP) que manipula el conocimiento almacenado en la BC e informaciones sobre estados iniciales, estados actuales de la solución del problema, etc., las cuales procesan dinámicamente en una estructura que se le llama Base de Datos (BD) o Memoria de Trabajo.
- ❖ **Interface de Usuario (IU):** Mediante ella el usuario plantea los problemas al SE, recibe preguntas del mismo y ofrece las explicaciones necesarias. La IU se compone de: preguntas, frases y menús. Pueden incluir módulos de explicaciones asociadas a cada regla para proveer información suplementaria. Usualmente la información que procesa la IU debe suministrarse con una sintaxis estricta, formateada de alguna forma.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

Los SE inicialmente se construían partiendo de cero hasta que surgieron los **Sistemas Expertos Vacíos o Esenciales** (armazón), constituidos por la máquina de inferencia y la interface con el usuario. Estos se caracterizan fundamentalmente por no contener BC permitiendo con un mismo armazón crear diferentes SE simplemente variando el contenido de su BC. Tienen, por lo general, facilidades de edición y actúan como un compilador o intérprete en el sentido de que las reglas entradas en el editor son compiladas por el armazón y llevadas a un código final que usan la MI y la IU.

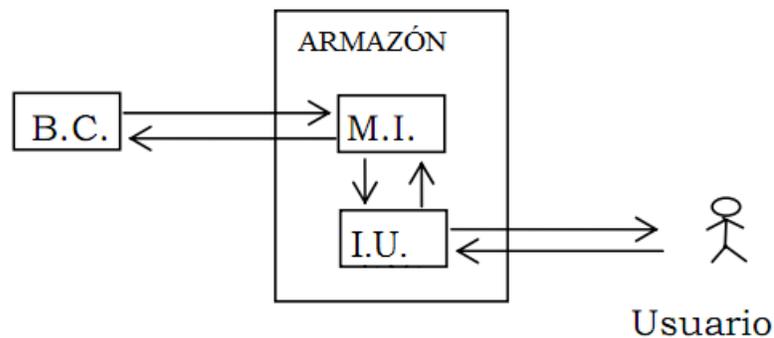


Figura 1.1 Arquitectura de un SE.

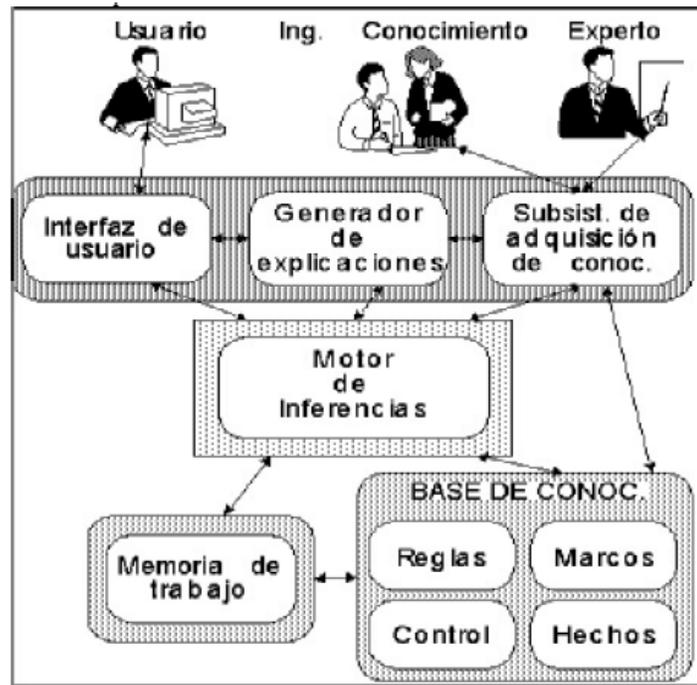


Figura 1.2 Funcionamiento de un SE.

1.3.2 Elementos comunes de los Sistemas Expertos

A pesar de que existen muchas implementaciones de SE, hay rasgos comunes que los distinguen: [12]

- ❖ **La acumulación y codificación del conocimiento:** La BC contiene el conocimiento tanto público como privado que posee el experto para la solución de problemas en un dominio de aplicación concreto.
- ❖ **El conocimiento del SE es explícito y accesible:** El conocimiento de los expertos es expresado bastante explícitamente en forma de un fichero en la computadora. De esta manera, la BC es fácilmente accesible por los expertos y modificada por ellos.
- ❖ **Alto nivel de experticia:** Es el rasgo más útil. Los SE resuelven problemas que requieren años de entrenamiento a los humanos.
- ❖ **Habilidad para ofrecer facilidades de entrenamiento:** Los SE contienen el conocimiento necesario y la habilidad para explicar su proceso de razonamiento.

- ❖ **Puede actuar como una teoría de procesamiento de información:** El SE permite al usuario evaluar los efectos potenciales de nuevos datos y comprender la relación entre ellos. Igualmente puede evaluar el efecto de nuevas estrategias o procedimientos mediante la inclusión o modificación de las reglas, o sea, permite analizar cómo se comporta el SE ante un cambio de los datos iniciales.
- ❖ **Es un modelo computacional para ayudar a la toma de decisiones:** Los SE constituyen un nuevo modelo computacional donde el formalismo para representar el conocimiento es independiente del método de solución empleado.
- ❖ **Debe ser capaz de adquirir nuevos conocimientos y perfeccionar el que posee:** Ellos deben ser capaces de tener atributos similares a los de los expertos humanos.

Un SE debe, fundamentalmente, poder explicar su funcionamiento ante dos situaciones esenciales. Primeramente debe ser capaz de explicar cómo alcanzó un determinado resultado en caso de que el usuario tenga dudas del mismo o para ayudarlo a aprender sobre el resultado alcanzado. Y en segundo lugar un SE debe poder explicar por qué se hace una pregunta cuando está le parezca irrelevante al usuario o en caso de que la respuesta requiera un esfuerzo adicional por parte del usuario.

1.3.3 Ventajas de los Sistemas Expertos

La experticia de los SE presenta notables ventajas sobre la de los expertos humanos. La primera es permanente y consistente ya que no está relacionada con el estado “físico o mental” del SE. Además es fácil de transferir y de documentar. Los SE además son relativamente baratos pues su desarrollo es costoso pero tienen bajo costo de operación y pueden trabajar las 24 horas del día sin inspiración pues su comportamiento es rutinario. Los SE además presentan ventajas frente a los programas convencionales ya que pueden ser modificados con facilidad sin afectar la estructura del programa. Dicha flexibilidad proporciona una mayor eficiencia en la programación y la comprensibilidad y precisamente, los distingue como un modelo computacional nuevo.

En resumen, los SE no sustituyen a los expertos humanos sino que los ayudan a progresar participando en actividades más desafiantes intelectualmente, por lo que el conocimiento contenido en un SE es una nueva fuente práctica de conocimientos. Son frecuentemente utilizados como apoyo a los usuarios expertos en algún dominio de aplicación definido.

1.3.4 Limitaciones actuales de los Sistemas Expertos

A pesar de lo anteriormente expuesto los SE tienen actualmente limitaciones tales como: [13]

- ❖ Dominio estrecho de experiencia.
- ❖ Incapacidad de realizar razonamiento con sentido común.
- ❖ Las MI son demasiado simples y no permiten una visión global de los procesos de inferencia.
- ❖ El conocimiento profundo y casual no es explotado.
- ❖ Rango limitado de la capacidad de explicación.
- ❖ Posibilidades restringidas de comunicación, pues los lenguajes orientados a un problema no son flexibles ni lo suficientemente naturales.
- ❖ Incapacidad del SE para reconocer el límite de su habilidad. Constituye la limitación más seria. No tienen un conocimiento profundo acerca de su propia gestión. No pueden razonar sobre su campo de acción y limitaciones.

1.3.5 Tipos de Sistemas Expertos

Principalmente existen tres tipos de Sistemas Expertos:

- ❖ Basados en reglas previamente establecidas.
- ❖ Basados en casos.
- ❖ Basados en redes bayesianas.

En cada uno de ellos la solución a un problema planteado se obtiene de la siguiente forma:

- ❖ *Aplicando reglas heurísticas apoyadas generalmente en lógica difusa para su evaluación y aplicación:* estos trabajan mediante la aplicación de reglas, comparación de resultados y aplicación de las nuevas reglas basadas en situación modificada. También pueden trabajar por inferencia lógica dirigida, bien empezando con una evidencia inicial en una determinada situación y dirigiéndose hacia la obtención de una solución, o bien con hipótesis sobre las posibles soluciones y volviendo hacia atrás para encontrar una evidencia existente (o una deducción de una evidencia existente) que apoye una hipótesis en particular.

- ❖ *Aplicando el razonamiento basado en casos, donde la solución a un problema similar planteado con anterioridad se adapta al nuevo problema:* es el proceso de solucionar nuevos problemas basándose en las soluciones de problemas anteriores. Un mecánico de automóviles que repara un motor porque recordó que otro auto presentaba los mismos síntomas está usando razonamiento basado en casos.
- ❖ *Aplicando redes bayesianas, basadas en estadística y el teorema de Bayes:* Una red bayesiana, o red de creencia, es un modelo probabilístico multi-variado que relaciona un conjunto de variables aleatorias mediante un grafo dirigido que indica explícitamente influencia causal. Gracias a su motor de actualización de probabilidades, el Teorema de Bayes, las redes bayesianas son una herramienta extremadamente útil en la estimación de probabilidades ante nuevas evidencias. Una red bayesiana es un tipo de red causal. Un híbrido de red bayesiana y Teoría de la Utilidad es un diagrama de influencia.

1.4 Razonamiento Basado en Casos

En los últimos años, el **Razonamiento Basado en Casos (RBC)** ha experimentado un rápido crecimiento desde su nacimiento en Estados Unidos. Lo que sólo parecía interesante para un área de investigación muy reducida, se ha convertido en una materia de amplio interés, multidisciplinar y de gran interés comercial.

El RBC es un paradigma de resolución de problemas. Sin embargo, lo que lo hace tan especial son sus diferencias con el resto de los acercamientos de la IA. Los sistemas que utilizan RBC no confían únicamente en el conocimiento general del dominio del problema ni realizan asociaciones a lo largo de relaciones entre descripciones del problema y conclusiones sino que son capaces de utilizar conocimiento específico de experiencias previas, es decir, situaciones de un problema concreto (casos). Un problema nuevo es resuelto cuando se encuentra un caso pasado similar y se reutiliza en la situación del problema nuevo. Otra diferencia importante a considerar es la capacidad de aprendizaje incremental y sostenido que tienen estos sistemas. Los mismos guardan una experiencia nueva cada vez que se resuelve un problema y de esta forma esta información pasa a estar disponible a partir de ese momento para la resolución de futuros problemas.

Para comprender mejor este tipo de sistemas primeramente se debe conocer que un “caso” es la definición completa, clara y precisa de las características de un problema particular que lo distinguen de otros problemas, y las acciones que se deben tomar para su corrección.

De forma general se puede decir que el principal objetivo que persigue el razonamiento basado en casos es identificar el problema con el que se encuentra, y para ello es necesario encontrar uno o varios casos anteriores a este, que proporcionen la solución para el caso en cuestión. Una vez encontrada la solución será posible añadir este último caso dentro de la base del conocimiento del sistema.

1.4.1 Ciclo del razonamiento basado en casos

El ciclo del RBC está formado por los siguientes elementos (Fig. 1.3):

1. Recuperación de casos.
2. Adaptación.
3. Crítica y justificación.
4. Evaluación.
5. Almacenamiento.

La **recuperación** de casos es la selección, en la base de conocimiento, de aquellos casos cuya descripción se ajusta más a la información presentada en el nuevo caso.

La **adaptación** consiste en adecuar la solución del caso más parecido a las condiciones del nuevo caso. Esto es necesario, dado que normalmente los fenómenos o síntomas que se presentan en un diagnóstico, no son idénticos a los ocurridos en los casos anteriores.

La etapa de **crítica y justificación**, consiste en la validación de la solución propuesta. Esta validación se realiza contrastando diferentes soluciones o simulando la solución para estimar qué tan acertada es. Esta etapa está altamente influenciada por el grado de conocimiento que tiene el experto sobre el fenómeno ocurrido, y es él el que juzga la efectividad de la solución propuesta con base en su experiencia.

En la etapa de **evaluación** se aplica la solución propuesta y se analiza el resultado de su aplicación. Si los resultados son los esperados se confirma la solución, pero si existen diferencias, se debe averiguar

por qué ocurrieron tales diferencias y cómo pueden evitarse. Esta información debe servir para mejorar la definición del caso.

Finalmente, el **almacenamiento** consiste en registrar, en la base de conocimiento, la información derivada del nuevo caso, ya sea como un caso nuevo o un caso mejorado.

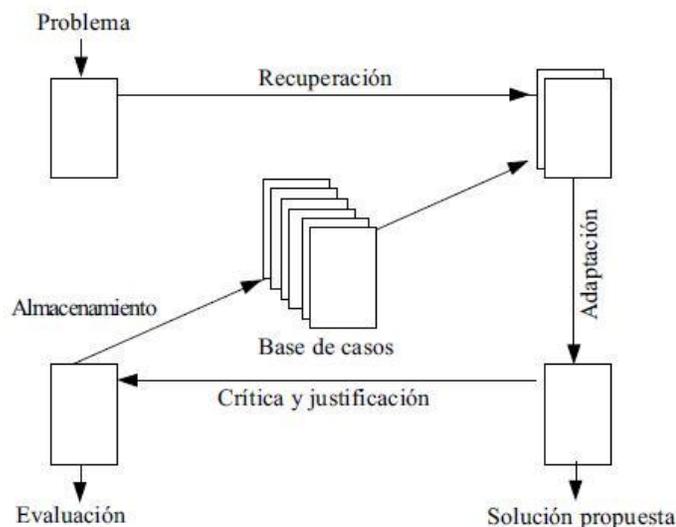


Figura 1.3 Funcionamiento del RBC.

Teniendo en cuenta el ciclo del razonamiento basado en casos se puede determinar que la calidad de un sistema que utilice RBC dependerá de los siguientes factores:

- ❖ La experiencia que tiene.
- ❖ La habilidad para entender situaciones nuevas en términos de experiencias pasadas.
- ❖ Su capacidad de adaptación.
- ❖ Su capacidad de evaluación y reparación.
- ❖ Su habilidad para integrar nuevas experiencias en su memoria adecuadamente.

Las tareas inherentes a los sistemas de RBC se clasifican en dos grupos fundamentales: tareas de clasificación y tareas de síntesis. En las tareas de clasificación los nuevos casos se emparejan con aquellos de la base de casos para determinar qué tipo, clase o caso es. La solución del caso que

mejor ajusta es la que se reutiliza. Por su parte las tareas de síntesis intentan crear una nueva solución combinando partes de soluciones previas. Los sistemas de RBC que realizan tareas de síntesis deben realizar adaptación y son normalmente sistemas híbridos que combinan RBC con otras técnicas.

1.4.2 Ventajas del uso de razonamiento basado en casos

El razonamiento basado en casos provee numerosas ventajas:

1. Reduce la tarea de adquisición de conocimiento pues la misma consiste, básicamente, en una colección relevante de experiencias (casos) y su representación y almacenamiento.
2. Evita la repetición de errores del pasado pues los sistemas que guardan tanto los fallos como los éxitos, así como las causas de los fallos, utilizan esta información para predecir fallos futuros.
3. Provee gran flexibilidad en el modelado del conocimiento pues usan las experiencias pasadas como dominio y dan una solución razonable, previa adaptación, de este tipo de problemas.
4. Permite al razonador proponer soluciones en dominios que no son del todo entendidos por él. Hay dominios que son imposibles de entender completamente, a menudo porque dependen del comportamiento humano impredecible, la economía por ejemplo. Sin embargo, el razonamiento basado en casos permite tomar ciertas premisas y predicciones basándonos en lo que funcionó en el pasado.
5. Permite hacer predicciones del posible éxito de una solución propuesta. Cuando la información se almacena teniendo en cuenta el nivel de éxito de las soluciones previas, el razonador basado en casos puede ser capaz de predecir el éxito de una solución propuesta para el problema actual.
6. Aprende con el tiempo. A medida que los razonadores basados en casos son más usados, encuentran más situaciones de problemas y crean más soluciones por lo que el sistema tendrá más variedad de situaciones y más grado de refinamiento y éxito.
7. Se propone soluciones a problemas rápidamente en dominios que requieren un gran procesamiento para crear una solución.
8. Provee un medio de justificación. El razonamiento basado en casos puede dar un caso previo y su solución (con éxito) para convencer al usuario, o justificar, una solución propuesta al problema actual. Si el usuario deseara una medida de calidad de la solución, el sistema podría cuantificar cuanto éxito tuvo el caso pasado y qué grado de similitud hay con el caso actual y el pasado.

9. Es un reflejo del razonamiento humano, ya que los seres humanos usan una forma de razonamiento basado en casos. Esto es una gran ventaja a la hora de poder entender el funcionamiento del sistema, así como la justificación de una solución propuesta por un sistema basado en casos. [9]

1.4.3 Inconvenientes de uso de razonamiento basado en casos

1. Puede haber una tendencia a usar los casos previos ciegamente, confiando en la experiencia previa sin validarla en la nueva situación.
2. Los casos previos pueden predisponer demasiado al razonador a la hora de resolver un nuevo problema.

Confiar en experiencias previas sin validarlas puede generar soluciones y evaluaciones ineficientes o incorrectas. La recuperación de casos inapropiados puede costar un tiempo considerable o llevar a errores muy costosos. El uso de sistemas basados en casos debe servir de apoyo o ayuda a la toma de decisiones pero siempre dejando al usuario humano un margen de razonamiento.

1.4.4 Funciones de Semejanza

Los componentes fundamentales de un Sistema Basado en Casos son: la base de casos, el módulo de recuperación y el módulo de adaptación.

El módulo de recuperación consta de dos etapas fundamentales: la etapa de acceso y la etapa de recuperación propiamente dicha. [40]

La etapa de acceso presenta entre sus técnicas el acceso exacto y el jerárquico; el sistema realizado utiliza esta última en la cual se seleccionan los casos más similares al nuevo problema. Para determinar qué tan similar es un caso a otro se han desarrollado varias técnicas. La más sencilla consiste en contar el número de características similares entre los dos casos. El problema de esta técnica es que la importancia de las características varía de un contexto a otro. Otra técnica consiste en utilizar un conjunto de heurísticas que permitan determinar cuáles características tienen mayor relevancia (peso) y formular una función de semejanza que involucre la similitud entre cada uno de los rasgos teniendo en cuenta el peso

de los mismos. Como modelo matemático de esta técnica se tiene la siguiente función de semejanza entre un nuevo problema a resolver O_0 y un caso O_t de la base: [40]

$$\beta(O_0, O_t) = \frac{\sum_{i=1}^n p_i \cdot \delta_i(O_0, O_t)}{\sum_{i=1}^n p_i}$$

Dónde:

n : Número de rasgos predictores.

p_i : Peso o relevancia del rasgo i .

$\delta_i(O_0, O_t)$: Función de comparación entre los casos O_0 y O_t atendiendo al rasgo i . Esta función puede estar definida de diferentes formas, por ejemplo:

Para rasgos numéricos.

$$\delta_i(O_0, O_t) = \begin{cases} 1 & \text{si } x_i(O_0) = x_i(O_t) \\ 0 & \text{e.o.c} \end{cases}$$

$$\delta_i(O, O_t) = 1 - \frac{|x_i(O) - x_i(O_t)|}{\max_i - \min_i}$$

\max_i, \min_i : Valores máximo y mínimo respectivamente que alcanza el rasgo i .

El resultado de aplicar esta función de semejanza en el módulo de recuperación, además, brinda la utilidad del caso analizando este problema como un problema para la toma de decisiones. Una vez seleccionados los casos similares en el módulo de recuperación, se efectúa el procedimiento de adaptación, que consiste en la modificación y combinación de las soluciones de los casos similares para formar una nueva solución, una interpretación, o una explicación dependiendo de la tarea que lleve a cabo el sistema. [40]

Entre los métodos que se utilizan en la adaptación se encuentran los siguientes:

- ❖ **Reinstanciación.** En este tipo de técnica se utiliza el marco o contexto de situaciones anteriores, pero con nuevos argumentos.
- ❖ **Ajuste de parámetros.** Se ajustan los parámetros de la solución de casos anteriores de acuerdo con las diferencias entre las descripciones de los casos en cuestión.
- ❖ **Búsqueda local.** Se realizan búsquedas dentro de jerarquías semánticas y se soluciona el problema por analogía. [40]

1.5 Análisis de soluciones existentes

Ámbito Nacional

Sistema Experto para el Diagnóstico y Tratamiento de Infecciones de Transmisión Sexual (DITRITS)

Este sistema fue desarrollado por la Universidad de Cienfuegos en conjunto con especialistas de segundo grado del Hospital Provincial Dr. Gustavo Aldereguía Lima. DITRITS se caracteriza por utilizar un modelo de producción basado en reglas. Está conformado por dos módulos fundamentales, un ejecutable desarrollado en Borland Delphi 7 y otro en Arity Prolog.

DITRITS como aplicación constituye el primer producto cubano para el diagnóstico y tratamiento de Infecciones de Transmisión Sexual (ITS). Como cualquier SE, cuenta con una Base de Conocimientos, una Base de Hechos y un Motor de Inferencia. En dicho sistema la interacción con el usuario se produce

en lenguaje natural de una forma interactiva y siguiendo los patrones de conversación entre seres humanos lo cual proporciona gran facilidad de uso. Sus ventajas están asociadas a las de la mayoría de los SE siendo relevantes la modularidad, pues cada regla es una unidad independiente de conocimiento que puede ser modificada sin afectar el funcionamiento de las demás reglas del sistema proporcionando así flexibilidad a la base de conocimientos. DITRITS además presenta gran uniformidad pues todo el conocimiento del sistema se expresa en el mismo formato y es capaz de dar solución a problemas que incluyen datos incompletos. Además, este sistema no solo puede ser usado para el apoyo al diagnóstico de ITS sino que su uso se extiende al de herramienta de apoyo a la docencia. A pesar de sus ventajas DITRITS centra sus diagnósticos solo en ITS por lo cual tiene un dominio de aplicación restringido.

Sistema Experto para el Diagnóstico de Anomalías Craneofaciales (DIAG)

El sistema DIAG es un SE que se utiliza como herramienta de diagnóstico para especialistas de ortodoncia y estomatología. Además puede ser utilizado como un sistema tutorial inteligente para el estudio de la ortodoncia. Para elaborar DIAG se utilizó Artificial Intelligence and Expert Systems (ARIES), que es un ambiente de trabajo desarrollado por especialistas del Laboratorio de Inteligencia Artificial del Instituto de Cibernética Física y Matemática del Ministerio de Ciencia-Tecnología y Medio Ambiente de Cuba.

La base de conocimientos de DIAG cuenta con 300 proposiciones relacionadas por 250 reglas de producción estructuradas en 8 módulos fundamentales: [14]

- ❖ Anomalías de posición ántero-posterior de los maxilares.
- ❖ Anomalías de volumen de los maxilares.
- ❖ Laterognatismos mandibulares.
- ❖ Alteraciones en la inclinación mandibular.
- ❖ Anomalías de forma mandibular.
- ❖ Anomalías verticales de la cara.
- ❖ Anomalías de posición y dirección de los dientes.
- ❖ Tejidos blandos.

Se incluyen en los módulos, además de los datos de las anomalías investigadas, algunas sugerencias acerca de las diferentes conductas a seguir en el plan de tratamiento, teniendo en cuenta el período de crecimiento y desarrollo en que se encuentra el paciente. Este sistema además presenta una interface de usuario amigable provista por el ambiente de trabajo ARIES y dirigida a especialistas y estudiantes de ortodoncia y estomatología.

En una prueba realizada al sistema DIAG con 80 Historias Clínicas Individuales (HCI), en 69 de las HCI procesadas por el sistema, que representan el 86,25 % del total de los casos analizados, el diagnóstico fue coincidente con el de los especialistas, mientras que en 5 casos (6,25 %) el diagnóstico fue erróneo y en 6 casos (7,5 %) no se pudo establecer diagnóstico alguno [15]. Se debe tener en cuenta que en mucho de los casos la confección de las HCI fue pobre sin embargo el sistema logró un diagnóstico acertado lo cual demuestra la efectividad de la representación del conocimiento lograda en cada módulo, así como de la flexibilidad admitida para diagnosticar con un mínimo de datos de peso.

DIAG ofrece establece un diagnóstico con rapidez y seguridad siempre que los datos aportados sean confiables y suficientes. Además está dotado de una ayuda en cualquier momento en que sea requerida que sirve de banco de conocimientos para los que se inician en la ortodoncia como de recordatorio para los expertos. Sin embargo, por la multiplicidad de desviaciones del crecimiento y desarrollo craneofacial que se pueden encontrar, tanto en la práctica diaria, como las reportadas en la literatura actual, el sistema en su versión inicial no es general y solo abarca un espectro limitado con relación a las múltiples combinaciones que se pueden producir. A pesar de esto, el sistema es susceptible de asimilar más conocimientos.

HyperWeb

Esta propuesta consiste en la creación de un sistema que permita al médico la ayuda a la toma de decisiones en temas como: la evaluación clínica del paciente (HC, examen físico, proceso de diagnóstico) y en la indicación y control del tratamiento no farmacológico personalizado. Con el fin de aportar al objetivo de promover estilos de vida más saludables, se propone la elaboración de un módulo que sea usado por la población en general, en el cual se orienten sobre cómo eliminar hábitos tóxicos, medidas para mantener el peso adecuado y una dieta balanceada, así como la aplicación de algunas técnicas de

autorrelajación, conocer sus factores de riesgo de HTA, calcular su probabilidad de padecer de Cardiopatía Isquémica, entre otras.

El sistema sobre tecnología Web con las siguientes características:

- ❖ ASP.NET como tecnología Web, utilizando la plataforma .NET (Framework 1.1) y Visual Studio .NET como IDE de desarrollo.
- ❖ Lenguaje de programación C# para la implementación.
- ❖ Dreamweaver MX 2004 como herramienta para el diseño de las páginas Web.
- ❖ SQL Server 2000 Enterprise Edition como gestor de la base de datos.
- ❖ El despliegue se realizará, al menos inicialmente, sobre el sistema operativo Windows 2003 Advanced Server Enterprise Edition e IIS 6.0 como servidor Web.
- ❖ Como aplicaciones clientes podrán utilizarse la mayoría de los navegadores integrados en los sistemas operativos actuales tales como: Internet Explorer, Mozilla, Netscape, entre otros.

Ámbito Internacional

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

Una de las primeras manifestaciones exitosas de SE se produjo precisamente en el campo de las Ciencias Médicas en la década del 70 con el desarrollo del sistema MYCIN, probablemente el más famoso de los SE, concebido para el diagnóstico de enfermedades infecciosas en la sangre. El mismo fue desarrollado por Edgar ShortLiffe, en la Universidad de Stanford, escrito en lenguaje de programación LISP e inspirado en el SE Dendral. Su desarrollo en este lenguaje de programación le permitió gran flexibilidad pero pérdida de modularidad y claridad. Como SE, MYCIN era capaz además de “razonar” el proceso seguido para llegar a sus diagnósticos, y de recetar medicaciones personalizadas a cada paciente (según su estatura, peso, etc.). El sistema diagnosticaba la causa de la infección usando el conocimiento relativo a la infección de los microorganismos con historiales de pacientes, síntomas y los resultados de los test de laboratorio. [16]

MYCIN se basaba para realizar los razonamientos en factores de certeza y certidumbre ya que sus autores decidieron no seguir la teoría de probabilidad porque: [17]

1. Los expertos se resisten a expresar sus procesos de razonamiento en términos probabilísticos coherentes.
2. La probabilidad requiere de grandes cantidades de datos y muchas aproximaciones y suposiciones.

Desarrollaron entonces la técnica de factores de certeza, cuya base es el concepto de confirmación - interpretación lógica de probabilidad subjetiva.

MYCIN era un sistema lo bastante robusto como para determinar correctamente conclusiones cuando algunas evidencias eran incompletas o incorrectas, podía explicar sus propios procesos de razonamiento y fue uno de los primeros sistemas genuinamente amigables. Razonaba heurísticamente, interactuando con los humanos utilizando lenguaje natural y razonaba acerca de descripciones simbólicas además de que podía contemplar hipótesis competitiva. Este sistema supuso el punto de partida para el desarrollo posterior de un conjunto de SE, muchos con aplicaciones en el campo de la medicina y otros en otras esferas.

En investigaciones realizadas por Stanford Medical School se determinó que MYCIN tenía una tasa de aciertos de aproximadamente el 65%, lo cual mejoraba las estadísticas de la mayoría de los médicos no especializados en el diagnóstico de infecciones bacterianas para aquella época. Sin embargo poco a poco este sistema fue cayendo en desuso por las debilidades que presentaba, muchas de ellas inherentes a su técnica, tales como:

- ❖ La evidencia es condicionalmente independiente de la hipótesis y su negación.
- ❖ La red de inferencia debe corresponder a un árbol para que los resultados sean coherentes.
- ❖ Las fórmulas para conjunción y disyunción (min y max) sólo son válidas si uno de los términos es subconjunto del otro.

Además, estas suposiciones no son válidas en muchas aplicaciones por lo que el método de MYCIN no se puede generalizar. Se agregaron es estos problemas también cuestiones éticas y legales pues si el diagnóstico de un paciente era incorrecto se caía en la disyuntiva de a quien se debía culpar, ¿al médico o al programador? Otro de los motivos se achaca a la excesiva dificultad que suponía el mantenimiento del programa. Era este uno de los principales problemas de MYCIN, y en general, de los sistemas expertos de

la época, en los cuales se dedicaban muchos esfuerzos y recursos a extraer el conocimiento necesario de los expertos en dominio para construir el motor de inferencia.

Sistema Experto para el Diagnóstico en Medicina Interna (INTERNIST/CADUCEUS)

El sistema INTERNIST contempla el diagnóstico de las enfermedades de medicina interna u hospitalaria. Fue desarrollado en la universidad norteamericana de Pittsburg en 1977 por el informático Harry E. Pople y el médico especialista en medicina interna Jack D. Myers. Este SE fue escrito en el lenguaje de programación INTERLISP. Simultáneamente sus creadores han trabajado en una versión más compleja de INTERNIST denominada CADUCEUS.

INTERNIST/CADUCEUS aborda todo el campo de la medicina interna por lo cual está obligado a tratar con un gran número de enfermedades y a considerar todas las combinaciones e interacciones posibles entre ellas. Este es un SE autónomo experimental que conoce 100 000 síntomas secundarios y 350 síntomas primarios de 500 enfermedades diferentes. Tanto los especialistas de IA como en medicina reconocen que INTERNIST/CADUCEUS incorpora profundos conocimientos de medicina interna y es capaz de diagnosticar el 75 % de las enfermedades conocidas. A pesar de esto su uso en sus inicios irritaba a los médicos ya que el sistema se demoraba demasiado considerando enfermedades irrelevantes para el caso que se trataba. Este problema se resolvió en gran medida en la versión denominada CADUCEUS, sin embargo, el médico debe emplear veinte o treinta minutos de diálogo interactivo de entrada de datos para la descripción del paciente. Esto, lógicamente, demora el proceso de consulta y hace engorroso el trabajo del médico a pesar de que el sistema le ayude en su diagnóstico.

DiagnosMD

DiagnosMD es una herramienta informática que tiene como objetivo ayudar al profesional médico durante su consulta para resolver dudas y ofrecerle información rápida y útil. Creado por el Farmacéutico e Ingeniero Javier Valilla Masegú, ha contado con la colaboración de prestigiosos médicos, profesores y catedráticos de distintas facultades de medicina españolas. Como sistema para el diagnóstico diferencial, DiagnosMD es mucho más potente que cualquier libro por su capacidad de combinar datos clínicos del paciente (desde los parámetros de edad, sexo, países, etc.) con los datos que puedan interactuar con su

estado de salud. Ejemplo de ello es la capacidad de presentar enfermedades tropicales en pacientes que hayan viajado o provengan de otros países. [18]

Para los médicos de APS la utilidad de DiagnosMD es muy grande ya que permite consultar rápidamente protocolos de APS ante los motivos de visita más frecuentes. También puede ayudarle a realizar mejores peticiones y un mejor direccionamiento al especialista más adecuado. Por su parte, a los especialistas de medicina interna puede ayudarles para el diagnóstico diferencial teniendo en cuenta que tiene un buen nivel y además que éste continuamente está mejorando. Su principal función es la ayuda al diagnóstico médico y para ello DiagnosMD ayuda a pensar en enfermedades poco frecuentes o raras (representan el 85% de las descritas y combina varios parámetros (hasta 30 signos y/o síntomas) para dar un listado de enfermedades teniendo en cuenta: sexo, edad del paciente, valor de cada parámetro en cada diagnóstico, reglas de decisión, posibilidad de fallos, etc. El listado de enfermedades posibles según el caso clínico introducido puede incluso ordenarse por prevalencias o por urgencias. Permite además valorar todos los países del mundo, de forma que la prevalencia de las enfermedades las varía e incluso las enfermedades que no se producen en un país, no las presenta. Es una opción muy útil para pacientes que viajan a otros países o inmigrantes.

Este sistema además cuenta con la novedosa e importante posibilidad de mostrar la posible influencia de los fármacos y plantas medicinales en la HCI del paciente. De modo que si se introducen los datos de los fármacos y las plantas medicinales que toma un paciente conjuntamente con los datos de los síntomas que presenta, DiagnosMD es capaz de determinar las posibles contraindicaciones, precauciones, interacciones, probables alteraciones analíticas y posibles efectos adversos presentes en el paciente. Esto resulta de gran utilidad dada la frecuente polimedicación, y en pacientes con un tratamiento mayor de seis fármacos es muy probable encontrar interacciones, y DiagnosMD las presenta en orden de importancia. Brinda la posibilidad también de realizar recetas con avisos automáticos y rápidos de posibles errores, como, por ejemplo, recetar fármacos que están contraindicados o que interaccionan con otros fármacos que ya toma el paciente. La rapidez de los avisos es casi instantánea y se produce automáticamente al mandar la orden de imprimir la receta.

Además dicho sistema permite consultas rápidas de enfermedades (más de 5.000), fármacos, dietas para enfermedades, interacciones de fármacos y plantas medicinales con alimentos concretos, actuación ante una intoxicación por fármacos, guías de actuación en APS y en Urgencias, información de pruebas

analíticas y dispone de un atlas de Medicina con 6.000 imágenes especialmente de Dermatología para Medicina Interna. Ofrece también la posibilidad de ver de forma rápida y sencilla los fármacos especialmente indicados en las distintas patologías. Los fármacos se presentan ordenados en función del año y del uso en la patología. Se pueden crear también informes propios para cada enfermedad e incluye 400 monografías de plantas medicinales.

A pesar de todas sus potencialidades hay que hacer notar que DiagnosMD resulta limitado para los especialistas. También, como SE, tiene sus restricciones, y sólo sirve para pacientes que presentan una enfermedad, lógicamente no sirve para ayudar a diagnosticar varias enfermedades a la vez, de todas formas el clínico puede introducir solo los datos relevantes de la enfermedad que pretende diagnosticar. Tampoco son incluidos en el motor de inferencia los efectos secundarios de los fármacos, si bien son manejados en la base de conocimientos de DiagnosMD en la consulta de fármacos. Sin embargo se están aumentando el número de reglas de la base de conocimientos, así como la elección cada vez más refinada de las reglas utilizadas en el motor de inferencia por lo que es considerado un sistema actualizable.

Casey

Casey es un sistema basado en casos tipo solucionador de problemas y enfocado al diagnóstico de enfermedades del corazón diseñado en 1989. Parte de una biblioteca de 25 casos resueltos por el sistema experto basado en reglas Heart Failure Program. Casey toma una descripción de los síntomas del paciente y produce con ellos una red causal de estados internos. Como resultado de este proceso, en la comparación no sólo se usan los síntomas actuales, sino también estados patológicos que el sistema infiere.

Este sistema diagnostica los pacientes aplicando heurísticas de adaptación y emparejamiento basado en el modelo independientes del dominio y son tan precisas como el modelo del dominio en el que se aplican. Al presentarse un nuevo caso, Casey busca los casos de pacientes con síntomas similares y si se encuentra caso que se asemeje lo suficiente trata de adaptar el diagnóstico recuperado. Para realizar dicha adaptación el sistema usa el modelo causal teniendo en cuenta las diferencias entre el problema y los casos. Como la adaptación se basa en un modelo causal válido, el diagnóstico es preciso como si estuviera hecho desde cero en ese mismo modelo, lo que hace al sistema más eficiente.

Otra etapa importante durante el ciclo del razonamiento basado en casos es la justificación de la solución como ya se ha visto anteriormente. En el sistema Casey las explicaciones también son incluidas en los casos y posteriormente usadas en la modificación de la solución.

De manera general se puede concluir que, a pesar de que muchos de estos SE se encuentran en funcionamiento y tienen un buen desempeño partiendo de sus potencialidades, solo son factibles en un dominio estrecho de aplicación y ninguno centra su diagnóstico en la HTA. Otro aspecto a tener en cuenta es que por las características de sus diseños muchos de ellos no permiten la integración con los sistemas que se encuentran en desarrollo actualmente en el país. Además, estos programas son del tipo software propietario por lo cual se deben pagar costos de compra y mantenimiento para su uso los cuales muchas veces son bastante elevados teniendo en cuenta la situación económica de Cuba.

1.6 Tendencias y tecnologías actuales a considerar

1.6.1 Sistemas Distribuidos. Modelo Cliente-Servidor

Un sistema distribuido se define como una colección de ordenadores autónomos conectados por una red, y con el software distribuido adecuado para que el sistema sea visto por los usuarios como una única entidad capaz de proporcionar facilidades de computación. Se podría ver un sistema distribuido como un grupo de ordenadores independientes que son percibidos por los usuarios como un único ordenador. [19]

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

1.6.2 Patrón Arquitectónico

De forma general un patrón es un modelo a seguir para darle solución a un determinado problema, estos surgen de la experiencia de los seres humanos al tratar de lograr ciertos objetivos, capturando la experiencia existente y probada para promover las buenas prácticas. Tienen como objetivo la creación de un lenguaje común para la comunidad de desarrolladores permitiéndole su uso miles de veces sin hacerlo

quiera en dos ocasiones de la misma forma. Existen varios tipos de patrones, dependiendo del contexto particular en la cual sean aplicados o de la etapa en el proceso de desarrollo, algunos de estos tipos son: de Diseño, de Arquitectura, de Negocios, de Análisis, para ambientes distribuidos, entre otros.

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista. [20]

Este modelo de arquitectura presenta varias ventajas:

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

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

1.6.3 Marcos de desarrollo, bibliotecas y componentes

Los marcos de desarrollo se pueden considerar como soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución). Aceleran el proceso de desarrollo, permiten reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Java Server Faces (JSF)

La tecnología Java Server Faces (JSF) es un marco de desarrollo de los componentes de la interfaz de usuario del lado del servidor y es válido para todas aquellas aplicaciones web basadas en la tecnología

JAVA. JSF traduce las distintas acciones del usuario en eventos que son respondidos por el servidor regenerando la página original y reflejando los cambios necesarios para la acción realizada.

RichFaces 3.3.1

RichFaces 3.3.1 es un marco de desarrollo de código abierto que añade capacidad JavaScript asíncrono y XML Ajax dentro de aplicaciones JSF 1.2 existentes sin recurrir a JavaScript. RichFaces 3.3.1 incluye ciclo de vida, validaciones, conversores y la gestión de recursos estáticos y dinámicos. Los componentes de RichFaces 3.3.1 están contruidos con soporte Ajax y un alto grado de personalización que puede ser fácilmente incorporado dentro de las aplicaciones JSF 1.2. [21]

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

Ajax4JSF

Ajax4jsf es una biblioteca de código abierto que se integra totalmente en la arquitectura de JSF 1.2 y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript.

Ajax4jsf presenta mejoras sobre los propios beneficios del marco de desarrollo JSF 1.2 incluyendo el ciclo de vida, validaciones, facilidades de conversión y el manejo de recursos estáticos y dinámicos. Permite definir un evento en una página que invoca una petición Ajax y luego las áreas de la página deberían sincronizarse con el árbol de componentes JSF 1.2 después de que la petición Ajax cambie los datos en el servidor.

Facelets

Facelets es un marco de desarrollo simplificado de presentación, donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF 1.2 específicos. Aporta mayor libertad al

diseñador y mejora los informes de errores que tiene JSF 1.2. Permite crear plantillas para construir un árbol de componentes de forma que puedan definirse como composición de otros.

Jboss Seam 4.2.2

JBoss Seam es un marco de desarrollo que integra la capa de presentación (JSF 1.2) con la capa de negocios y persistencia (EJB 3.0). Con Seam basta agregar anotaciones propias de éste a los objetos Entidad y Session de EJB 3.0, logrando escribir menos código Java y XML.

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

Hibernate 3.3

Hibernate 3.3 es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Genera sentencias SQL a partir de sentencias HQL (Hibernate Query Language). A través de la implementación del estándar JPA que provee Hibernate 3.3, se puede realizar el acceso a datos. [22]

1.6.4 Lenguaje de Programación

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

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

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

1.6.5 Servidor de Aplicaciones

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB 3.0), y esto hace que el desarrollo de las aplicaciones sean mucho más simples. [23]

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

1.6.6 Sistema Gestor de Base de Datos

PostgreSQL 8.4 es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Berkeley Software Distribution) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. [24]

PostgreSQL 8.4 utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. PostgreSQL 8.4 funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema . [25]

1.6.7 Metodologías de Desarrollo de Software

En la Universidad de las Ciencias Informáticas se lleva a cabo una guía para el desarrollo del software propuesta por el proceso de mejora en el cual se encuentra inmerso la misma; esta guía se aplica con el objetivo de garantizar que el sistema sea gestionado y controlado durante el desarrollo del mismo. Este proceso de mejora la universidad lo hace para alcanzar el nivel 2 de Integración de Modelos de Madurez de Capacidades (CMMI). Las áreas de procesos que lo conforman son: Administración de Requisitos (REQM), Planeación del Proyecto (PP), Monitoreo y Control del Proyecto (PMC), Medición y Análisis (MA), Aseguramiento de la Calidad de Procesos y Productos (PPQA), Administración de la Configuración (CM) y Administración de Acuerdos con Proveedores (SAM).

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

A esos procesos se aplica el Proceso Unificado de Desarrollo de Software (RUP) para la descripción de los productos de trabajo resultantes de cada uno de sus flujos de trabajo, así como de las fases del ciclo vida definido en el IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos se utiliza el Lenguaje Unificado de Modelado (UML), con la especificación del uso del estándar Modelado de Procesos de Negocio (BPMN) para la descripción de los procesos de negocio.

Fue utilizado el estándar BPMN, que proporciona a los negocios la capacidad de entender sus procedimientos internos en una notación gráfica, facilitando a las organizaciones la habilidad para comunicar esos procedimientos bajo un patrón único. Para el modelado de estos procesos se usó la herramienta case Visual Paradigm que soporta el ciclo de vida completo del desarrollo de software.

Integración de Modelos de Madurez de Capacidades (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Las mejores prácticas de CMMI se publican en documentos llamados modelos, los cuales contienen el conjunto de prácticas relacionadas que son ejecutadas de forma conjunta para conseguir determinados objetivos.

Así es como el modelo CMMI establece una medida del progreso, conforme al avance en niveles de madurez. Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. El alcanzar estas áreas o estadios se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables.

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

El objetivo de la Administración de Requisitos es gestionar los requisitos de los elementos del proyecto y sus componentes e identificar inconsistencias entre estos requisitos, el plan de proyecto y los elementos de trabajo. En este proceso se deben gestionar todos los requisitos del proyecto, tanto técnicos como no técnicos. Estos requisitos han de ser revisados conjuntamente con la fuente de los mismos así como con las personas que se encargarán del desarrollo posterior.

El IPP- 3510:2009, Libro de Proceso para la Administración de Requisitos establece el ciclo de vida a seguir asociado a los proyectos involucrados en el proceso de mejora, el cual consta de 9 fases (Estudio Preliminar, Modelación del Negocio, Requisitos, Análisis y Diseño, Pruebas Internas, Pruebas de

Liberación, Despliegue y Soporte). En este se establece por cada fase la relación con los subprocesos descritos en el libro de procesos específico para el área de Administración de Requisitos, incluye la definición de roles, sus responsabilidades y las habilidades en la ejecución de las actividades de los distintos procesos y los productos típicos de trabajos que se obtienen como resultado de la ejecución de dichas actividades.

1.6.8 Lenguaje Unificado de Modelado (UML)

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

1.6.9 Herramientas

Toda metodología de desarrollo de sistemas debe de sustentarse sobre la base de una herramienta case que garantice un mejor análisis del sistema y la calidad del resultado final, a su vez estas herramientas proporcionan el potencial para mejorar la productividad del analista. La herramienta seleccionada para el sustento de la metodología fue Visual Paradigm suite 3.4, esta herramienta da soporte al modelado visual de UML 2.0.

Para el desarrollo de la aplicación web, se tiene al Entorno de Desarrollo Integrado (IDE) Eclipse SDK en su versión 3.1, con la colección de plug-ins de JBoos Tools, necesarios para utilizar los marcos de desarrollo que se proponen. Además, se utiliza el pgAdmin 1.10.5 para la administración de la base de datos.

En este capítulo se profundizó en los conceptos asociados a la informatización del proceso de toma de decisiones para el diagnóstico de la HTA y que facilitan la comprensión de la investigación. Se realizó un estudio de los sistemas existentes vinculados al problema y de las tecnologías que serán utilizadas en el desarrollo del sistema propuesto. Además, se expuso la justificación para la elección de los lenguajes de programación, la arquitectura, Sistema de Gestor de Bases de Datos, la metodología y las herramientas a utilizar, evaluándose cada una de ellos por sus características y ventajas que aportan facilidad para el desarrollo de la aplicación.

CARACTERÍSTICAS DEL SISTEMA

En este capítulo se abordan los aspectos relacionados con el dominio de la investigación y el sistema. Se hace una explicación detallada de los conceptos fundamentales que sirven de guía para confeccionar el modelo de dominio. Este modelo permite especificar las condiciones, capacidades y cualidades que el sistema debe tener así como definir las actividades que serán objeto de automatización. Se describen además, los requerimientos tanto funcionales como no funcionales del sistema a desarrollar.

2.1 Conceptos Fundamentales

Paciente: Término sanitario el cual se refiere a la persona que recibe un servicio brindado por un profesional de la salud.

Síntoma: Fenómeno que revela una enfermedad. El síntoma es referido de manera subjetiva por el paciente cuando percibe algo anómalo en su organismo.

Diagnóstico: es aquello perteneciente o relativo a la diagnosis. Este término, a su vez, hace referencia a la acción y efecto de diagnosticar (recoger y analizar datos para evaluar problemas de diversa naturaleza). En la medicina, un diagnóstico es el acto de conocer la naturaleza de una enfermedad a través de la observación de sus síntomas y signos.

Tratamiento: Conjunto de medios dígame higiénicos, farmacológicos, quirúrgicos o bien físicos, los cuales tendrán como finalidad primaria la curación o el alivio de enfermedades o algunos síntomas de estas, una vez que ya se ha llegado al diagnóstico de las mismas.

Farmacológico: Tratamiento cuando el paciente es declarado hipertenso y se procede a la toma de medicamentos.

No Farmacológico: Tratamiento recomendado por el médico al paciente antes de ser declarado hipertenso, para evitar llegar a la enfermedad.

Hábitos Tóxicos: Conducta adoptada por la persona que se repite en el tiempo de modo sistemático y provoca daños a su salud.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

Cuestionarios de Preguntas: Sistema de preguntas que se le realizan al paciente para determinar los síntomas, hábitos tóxicos y estilos de vida del mismo.

Casos Similares: Conjunto de casos con rasgos parecidos al nuevo caso a tratar.

Nuevo Caso: Rasgos que presenta el paciente que está siendo diagnosticado.

Base de Casos: Es el conjunto de casos con los que cuenta el sistema para realizar comparaciones con el nuevo caso.

Justificación: Es brindarle al especialista una mayor información acerca del caso que trata y el tratamiento propuesto por el sistema.

Médico: Especialista de la salud que utilizará el sistema realizado para apoyarse a la hora de emitir un diagnóstico y justificación de la enfermedad consultando casos de pacientes anteriores.

2.1.1 Modelo de Dominio

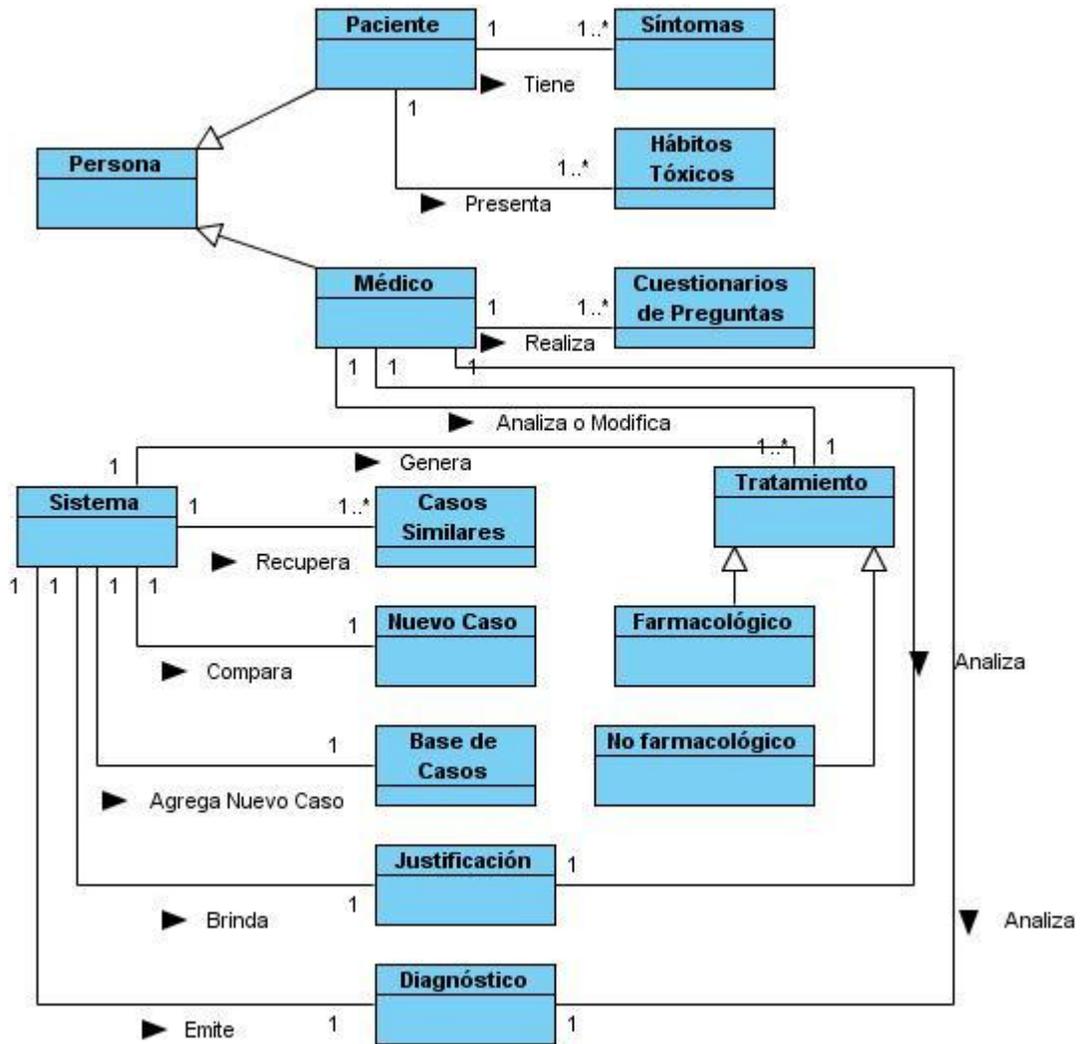


Figura 2.1 Modelo de Dominio.

2.2 Propuesta del Sistema

Para resolver el problema planteado se propone como solución un sistema inteligente para el diagnóstico de la hipertensión arterial, este sistema formará parte del Sistema Integral para la Atención Primaria de Salud (alás SIAPS) perteneciente al Departamento de Atención Primaria de Salud del Centro de Informática Médica. Las características básicas de dicho sistema inteligente estarán centradas en:

- ❖ Registrar los síntomas de los pacientes y tributar al diagnóstico basándose en la recuperación de la información de casos anteriores. Ver **Anexo #3**.
- ❖ Recuperar los casos más semejantes en frente a un nuevo caso.
- ❖ Adaptar la solución del caso más parecido a las condiciones del nuevo caso.
- ❖ Validar y evaluar la solución.
- ❖ Almacenar la información como un nuevo caso o un caso mejorado.
- ❖ Calcular el índice de masa corporal (IMC), peso ideal, porciento de grasa y la clasificación según el IMC.
- ❖ Mostrar tabla de clasificación de la hipertensión arterial.

Para corroborar el sistema a desarrollar se realizaron entrevistas a diferentes especialistas obteniéndose como resultado los criterios que se muestran en el **Anexo #1**.

2.3 Especificación de los requerimientos de software

Los requerimientos son la descripción de las funciones que debe cumplir un software determinado. Se clasifican en: funcionales, que representan las funcionalidades con que debe contar el producto una vez concluido; y no funcionales, que son las cualidades o propiedades que debe poseer el sistema.

2.3.1 Requerimientos Funcionales

Los requerimientos funcionales de un sistema surgen a partir del análisis en profundidad de la situación problémica. Para la definición de los requerimientos del sistema que se propone se utilizó la técnica de entrevista con el objetivo de conocer las necesidades reales de los médicos especialistas como usuarios finales del producto. Como resultado de este estudio, a continuación se listan los requerimientos funcionales definidos.

RF1. Conformar Caso: Permite al especialista completar un cuestionario que le brindará información necesaria para conocer los síntomas del paciente en cuestión.

RF2. Recuperar casos similares: Brinda la posibilidad de buscar en la base de casos del sistema aquellos cuya descripción se ajusta más a la información del nuevo caso.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

RF3. Adaptar casos: Posibilita que se pueda adecuar la solución del caso más parecido a las del nuevo caso.

RF4. Justificar casos: Se valida la solución propuesta para el caso.

RF5. Almacenar caso seleccionado: Registra en la base de casos la información generada por el nuevo caso, ya sea como un nuevo caso o como uno mejorado.

RF6. Realizar propuesta de diagnóstico: Realiza una propuesta de diagnóstico de Hipertensión Arterial a un paciente una vez introducidos los síntomas en el sistema.

RF7. Emitir tratamiento: Permite aplicar un tratamiento farmacológico o no farmacológico al paciente de acuerdo a la propuesta de diagnóstico que brinda el sistema, además justifica la aplicación del mismo teniendo en cuenta las características del paciente.

RF8. Cálculo del IMC, Cálculo del porcentaje de grasa, peso ideal: Permite modificar un tratamiento previamente propuesto por el sistema para ajustarlo a las características del paciente actual.

RF9. Mostrar tabla de clasificación: Muestra las tablas de clasificación de la HTA según el JNC y la junta cubana.

RF10. Mostrar acceso al Join National Comitee (JNC): Link al sitio oficial donde se almacena la información recogida.

RF11. Mostrar cartilla de prevención al paciente: Muestra una ayuda con datos relevantes referente a la enfermedad.

RF12. Justificar diagnóstico: Permite justificar la emisión de un diagnóstico propuesto por el sistema.

2.3.2 Requerimientos no Funcionales

Seguridad de acceso y administración de usuarios

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

- ❖ Se mantendrá seguridad y control a nivel de usuario, garantizando su acceso sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.
- ❖ Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando únicamente la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.
- ❖ Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la base de datos.

Restricciones de diseño

La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio. La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario. La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate 3.3.

Requerimientos de interfaz

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

Requerimientos de hardware

Estaciones de trabajo

En la solución se incluyen estaciones de trabajo para las consultas del Sistema para la Atención Primaria alas SIAPS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web. Por lo que se escogieron estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0 Hz.

Servidores

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. Servidores de Base de datos: Procesador Dual - Core 4GB de memoria y 2x72GB de disco. Servidores de Aplicaciones: Procesador Dual - Core 4GB de memoria y 2x72GB de disco.

Requerimientos de software

El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma Java (Java Virtual Machine, JBoss AS y PostgreSQL). Además se deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2 o versiones superiores de estos.

En este capítulo se realizó el análisis de los conceptos asociados al dominio de la investigación, se explicaron los mismos de forma detallada y se confeccionó el diagrama del modelo de dominio correspondiente. Se describieron además los requerimientos funcionales del sistema que se propone y que demuestran las necesidades que se resolverán al desarrollarse el mismo. También se especificaron los requerimientos no funcionales necesarios para el correcto despliegue y funcionamiento del producto.

DISEÑO DEL SISTEMA

En el presente capítulo se describe la concepción arquitectónica del sistema alas – SIAPS del cual formará parte el sistema para el diagnóstico de la hipertensión arterial. Para una mejor comprensión se especifica el modelo de clases del diseño y se describen los elementos que lo componen. Se justifica además el uso de los patrones utilizados para la construcción de la solución.

3.1 Descripción de la arquitectura

La arquitectura de software puede ser considerada como el diseño de más alto nivel de la estructura de un sistema. La misma debe describir diversos aspectos del software que se comprenden mejor si se utilizan distintos modelos o vistas y establece los fundamentos para que todo el grupo de desarrollo trabaje en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. Es el resultado de ensamblar los elementos arquitectónicos con el objetivo de satisfacer los requerimientos tanto funcionales como no funcionales de un sistema.

3.1.1 Fundamentación del uso de patrones

De manera general puede definirse a un patrón como un modelo a seguir para la realización de una actividad. Existen muchos tipos de patrones, aplicables en dependencia del sistema que se desee implementar. En este caso serán abordados patrones de arquitectura y diseño aplicados al desarrollo de la solución propuesta.

Patrón arquitectónico

En el desarrollo de la solución se utilizará el patrón **Modelo-Vista-Controlador (MVC)**. El mismo brinda la ventaja de separar en componentes distintos los datos, la interfaz del usuario y la lógica del negocio. Esta separación garantiza cierta independencia posibilitando realizar cambios en dichos componentes sin que se afecten los demás.

El uso de este patrón se ve implementado con la utilización del framework Java Server Faces 1.2 (JSF). El mismo es el encargado de recoger y representar la información al usuario por tanto implementa la vista del patrón MCV. JSF permite además controlar las acciones del usuario mediante las funcionalidades

implementadas en las clases controladoras también denominadas Beans y que se encargan de manipular la información contenida en el modelo. También en el componente de la vista se usa la librería de componentes Richfaces 3.3.1 que permite la generación de vistas no necesariamente basadas en HTML y controles como Ajax4Jsf para los controles JSF básicos.

En esta capa se incluye además conversión y validación de campos, establecimiento de reglas de navegación declarativas, la internacionalización y accesibilidad de la interfaz de usuario, un modelo orientado a eventos y combinado con Facelets, se elimina la necesidad de dos motores de renderización (uno para JSF y otro para Java Server Page) mejorando el rendimiento en general además de que brinda la capacidad añadida de la tecnología de plantillas de Facelets. Por su parte los controles para interfaz de usuario de Seam adicionan varias mejoras a JSF, desde validación, integración de la navegación en la interfaz de usuario basada en flujos de navegación o procesos del negocio, entre otros [26].

Por su parte el componente controlador tiene la tarea de manejar y responder las solicitudes de los usuarios, procesar la información y modificar los datos en caso de que sea necesario. En esta capa se utiliza como framework de integración Seam. El mismo es una plataforma de desarrollo potente y de código abierto que permite la integración de tecnologías como Java Script asíncrono y XML con el uso de AJAX, JSF, Java Persistence, Enterprise Java Beans y Business Process Management en una única solución.

El uso de dicha plataforma está fundamentado por la posibilidad que brinda la misma para eliminar la complejidad tanto en la arquitectura como en los niveles de la API. Esto permite a los desarrolladores ensamblar complejas aplicaciones web usando simples clases Java, un conjunto enriquecido de componentes de interfaz de usuario y muy poco de XML. Además Seam introduce el concepto de conversación que da la posibilidad de definir funcionalidades atómicas que involucren varios pedidos al servidor y donde los cambios realizados a las entidades solo serán persistidos en la base de datos una vez terminada la conversación.

Para el acceso a datos, en el modelo, se utiliza la implementación de Java Persistence API (JPA) de Hibernate 3.3 permitiendo así minimizar las configuraciones XML sin chequeo de tipos y eliminar gran parte del código infraestructural en cuanto a transacciones. Gracias a Hibernate Validators se pueden también establecer validaciones. Además en esta capa se encuentran presentes varios patrones de

diseño como: Active record, Identity map, Identity field, Foreign key mapping, Association table mapping, Lazy load y Query object.

El uso de Hibernate como framework de persistencia en el desarrollo de la aplicación permite un trabajo más fácil con las bases de datos relacionales. Esta ventaja da la posibilidad a los desarrolladores de preocuparse solamente por el negocio del sistema y no del cómo se persiste, recupera y elimina la información de la base de datos.

Por último, existen también incumbencias que son horizontales, o sea que se ven reflejadas en todos los componentes del MVC. Entre ellas se tiene el tema de la seguridad. Toda la autorización, desde el acceso a directorios, páginas, controles, opciones del menú, servicios del negocio, está basado en reglas permitiendo así que las reglas de negocio no se escriban en el código y que un cambio en las mismas se aplique solo al fichero de configuración correspondiente sin afectar el código de la aplicación. Para esto se utiliza Seam Security Framework que permite estas ventajas gracias a su integración con el potente motor de reglas JBoss Rules.

Patrones de diseño

Para el desarrollo del sistema se utilizan los patrones de diseño **GRASP**, más conocidos como patrones generales para la asignación de responsabilidades. Destacan por su utilización en la propuesta de solución:

- ❖ **Experto:** es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo.
- ❖ **Creador:** el patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, almacena o maneja varias instancias de la clase.
- ❖ **Alta cohesión:** Expresa que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase.

- ❖ **Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

La aplicación de estos patrones en el sistema se ve reflejada con la asignación de las responsabilidades por clases para realizar solo las funcionalidades correspondientes a la información que las mismas contienen evitando así la sobrecarga a las mismas. Además se le da la responsabilidad de crear instancias de otras clases solamente a aquellas que contengan a las mismas. Por otra parte se modelan también clases controladoras, encargadas de realizar las operaciones del sistema y el diseño de la aplicación permite la interacción entre las mismas sin afectar su reutilización y su correcto funcionamiento por separado.

3.2 Modelo de diseño

Para la fase de implementación de un sistema una de las principales entradas es el modelo de clases del diseño. Este tiene como objetivo fundamental la construcción de un modelo lógico del sistema que se desea implementar. Dicho modelo incluye elementos importantes para los desarrolladores como son los requisitos funcionales y no funcionales y las restricciones impuestas por el lenguaje de programación que se desea utilizar. Para garantizar una óptima representación del modelo de diseño, se realizan dos actividades fundamentales: diseñar la arquitectura y diseñar clases del diseño (identificar operaciones, atributos y relaciones).

3.2.1 Definición de elementos del diseño

Teniendo en cuenta que la propuesta de sistema es una aplicación web se definen tres estereotipos de clases fundamentales. Las páginas servidoras o Server Pages que tienen la tarea de construir o generar las páginas clientes o Client Page. Estas últimas tienen como función fundamental mostrar la información al usuario e interactuar con el mismo y solo pueden ser creadas por una única Server Page. Además, las Client Pages pueden presentar vínculos hacia otras páginas clientes. También se definen los Form o formularios, encargados de enviar los datos a las páginas servidoras para que sean procesados los pedidos. Los formularios están contenidos dentro de las Client Pages.

Para modelar los diagramas de paquetes y los de clases del diseño se utilizará la siguiente nomenclatura: Diagrama de paquetes: Diagrama de paquetes_<Nombre del módulo>, Diagramas de clases del diseño: DCD_<Nombre de la funcionalidad>.

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

En los diagramas de clases del diseño se encuentran además otros elementos de importancia como son los marcos de trabajo JSF 1.2, EJB 3.0, JasperReport, JPA, Hibernate 3.3 y Seam. Existen también un conjunto de clases entre las que se encuentran librerías, JavaScript, JSF (Servlets), Seam (Servlets), Bitácora, entre otras.

3.2.2 Diagramas de clases del diseño

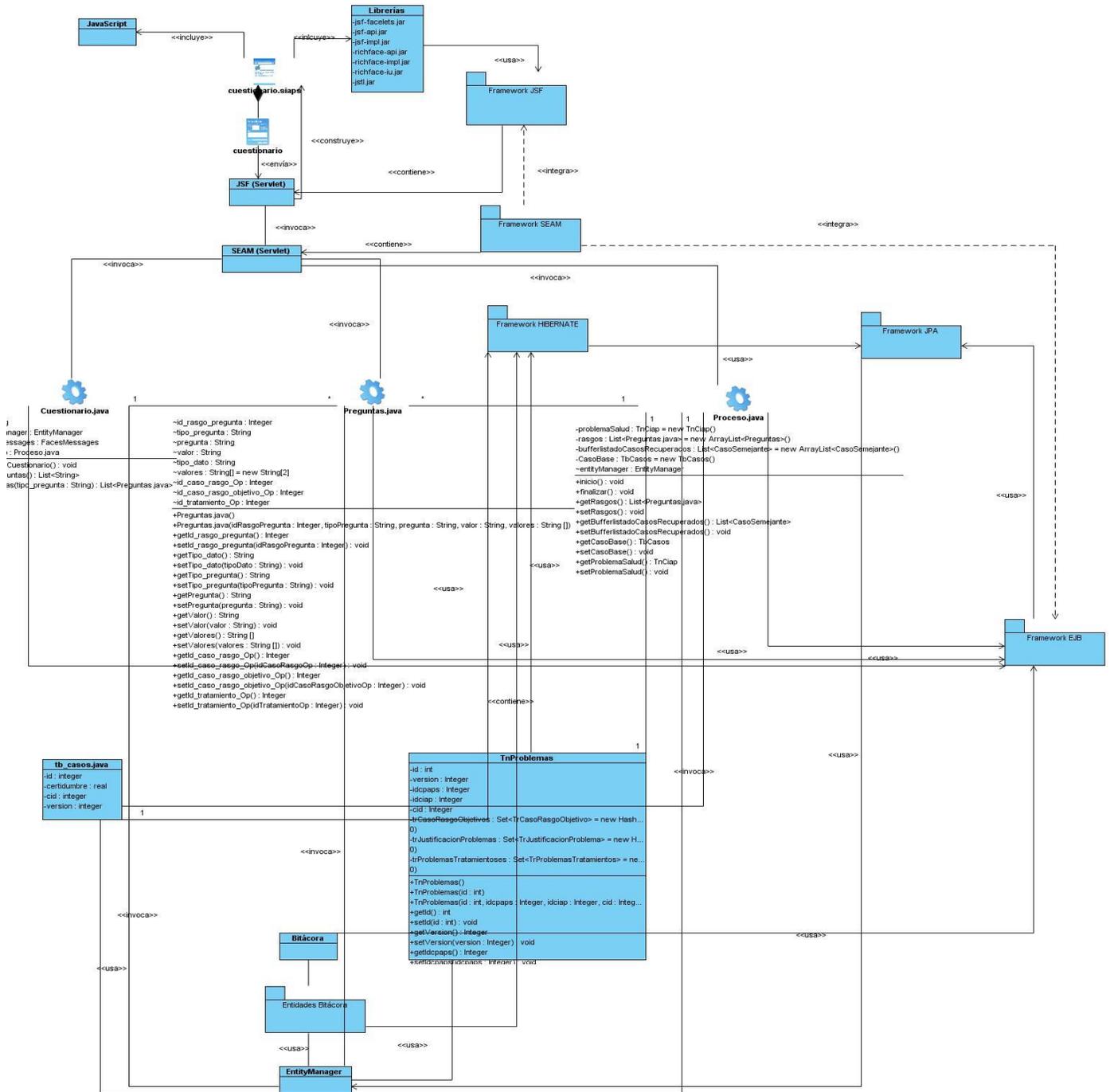


Figura 3.1 Diagrama de Clases del Diseño. Realizar Cuestionario.

3.2.3 Descripción de las clases

A continuación se describen las clases de la que se hace uso en el modelo de diseño correspondiente a la funcionalidad Realizar Cuestionario con el objetivo de comprender la estructura general que tendrá la solución propuesta en términos de clases de implementación.

La página cuestionario.siaps es la que permitirá recoger los síntomas o signos que el paciente presenta. La misma cuenta de un formulario y se ejecuta sobre un navegador haciendo posible la captura de datos que serán persistidos en la base de datos, además de mostrarle información útil al usuario. Dicha página está controlada por la clases CCProceso.java, CCPreguntas.java y CCCuestionario.java que son las encargadas de ejecutar los métodos necesarios para realizar las operaciones de inserción. Estas clases poseen un conjunto de validaciones en JavaScript que no permiten realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utilizan diferentes librerías basadas en el Framework JSF.

En la capa de datos se encuentran las clases entidad, las cuales serán persistidas por las clases servidoras para darle una respuesta a las páginas clientes, estas se encargan de proveer el mapeo con la base de datos. En el diagrama de diseño de la funcionalidad Realizar Cuestionario se encuentran las clases entidad: tb_casos.java, tn_problemas.java.

Las clases anteriormente mencionadas son ejecutadas del lado del servidor. Están representadas por las entidades que son un objeto del dominio de persistencia. Normalmente, las entidad representan una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. El estado de persistencia de cada una de estas entidades se representa a través de campos persistentes o propiedades persistentes. Estos campos o propiedades usan anotaciones para el mapeo de estos objetos en el modelo de base de datos. Hacen uso del Framework Hibernate y JPA.

Para una mejor comprensión del resto de las descripciones de clases para los diagramas de clases del diseño remitirse al expediente de proyecto [27]. Los restantes diagramas de clases del diseño se pueden encontrar en los anexos. Ver **Anexo #2**

En este capítulo se realizó la descripción de la arquitectura de la que se hará uso para implementar el sistema propuesto. Dicha arquitectura hace uso de un conjunto de patrones arquitectónicos y de diseño de los cuales se justificó su uso. Se presentaron los diagramas de clases del diseño y la descripción de cada uno de sus elementos con el objetivo de comprender la estructura de las clases de la solución propuesta para su posterior implementación.

IMPLEMENTACIÓN Y PRUEBA

El capítulo recoge toda la documentación del proceso de implementación de las funcionalidades identificadas en la descripción del diseño. El artefacto que se modela es el diagrama de despliegue. También se recogen los resultados de las pruebas realizadas a los diferentes componentes.

4.1 Implementación

4.1.1 Diagrama de despliegue

El diagrama de despliegue define la arquitectura física que presenta el sistema. Se usa para modelar detalladamente los nodos físicos y las comunicaciones existentes entre ellos. De ese mismo modo queda especificado qué hardware, sistema operativo, software de interfaces y soporte conformarán los componentes implementados.

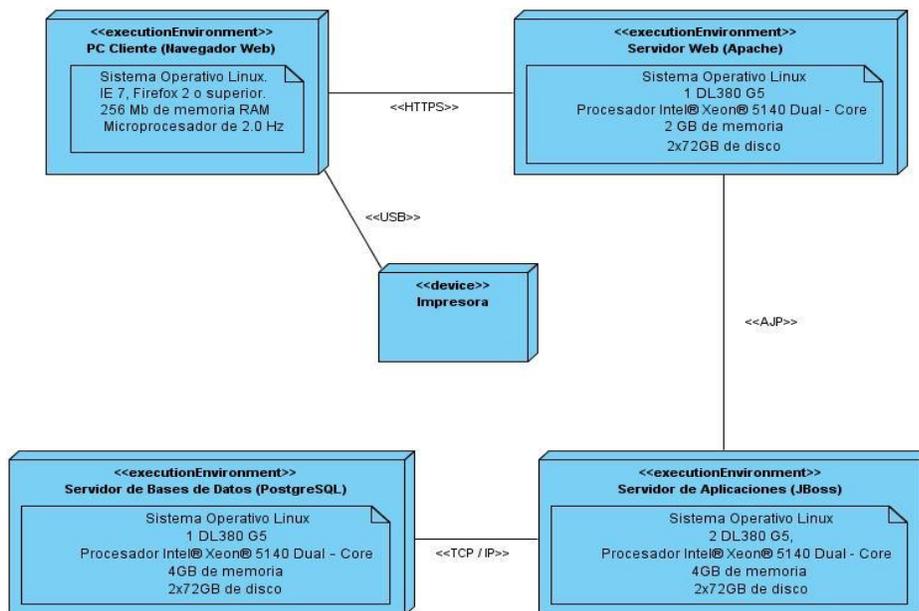


Figura 4.1 Diagrama de Despliegue.

Para la implantación y la utilización de la aplicación en un policlínico o consultorio de familia el usuario debe conectarse a ésta mediante una PC cliente utilizando un navegador web. Dicha PC interactuará mediante una conexión segura con el Servidor Web Apache, encargado de proveer el servicio de interfaz

al usuario y que actuará como fachada a Internet, al mismo tiempo que será el puente para entrar al clúster de servidor de aplicaciones que proporciona Jboss. Este servidor hospedará la solución integrada y garantizará la disponibilidad de la información que será almacenada en el Servidor de Datos PostgreSQL. Además el sistema debe contar con una impresora conectada a las estaciones de trabajo de los clientes para permitir realizar la impresión de los reportes del sistema.

4.1.2 Tratamiento de errores

En un sistema informático, las excepciones son situaciones excepcionales que pueden presentarse y que requieren un tratamiento especial en el mismo. Dichas situaciones pueden ocurrir dentro del programa durante su ejecución, y de ser así, podrían interrumpir el correcto funcionamiento del mismo.

Para el tratamiento de las excepciones en el sistema se usaron todas las potencialidades que brinda la plataforma Java en este aspecto. Con el uso de las diferentes tecnologías y la integración de las mismas es posible capturar y controlar situaciones anómalas en diferentes puntos de la aplicación. Se cuenta, en las páginas clientes, con un conjunto de componentes llamados validadores los cuales permiten establecer tipos de datos y formatos controlando que el envío de los activos al servidor sean los esperados.

Por otro lado, el marco de trabajo SEAM tiene un conjunto de excepciones predefinidas que, junto a la clase FacesMessages, permite el tratamiento de las situaciones erróneas desde las clases controladoras. Seam permite además, mediante el fichero de configuración page.xml, todo un flujo de navegación basado en excepciones.

Seguridad

Para que cualquier sistema informático sea calificado como seguro debe contar con un correcto equilibrio entre las siguientes características:

Integridad: La información sólo puede ser modificada por quien está autorizado y de manera controlada.

Confidencialidad: La información sólo debe ser legible para los autorizados.

Disponibilidad: La información debe estar disponible cuando se necesita.

Irrefutabilidad (No repudio): El uso y/o modificación de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar dicha acción.

En correspondencia con dichas características se plantean a continuación un conjunto de acciones que, llevadas a cabo, permitirá a los usuarios finales disfrutar de un software seguro:

- ❖ Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan.
- ❖ Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.
- ❖ Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.
- ❖ Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- ❖ Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude. Los mecanismos serán capaces de informar al personal autorizado sobre posibles irregularidades que den indicios sobre la introducción de información falseada.
- ❖ El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.
- ❖ Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el subsistema, este elemento ya no exista.
- ❖ Las informaciones médicas relacionadas con los pacientes y que vayan a ser intercambiadas con otros policlínicos por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas.
- ❖ El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

4.1.3 Estrategias de codificación. Estándares y estilos a utilizar

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. El mantenimiento del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el

rendimiento. Aunque la legibilidad y la mantenibilidad son el resultado de muchos factores, una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones.

Es fundamental mantener una buena indentación para lograr una estructura uniforme en los bloques de código así como para los diferentes niveles de anidamiento. Se recomienda dejar cuatro espacios en blanco desde la instrucción anterior para el inicio y fin de bloque `{}`. Lo mismo sucede para el caso de las instrucciones `if`, `else`, `for`, `while`, `do`, `while`, `switch`, `foreach`.

Deben escribirse comentarios al principio de cada clase y método brindando una breve descripción de los propósitos generales de cada funcionalidad. Los nombres de clases deben ser palabras completas, en mayúsculas y minúsculas, con la primera letra de cada palabra en mayúscula. Los nombres de clases deben ser simples y descriptivos, utilizando palabras completas y acrónimos o abreviaturas (a no ser que la abreviatura sea ampliamente conocida, como URL o HTML).

Los nombres de interfaces deberían seguir las mismas reglas indicadas para las clases. Los métodos deberían ser verbos, en mayúsculas y minúsculas, con la primera letra en minúscula, y la primera letra de cada una de las palabras internas en mayúscula. Todos los nombres de variables de instancia o de clase deben estar constituidos por palabras con la primera letra de la primera palabra en minúscula y la primera letra de las palabras internas en mayúscula. Los nombres de variables deben ser cortos y significativos. La elección de un nombre de variable, debe ser mnemotécnico, es decir, pensado para que un lector casual al verla comprenda su uso. Se deben evitar las variables de una sola letra, excepto en variables temporales de corto uso. Nombres comunes para este tipo de variables son: `i`, `j`, `k`, `m` y `n` para enteros; `c`, `d`, y `e` para caracteres. Con respecto al idioma se debe utilizar el español y las palabras no se acentuarán.

4.2 Pruebas de Software

Las pruebas de software son un elemento crítico para la garantía de la calidad de software, su objetivo principal es garantizar la calidad del producto desarrollado, esta etapa implica verificar la iteración de los componentes y su adecuada integración; verificar la correcta implementación de todos los requisitos e identificar y asegurar que los defectos encontrados se han corregido antes de entregar el producto.

4.2.1 Niveles de Prueba

El proceso de pruebas pasa por diferentes niveles los cuales se implementan de manera secuencial relacionándose con los productos de desarrollos estos niveles son:

- ❖ **Prueba de Unidad:** esta se centra en el módulo haciendo uso de la descripción del diseño detallado como guía, se prueban los caminos de control importantes con el fin de descubrir errores dentro del ámbito del módulo. La prueba de unidad se basa principalmente en las técnicas de prueba de caja blanca.
- ❖ **Prueba de Integración:** su objetivo principal es utilizar los módulos probados en el nivel anterior y construir una estructura de programa que esté de acuerdo con lo que especifica el diseño.

Existen dos maneras de realizar la integración, estas son:

- *Integración no incremental:* Se combinan todos los módulos por anticipado y se prueba todo el programa en conjunto.
 - *Integración incremental:* El programa se construye y se prueba en pequeños segmentos. En la prueba de integración el foco de atención es el diseño y la construcción de la arquitectura del software.
- ❖ **Prueba de Validación:** en esta etapa el software totalmente ensamblado se prueba como un todo para comprobar si cumple los requisitos funcionales y de rendimiento, facilidad de mantenimiento, recuperación de errores, etc.
 - ❖ **Prueba del Sistema:** el software ya validado se integra con el resto del sistema (rendimiento, seguridad, recuperación y resistencia)
 - ❖ **Prueba de aceptación:** el usuario comprueba en su propio entorno de explotación si lo acepta o no en su fase final.

4.3 Métodos de Prueba

4.3.1 Pruebas de Caja Negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, y es completamente indiferente al comportamiento interno y la estructura del programa. Estas pruebas se realizan con el objetivo detectar errores tales como funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación.

4.3.2 Diseño de Casos de Prueba

Escenario	Descripción	Cefalea	Tinnitus	Epistaxis	Dolor en la nuca	Náuseas	Ardor al orinar	Dolor en el pecho	Calambre	Sal	Descompensación PA	Hemoglobina Alta	Colesterol Alto	Electrocardiograma Alterado	Antecedentes	Alcohol	Fumador	Café	Ejercicios Físicos	Grasa	HTA Primaria	Edad	Peso	Raza	Lesión Órg. Diana	Hipertrofia Ventricular Izq.	Angiotonía	Proteinuria	Placas de aterosclerosis arterial	Respuesta del sistema	Flujo central	
Datos correctos	Se introducen los datos correctamente.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	30	45	N	V	V	V	V	V	V	No muestra mensaje porque los datos son correctos.	Insertar los datos correctos y Finalizar para añadir los datos.
Datos incorrectos	Se introduce al menos un valor de un campo de texto con caracteres incorrectos.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	I	V	I	V	V	A	E	7	V	I	V	V	V	V	Muestra el mensaje de información: "Datos Incorrectos".	Presionar el botón Finalizar.
Campos vacíos	Dejar campos de textos obligatorios en blanco.	V	V	V	V	V			V	V	V	V																			Muestra el mensaje de información: "Datos Incompletos".	Presionar el botón Finalizar.

4.4 Resultados Obtenidos

En este capítulo se relacionaron los principales elementos de la implementación del sistema. Se definió una propuesta para el despliegue para la representación estructural del sistema. Además se describieron las convenciones de código o estándares de codificación definidos para la implementación y se expusieron los distintos métodos usados para el tratamiento de errores en la aplicación. Se realizaron pruebas de funcionalidad al sistema mediante las pruebas de caja negra sustentadas en diseños de casos de prueba.

CONCLUSIONES

La realización del presente trabajo ha posibilitado cumplir con los objetivos propuestos, por lo que se pueden plantear las siguientes conclusiones:

Se profundizó en los conceptos asociados a la informatización del proceso de toma de decisiones para el diagnóstico de la HTA y que facilitan la comprensión de la investigación. Se realizó un estudio de los sistemas existentes vinculados al problema y de las tecnologías que serán utilizadas en el desarrollo del sistema propuesto.

Se seleccionaron los lenguajes de programación, la arquitectura, Sistema de Gestor de Bases de Datos, la metodología y las herramientas a utilizar, evaluándose cada una de ellos por sus características y ventajas que aportan facilidad para el desarrollo de la aplicación. Se realizó el análisis de los conceptos asociados al dominio de la investigación, se explicaron los mismos de forma detallada y se confeccionó el diagrama del modelo de dominio correspondiente.

Se describieron además los requerimientos funcionales del sistema que se propone y que demuestran las necesidades que se resolverán al desarrollarse el mismo. También se especificaron los requerimientos no funcionales necesarios para el correcto despliegue y funcionamiento del producto. Se realizó la descripción de la arquitectura de la que se hará uso para implementar el sistema propuesto. Dicha arquitectura hace uso de un conjunto de patrones arquitectónicos y de diseño de los cuales se justificó su uso. Se presentaron los diagramas de clases del diseño y la descripción de cada uno de sus elementos con el objetivo de comprender la estructura de las clases de la solución propuesta para su posterior implementación.

Se relacionaron los principales elementos de la implementación del sistema. Se definió una propuesta para el despliegue para la representación estructural del sistema. Además se describieron las convenciones de código o estándares de codificación definidos para la implementación y se expusieron los distintos métodos usados para el tratamiento de errores en la aplicación. Se realizaron pruebas de funcionalidad al sistema mediante las pruebas de caja negra sustentadas en diseños de casos de prueba.

RECOMENDACIONES

Las recomendaciones de la investigación están dirigidas a sugerir acciones para complementar el producto obtenido. Por lo que para el buen desempeño y puesta en marcha de la aplicación se hacen las siguientes recomendaciones:

- ❖ Agregar más información a la Base de Casos garantizando de esta manera que el sistema sea más robusto y disminuya la tasa de error referente a los casos recuperados.
- ❖ Resolver el problema de la inconsistencia en los Sistemas Basado en Casos.
- ❖ Agregar un lector de RSS para que los usuarios puedan visualizar las guías de HTA desde su sitio oficial.

REFERENCIAS BIBLIOGRÁFICAS

1. *Revista Investigación Operacional*. **Gutierrez Martínez, Iliana, Bello Pérez, Rafael E. y Tellería Rodríguez, Andrés. 2002.** 2, Santa Clara : s.n., 2002, Vol. 23.
2. Idem 1.
3. **Rodríguez Rivas, Migdalia. 2007.** Portales Médicos.com. *Hipertension arterial en la atención primaria de salud. Estrategia de intervención para su óptimo tratamiento.* [En línea] 14 de Junio de 2007. <http://www.portalesmedicos.com/publicaciones/articulos/551/1/-Hipertension-arterial-en-la-atencion-primaria-de-salud-Estrategia-de-intervencion-para-su-optimo-tratamiento.html>.
4. **Interna, Grupo Nacional de Medicina. 1981.** *Estado actual y perspectivas de desarrollo de la Medicina Interna.* La Habana : s.n., 1981.
5. *Revista médica de Chile. ¿Qué es Medicina Interna?* **Reyes B, Humberto. 2006.** 10, Santiago de Chile : s.n., 2006, Vol. 134.
6. Idem 4.
7. **Rigol Ricardo O, Pérez Carballás F, Perea Corral J, Fernández Sacasas J, Fernández Mirabal J. 1985.** *Medicina General Integral.* La Habana : Editorial Ciencias Médicas, 1985.
8. Carpeta Metodológica de Atención Primaria de Salud y Medicina Familiar. [En línea] <http://www.pdcorynthia.sld.cu/Documentos/biblioteca/carpeta%20metodologica%20aps.pdf>.
9. CAPITULO IV: SISTEMAS BASADOS EN EL CONOCIMIENTO. *Inteligencia Artificial.*
10. **Barite, Mario. 2000.** KODictionary. [En línea] 2000. http://www.eubca.edu.uy/diccionario/letra_s.htm.
11. Idem 7.
12. Idem 7.
13. Idem 7.
14. *Revista Cubana de Investigaciones Biomédicas.* **Guevara López, Miguel A., Rodríguez Rodríguez, Mario y González Pestano, Norma. 1997.** 2, Ciudad de la Habana : s.n., 1997, Vol. 1
15. Idem 7.
16. **Santaella Vallejo, Juan.** Un Sistema Experto: MYCIN. [En línea] <http://www.it.uc3m.es/jvillena/irc/practicas/estudios/MYCIN.pdf>.
17. Idem 7.

REFERENCIAS BIBLIOGRÁFICAS

18. **2009.** Pymes. Publicaciones y Medios Especializados. *Un innovador sistema de ayuda al diagnóstico clínico.* [En línea] Marzo de 2009.
<http://publicacionesymedios.net/articulo.php?ida=1331>.
19. **Rojo, J. Oscar. 2003.** AUGCyL. Introducción a los sistemas distribuidos. [En línea] 2003.
<http://www.augcyl.org/?q=glol-intro-sistemas-distribuidos>
20. Arquitectura de Software. Capítulo II. *Arquitectura de Software.*
21. Madeja. RichFaces. [En línea] <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces>.
22. **Alcantar Hernández, Fernando. 2008.** GestioPolis.com. *Mapeo entre clases persistentes utilizando el software Hibernate My Eclipse para conectar a una base de datos con XML.* [En línea] 6 de Junio de 2008. <http://www.gestiopolis.com/administracion-estrategia/hibernate-para-bases-de-datos-con-xml.htm>.
23. **Ingeman Rasmussen, Anders. 2006.** Osalt.com. *JBoss Application Server 4.2.* [En línea] 2006.
<http://www.osalt.com/jboss>
24. Idem 23.
25. Idem 23.
26. **Campos Cosme, Anny y Rodríguez Ronquillo, Reynier A. 2010.** *Informatización de la Historia de Salud Familiar de la Atención Primaria de Salud.* Ciudad de La Habana : s.n., 2010.

BIBLIOGRAFÍA

1. **Alcantar Hernández, Fernando. 2008.** GestioPolis.com. *Mapeo entre clases persistentes utilizando el software Hibernate My Eclipse para conectar a una base de datos con XML.* [En línea] 6 de Junio de 2008. <http://www.gestipolis.com/administracion-estrategia/hibernate-para-bases-de-datos-con-xml.htm>.
2. **Álvarez Martínez, David , y otros. 2006.** *Sistema de Asistencia Interactiva.* Madrid : s.n., 2006.
3. Arquitectura de Software. Capitulo II. *Arquitectura de Software.*
4. Asistencia Sanitaria Privada. Diccionario Médico. [En línea] http://www.medicinaprivada.net/diccionario_medico.php.
5. **Barite, Mario. 2000.** KODictionary. [En línea] 2000. http://www.eubca.edu.uy/diccionario/letra_s.htm.
6. **Campos Cosme, Anny y Rodríguez Ronquillo, Reynier A. 2010.** *Informatización de la Historia de Salud Familiar de la Atención Primaria de Salud.* Ciudad de La Habana : s.n., 2010.
7. CAPITULO IV: SISTEMAS BASADOS EN EL CONOCIMIENTO. *Inteligencia Artificial.*
8. Carpeta Metodológica de Atención Primaria de Salud y Medicina Familiar. [En línea] <http://www.pdcorynthia.sld.cu/Documentos/biblioteca/carpeta%20metodologica%20aps.pdf>.
9. **Corona Saenz , Tomás, Almaguer Vargas, Gustavo y Maldonado Torres , Ranferi.** Sistema Computarizado Experto en Diagnóstico Nutritional en Naranjo. [En línea] <http://redalyc.uaemex.mx/pdf/573/57318210.pdf>.
10. **Delgado Ramos, Ariel. 2008.** *Informatica en Salud 2009. Informatización en el Sistema Nacional de Salud de Cuba.* [En línea] 2008. <http://informatica2009.sld.cu/conferencias/informatizacion-en-el-sistema-nacional-de-salud-de-cuba/>.
11. DiagnosMD. [En línea] <http://www.diagnosmd.com/index.php>.
12. **González Naranjo, Alixandra y Expósito Rojas, Leonardo. 2010.** *Centro de Toma de Decisiones en el Sistema Integral para la Atención Primaria de la Salud.* Ciudad de La Habana : s.n., 2010.
13. **Gutiérrez Martínez, Iliana; Bello Pérez, Rafael E.; Tellería Rodríguez, Andrés, 2002.** UN SISTEMA BASADO EN CASOS PARA LA TOMA DE DECISIONES EN CONDICIONES DE

INCERTIDUMBRE. Departamento de Ciencia de la Computación, Universidad Central de Las Villas, Santa Clara, Cuba.

14. **Ingeman Rasmussen, Anders. 2006.** Osalt.com. *JBoss Application Server 4.2*. [En línea] 2006. <http://www.osalt.com/jboss>.
15. **Interna, Grupo Nacional de Medicina. 1981.** *Estado actual y perspectivas de desarrollo de la Medicina Interna*. La Habana : s.n., 1981.
16. **King, David y Harmon, Paul . 1988 .** *Sistemas Expertos: Aplicaciones de la Inteligencia Artificial en la Actividad Empresarial*. 1988 .
17. *Libro de Medicina Interna. Diagnóstico y Tratamiento. Parte II. Enfermedades del Siatema Cardiovascular.*
18. **Loaiza, Roger.** ¿Qué es la inteligencia artificial? [En línea] http://bvs.sld.cu/revistas/san/vol2_2_98/san15298.htm.
19. **Lozano Guzmán, Edith Aída. 2004.** GestioPolis.com. *Aplicación de sistemas de soporte a la decisión en telemedicina*. [En línea] Junio de 2004. <http://www.gestiopolis.com/canales2/gerencia/1/ddsmed.htm>.
20. **Lozano, Laura y Fernández, Javier .** *Razonamiento Basado en Casos: Una Visión General*.
21. Madeja. RichFaces. [En línea] <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces>.
22. **Piñera Tapia, P. 1998.** *Razonamiento basado en caso: una herramienta para el abordaje del paciente epiléptico con trastorno psiquiátrico*. 1998.
23. **2009. Pymes. Publicaciones y Medios Especializados. Un innovador sistema de ayuda al diagnóstico clínico.** [En línea] Marzo de 2009. <http://publicacionesymedios.net/articulo.php?ida=1331>.
24. **Rauch-Hindin, Wendy B.** *Aplicaciones de la inteligencia artificial en la actividad empresarial, la ciencia y la industria*.
25. *Revista Cubana de Investigaciones Biomédicas.* **Guevara López, Miguel A., Rodríguez Rodríguez, Mario y González Pestano, Norma. 1997.** 2, Ciudad de la Habana : s.n., 1997, Vol. 16.

26. *Revista Cubana de Investigaciones Biomédicas. DIAG, un sistema experto para el diagnóstico de anomalías craneofaciales.* **Guevara López, Miguel A., Rodríguez Rodríguez, Mario y González Pestano, Norma . 1997.** 2, Ciudad de la Habana : s.n., 1997, Vol. 16 .
27. *Revista Cubana de Medicina General Integral. Hipertensión arterial: comportamiento de su prevalencia y de algunos factores de riesgo.* **Hernández Cisneros, Freddie, y otros. 1996.** 2, Ciudad de La Habana : s.n., 1996, Vol. 12 .
28. *Revista Cubana de Medicina. Medicina Interna ¿Qué fuiste, Qué eres, Qué serás?* **Aldereguía Lima, Gustavo . 1999.** 1, Ciudad de la Habana : s.n., 1999, Vol. 38.
29. *Revista Investigación Operacional.* **Gutierrez Martínez, Iliana, Bello Pérez, Rafael E. y Tellería Rodríguez, Andrés. 2002.** 2, Santa Clara : s.n., 2002, Vol. 23.
30. *Revista médica de Chile. ¿Qué es Medicina Interna?* **Reyes B, Humberto. 2006.** 10, Santiago de Chile : s.n., 2006, Vol. 134.
31. **Rigol Ricardo O, Pérez Carballás F, Perea Corral J, Fernández Sacasas J, Fernández Mirabal J. 1985.** *Medicina General Integral.* La Habana : Editorial Ciencias Médicas, 1985.
32. **Rodríguez Armas, Gisela. 2002.** Mujeres. UN DIAGNÓSTICO INTELIGENTE. [En línea] 18 de Octubre de 2002. <http://www.mujeres.cubaweb.cu/articulo.asp?a=2002&num=106&art=13>.
33. **Rodríguez Rodríguez, Mario , Guevara López, Miguel Angel y Gónzalez Pestano, Norma.** Diag: Un prototipo inicial de sistema experto para el diagnóstico de las anomalías cráneos faciales. [En línea] http://bvs.sld.cu/revistas/mciego/vol1_01_95/a8_v1_0195.html.
34. **Rojo, J. Oscar. 2003.** AUGCyL. Introducción a los sistemas distribuidos. [En línea] 2003. <http://www.augcyl.org/?q=glol-intro-sistemas-distribuidos>.
35. **Samper Márquez, Juan José.** RED Científica. Introducción a los Sistemas Expertos. [En línea] [Citado el: 10 de Diciembre de 2010.] <http://www.redcientifica.com/doc/doc199908210001.html>.
36. **Santaella Vallejo, Juan.** Un Sistema Experto: MYCIN. [En línea] <http://www.it.uc3m.es/jvillena/irc/practicas/estudios/MYCIN.pdf>.
37. **Santander, Francisco. 2007.** Apuntes de Medicina. [En línea] 2007. [Citado el: 22 de Noviembre de 2010.] <http://apuntesdemedicina.awardspace.com/>.

38. Sappiens.com. *E. Inteligencia artificial*. [En línea]
<http://www.sappiens.net/sappiens/comunidades/progmini.nsf/0/157089e28379973541256a9200616cdc!OpenDocument&Click=>.
39. Somerset. Medical Center. [En línea] [Citado el: 22 de Noviembre de 2010.]
http://somerset.convertlanguage.com/somerset/enes/24/www_smchealthwise_com/body.cfm?id=476.
40. **San Miguel Carrillo, José Antonio . 2007.** *Introducción al Razonamiento Basado en Casos (CBR). Practica de Inteligencia Artificial II*. Valladolid : s.n., 2007.

GLOSARIO DE TÉRMINOS

Aplicación o Sistema Informático: Programas con los cuales el usuario final interactúa a través de una interfaz y que realizan tareas útiles para éste.

Cliente Servidor: Modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información.

Código Alfanumérico: Código que contienen datos alfabéticos (letras) y caracteres especiales (tales como símbolos de puntuación y matemáticos) en adición a números.

Concurrencia: Ejecución simultánea de dos o más actividades durante el mismo intervalo de tiempo.

Componente: Parte física y reemplazable de un sistema que se ajusta a, y proporciona la realización de, un conjunto de interfaces.

Deficiencia: Es toda pérdida o anomalía de una estructura o función psicológica, fisiológica o anatómica.

Dependencia: Relación semántica entre dos elementos, en la cual un cambio en uno puede afectar al otro.

Dominio: Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminología comprendidos por los practicantes de ese dominio.

Equipos Básicos de Salud: Binomio conformado por el médico y enfermera de la familia, que atiende una población geográficamente determinada, que puede estar ubicado en la comunidad, centros laborales o educacionales.

Interoperabilidad: Condición necesaria para que los usuarios (humanos o mecánicos) tengan un acceso completo a la información disponible. Entre las iniciativas recientes más destacadas para dotar a la Web de interoperabilidad se encuentran los servicios Web y la Web semántica.

Internet: Método de interconexión de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red (lógica) única.

Informática: Disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

Informatizar: Proceso de aplicar sistemas o equipos informáticos al tratamiento de la información.

Morbilidad: Es el estudio de los efectos de una enfermedad en una población en el sentido de la proporción de personas que enferman en un sitio y tiempo determinado.

Paquete: Mecanismo de propósito general para organizar elementos en grupos.

Policlínico: Unidad de salud donde se brindan servicios médicos a una población geográficamente determinada perteneciente al nivel asistencial de Atención Primaria de Salud.

Servicio: Unidad de software que encapsula alguna funcionalidad de negocio y proporciona estas a otros servicios a través de interfaces públicas bien definidas.

Servicio Web: es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Software: Conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Subsistema: Agrupación de elementos, de los que algunos constituyen una especificación del comportamiento ofrecido por los elementos contenidos.

Taxonomía: Clasificación u ordenación en grupos de cosas que tienen unas características comunes.

Unidad de Salud: Centro de trabajo que pertenece al Ministerio de Salud Pública (MINSAP).

ANEXOS

Anexo #1: Tabla de criterios de expertos para validar los datos de la base de datos.

Síntomas	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Promedio
cefalea	9	7	6	5	6.75
tinnitus	6	7	7	3	5.75
epistaxis	6	8	6	1	5.25
dolor en el pecho	9	8	5	2	6
sin del cuello	5	6	6	2	4.75
náuseas	3	5	4	1	3.25
ardor al orinar	2	5	1	1	2.25
calambre	5	4	2	2	3.25
asintomático	8	10	9	9	9
lesión órganos diana	8	6	7	8	
hipertrofia ventricular izquierda	6	7	5	9	
Malos Hábitos					
fumador	9	9	9	2	7.25
café	8	7	9	2	6.5
droga, alcohol	9	8	9	3	7.25
grasa	9	8	9	3	7.25
sal	9	9	9	2	7.25
Estilos de vida					
sedentarismo	8	9	8	5	7.5
Antecedentes familiares					
antecedentes	9	8	9	2	7
Exámenes complementarios					
hemoglobina alta	2	8	8	4	5.5
colesterol alto	8	8	8	3	6.75
electrocardiograma alto	9	8	8	9	8.5
descompensación de la presión arterial	10	9	9	9	9.25
Otros					
raza	7	7	8	10	8

sexo	5	4	7	8	6
peso	7	9	7	6	7.25
edad	8	7	6	6	6.75

Tabla 1.1 Criterios emitidos por los expertos.

Criterio 1: Edilberto Fernández Cumbá. Especialista MGI 1^{er} grado.

Criterio 2: Denis Deribet Thareux.

Criterio 3: Sarisbel Borroto Perello. Especialista MGI 2^{do} grado.

Criterio 4: Juan Mario Olivera Chirino. Especialista MI 1^{er} grado. Master en Urgencia y Emergencia.

Anexo #2: Diagramas de clases del diseño.

Descripción del diagrama de clases del diseño Adaptar Casos.

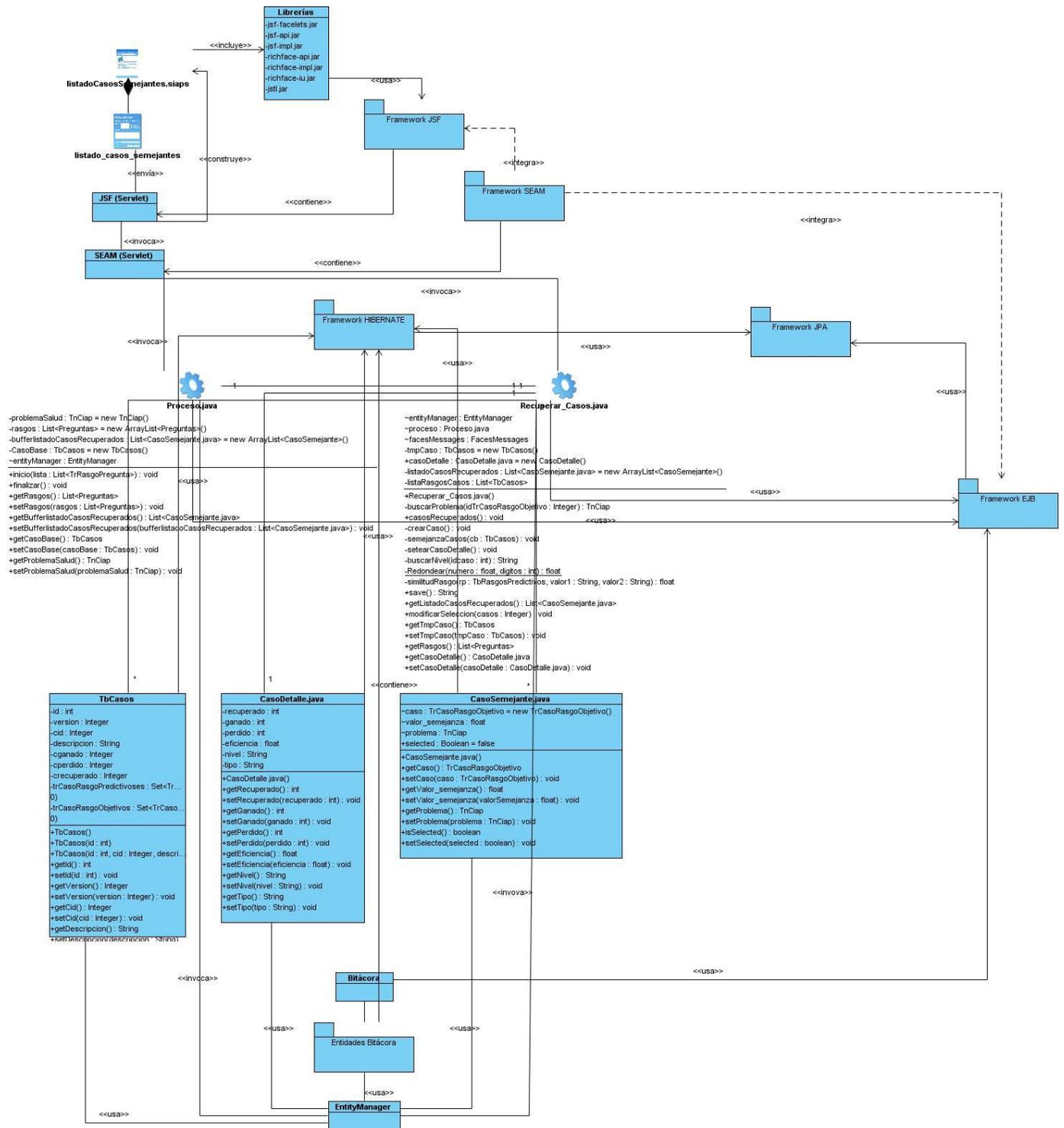


Figura 2.2 Diagrama de clases del diseño Recuperar Casos.

Descripción del diagrama de clases del diseño Índice de Masa Corporal.

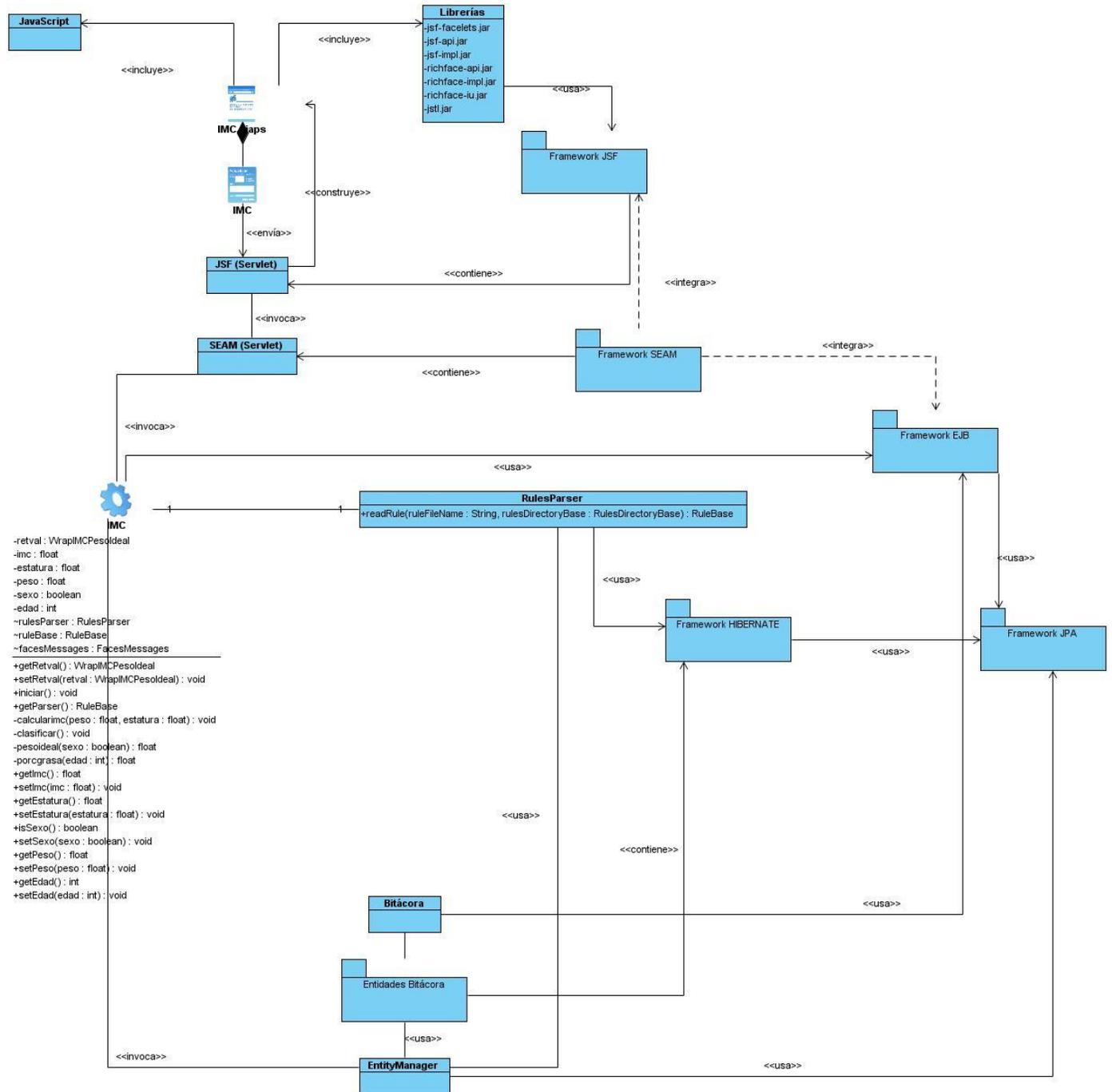


Figura 2.3 Diagrama de clases del diseño Índice de Masa Corporal.

Descripción del diagrama de clases del diseño Clasificar Casos.

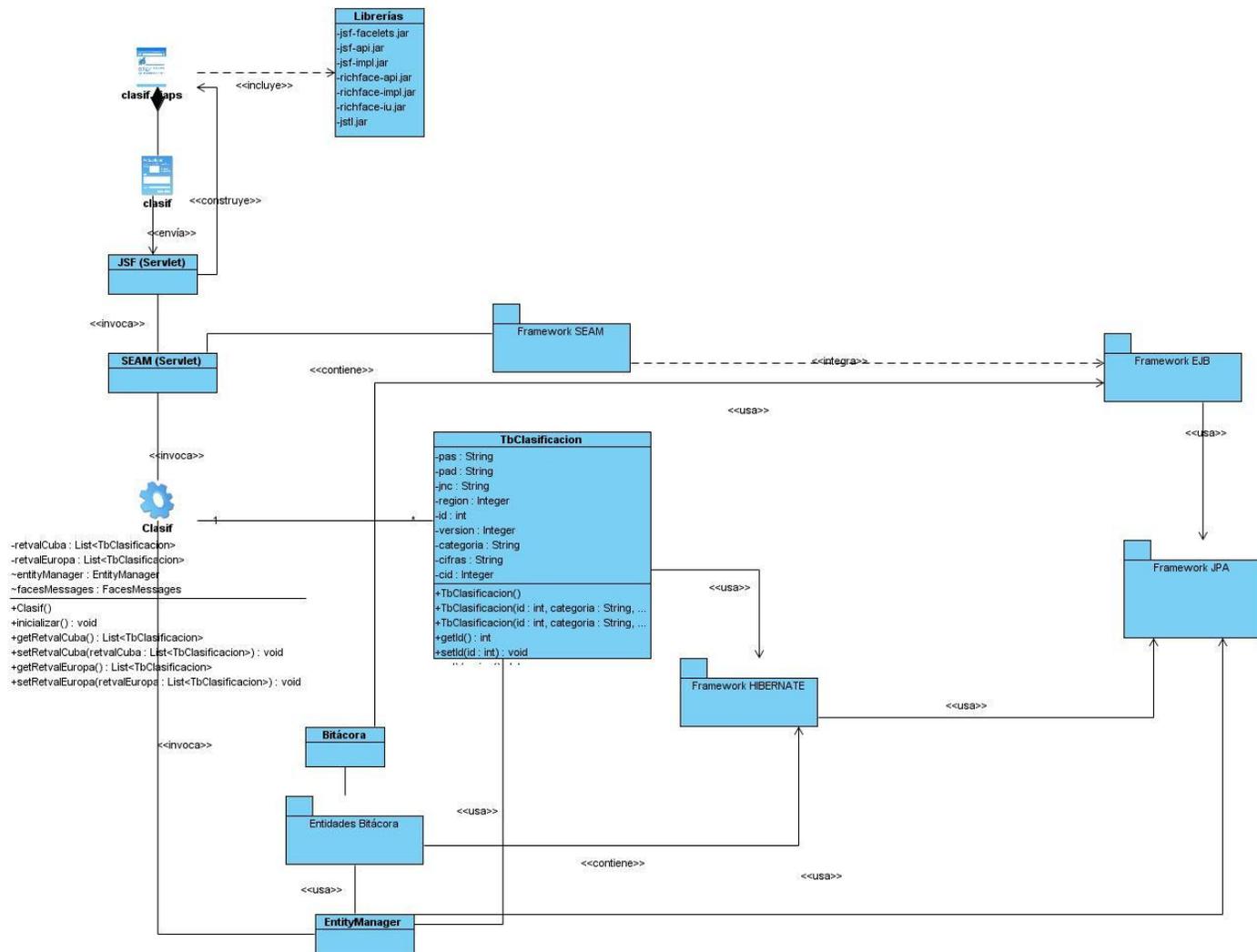


Figura 2.4 Diagrama de clases del diseño Clasificar Casos.

Anexo #3: Tabla de casos que inicializan la base de datos para el diagnóstico de la Hipertensión Arterial.

	Edad	Raza	Sexo	Peso	Síntomas	Antecedentes	Tratamiento	Factores de riesgo
1	76	N	f	55kg	cefalea	con antecedentes	captopril	fuma
2	63	N	f	63 kg	tinnitus, cefalea	con antecedentes	amlodipino	electrocardiograma alterado
3	42	B	m	72kg	asintomático	con antecedentes	atenolol	sedentaria
4	56	N	m	70kg	cefalea	sin antecedentes	enalapril	fuma
5	80	N	f	82kg	cefalea	con antecedentes	clortalidona	colesterol alto
6	62	N	f	75kg	cefalea, epistaxis	con antecedentes	captopril	fuma, alcohol

7	61	B	f	54kg	cefalea, mareos	sin antecedentes	captopril	sedentaria
8	59	N	f	57kg	cefalea, calambres	con antecedentes	enalapril	fuma y café
9	71	N	f	80kg	cefalea	con antecedentes	captopril	Fuma
10	54	B	m	57kg	tinnitus, cefalea	con antecedentes	amlodipino	sedentaria
11	38	B	m	80kg	asintomático	sin antecedentes	atenolol	sedentarismo
12	56	N	m	81kg	cefalea	sin antecedentes	enalapril	fuma
13	61	N	f	60kg	cefalea	con antecedentes	clortalidona	colesterol alto
14	88	N	f	55kg	cefalea, epistaxis	con antecedentes	captopril	fuma, alcohol
15	70	B	m	79kg	cefalea, epistaxis	con antecedentes	captopril	sedentaria
16	92	N	f	72kg	cefalea, mareos	con antecedentes	amlodipino	sedentaria
17	58	N	m	61kg	cefalea, calambres	sin antecedentes	atenolol	sedentarismo
18	42	N	m	70kg	cefalea epistaxis	sin antecedentes	enalapril	fuma y café
19	21	N	f	61kg	cefalea, calambres	con antecedentes	captopril	sedentarismo
20	39	N	f	70kg	tinnitus, cefalea	sin antecedentes	amlodipino	Fuma
21	26	N	f	63kg	asintomático	con antecedentes	atenolol	sedentarismo
22	41	N	m	59kg	cefalea	con antecedentes	enalapril	fuma
23	59	N	m	73kg	cefalea	sin antecedentes	clortalidona	colesterol alto
24	31	N	m	84kg	cefalea, epistaxis	con antecedentes	captopril	fuma, alcohol
25	63	N	f	57kg	cefalea, epistaxis	sin antecedentes	captopril	Fuma
26	65	N	m	78kg	cefalea, mareos	sin antecedentes	amlodipino	fuma, alcohol
27	52	B	m	89kg	cefalea, calambres	con antecedentes	atenolol	sedentarismo
28	45	N	m	93kg	cefalea epistaxis	con antecedentes	enalapril	fuma y café
29	45	N	m	62kg	cefalea, mareos,	sin antecedentes	captopril	fuma, obesidad
30	52	N	m	55kg	cefalea, mareos	con antecedentes	amlodipino	sedentarismo, obesidad
31	35	N	f	79kg	cefalea, mareos,	sin antecedentes	atenolol	fuma ,obesidad
32	40	B	m	70kg	cefalea, mareos,	con antecedentes	enalapril	obesidad
33	33	B	m	61kg	cefalea, mareos	sin antecedentes	clortalidona	obesidad
34	50	B	f	70kg	cefalea, mareos	con antecedentes	captopril	sedentarismo, fuma, obesidad
35	46	B	f	61kg	cefalea, mareos	sin antecedentes	captopril	sedentarismo, fuma, obesidad
36	25	B	m	76kg	cefalea	con antecedentes	enalapril	fuma, obesidad
37	46	B	f	42kg	cefalea, mareos	con antecedentes	captopril	sedentarismo, fuma, obesidad
38	46	B	m	59kg	mareos, cefalea	sin antecedentes	amlodipino	sedentarismo, obesidad
39	49	B	m	43kg	cefalea, mareos,	con antecedentes	clortalidona	sedentarismo, obesidad, fuma
40	61	B	f	60kg	cefalea, calambres	con antecedentes	clortalidona	sedentarismo
41	88	B	f	55kg	cefalea, epistaxis	con antecedentes	captopril	fuma, alcohol
42	70	B	m	79kg	tinnitus, cefalea	con antecedentes	captopril	Fuma

43	92	B	f	63kg	asintomático	con antecedentes	amlodipino	sedentarismo, obesidad, fuma
44	26	B	m	79kg	cefalea, calambres	sin antecedentes	atenolol	sedentarismo
45	42	B	m	70kg	cefalea epistaxis	sin antecedentes	enalapril	fuma y café
46	26	B	f	63kg	cefalea, calambres	con antecedentes	captopril	sedentarismo
47	70	B	f	79kg	tinnitus, cefalea	sin antecedentes	amlodipino	sedentarismo
48	26	B	f	63kg	asintomático	con antecedentes	atenolol	sedentarismo
49	70	B	m	59kg	cefalea	con antecedentes	enalapril	fuma
50	33	B	m	61kg	cefalea, mareos	sin antecedentes	amlodipino	obesidad
51	40	N	f	60kg	cefalea, tinnitus	con antecedentes	enalapril	obesidad, fuma, alcohol
52	38	N	f	74kg	cefalea, mareos, calambre	con antecedentes	captopril	obesidad
53	42	N	m	45kg	cefalea,	sin antecedentes	amlodipino	sedentarismo
54	75	N	f	52kg	cefalea, mareos	con antecedentes	clortalidona	fuma, alcohol
55	42	N	f	55kg	cefalea, mareos	con antecedentes	clortalidona	sedentarismo, obesidad
56	80	N	m	45kg	cefalea, mareos	con antecedentes	atenolol	fuma, café
57	39	N	m	45kg	cefalea	con antecedentes	enalapril	sedentarismo
58	36	N	f	62kg	cefalea, mareos	sin antecedentes	clortalidona	fuma, alcohol, obesidad
59	71	N	m	42kg	cefalea, mareos	sin antecedentes	captopril	sedentarismo
60	42	B	m	65kg	cefalea, mareos	con antecedentes	captopril	sedentarismo, obesidad
61	43	B	f	60kg	cefalea, mareos	con antecedentes	enalapril	fuma, café
62	40	N	f	56kg	cefalea, mareos, dolor en el cuello	con antecedentes	captopril	sedentarismo
63	40	N	f	45kg	cefalea, mareos	sin antecedentes	amlodipino	sedentarismo, obesidad
64	33	N	f	54kg	cefalea	con antecedentes	clortalidona	fuma, café
65	70	N	f	65kg	cefalea, mareos, dolor en el cuello	sin antecedentes	amlodipino	fuma, alcohol, obesidad
66	50	B	f	42kg	cefalea, mareos	sin antecedentes	atenolol	fuma, café
67	60	B	f	54kg	cefalea, mareos	con antecedentes	enalapril	fuma, alcohol, obesidad
68	40	N	m	42kg	cefalea, mareos	con antecedentes	captopril	sedentarismo
69	39	B	m	55kg	cefalea, mareos	con antecedentes	amlodipino	fuma, café
70	60	N	m	64kg	cefalea, mareos, dolor en el cuello	con antecedentes	atenolol	fuma, alcohol, obesidad
71	57	N	f	60kg	cefalea	con antecedentes	enalapril	sedentarismo, obesidad
72	43	N	m	62kg	cefalea	con antecedentes	captopril	fuma, café, obesidad
73	50	N	f	65kg	cefalea, mareos, dolor en el cuello	sin antecedentes	amlodipino	sedentarismo