

# TABLA DE CONTENIDO

	<b>Página.</b>
<b>1. INTRODUCCIÓN.....</b>	<b>09</b>
1.1. ANTECEDENTES.....	10
1.2. PLANTEAMIENTO DEL PROBLEMA.....	12
1.2.1. Problema Principal.....	12
1.2.2. Problemas Secundarios.....	12
1.3. SISTEMATIZACIÓN.....	13
1.3.1. Diagnóstico.....	13
1.3.2. Control del pronóstico.....	17
1.4. OBJETIVOS.....	17
1.4.1. Objetivo General.....	17
1.4.2. Objetivos Específicos.....	17
1.5. JUSTIFICACIÓN.....	18
1.5.1. Justificación Teórica.....	18
1.5.2. Justificación Práctica.....	18
1.5.3. Justificación Metodológica.....	19
1.6. ALCANCE Y LIMITACIONES.....	20
1.7. ESTUDIO DE FACTIBILIDAD.....	21
1.7.1. Factibilidad Técnica.....	21
1.7.2. Factibilidad Económica.....	25
1.7.2.1. Costos de inversión.....	25
1.7.2.2. Costo de equipos.....	26
1.7.2.3. Costo de Software.....	26
1.7.2.4. Costo de la Herramienta Desarrollada.....	27
1.7.2.5. Beneficios.....	27
1.7.3. Factibilidad Operativa.....	28

<b>2. MARCO DE REFERENCIA</b> .....	28
2.1. MARCO TEÓRICO.....	28
2.1.1. Sistemas Expertos y El Conocimiento.....	29
2.1.2. Definición de Sistema Experto.....	31
2.1.3. Diferencias entre un Sistema Experto y un Sistema Convencional.....	33
2.1.4. Motivaciones para el uso de Sistemas Expertos.....	34
2.1.5. Problemas y Limitaciones de los Sistemas Expertos.....	35
2.1.5.1. Limitaciones.....	35
2.1.6. Etapas de un Sistema Experto.....	36
2.1.7. Ciclo de vida de un Sistema Experto.....	37
2.1.8. Áreas de Aplicación Sistema Experto.....	37
2.1.9. Arquitectura de un Sistema Experto.....	39
2.1.10. Módulo de Explicaciones.....	39
2.1.11. Módulo de adquisición de conocimiento.....	40
2.1.12. Conocimiento y su relación con los datos.....	40
2.1.13. Componentes del conocimiento.....	42
2.1.14. Medios del conocimiento.....	43
2.1.15. Organización.....	44
2.1.16. Denominación.....	46
2.1.17. Restricciones.....	46
2.1.18. Otras relaciones.....	47
2.1.19. Clasificación del conocimiento.....	48
2.1.20. Técnicas de representación El Conocimiento.....	49
2.1.21. Weka y Redes Neuronales.....	50
2.1.21.1 Ficheros .arff.....	51
2.1.21.2. Introducción A Weka.....	53
2.1.21.3. Datos de entrenamiento.....	54
2.1.21.4. Clasificadores.....	55
2.1.21.5. Use training set.....	56
2.1.22. Multilayer Perceptron.....	56
2.1.22.1. Tipos.....	57
2.1.22.2. Historia.....	57
2.1.22.3. Características.....	58
2.1.22.4. Limitaciones.....	58

2.1.22.5. Aplicaciones.....	59
2.1.23. Redes Neuronales.....	59
2.1.23.1. Componentes de una Red Neuronal.....	61
2.1.23.2. La Neurona.....	61
2.1.24. Función de Integración.....	62
2.1.24.1. Función de Transferencia o Activación.....	62
2.1.24.1.1. Función de transferencia lineal (purelin).....	62
2.1.24.1.2. Función de transferencia sigmoidal (logsig)...	63
2.1.25. Red de Retro-Propagación.....	64
2.1.25.1. Arquitectura de una Red de Retro-propagación.....	65
2.1.26. Marco conceptual.....	66
2.1.26.1. Base de conocimiento.....	67
2.1.26.2. Base de hechos ( <u>Memoria</u> de trabajo).....	67
2.1.26.3. Motor de inferencia.....	67
2.1.26.4. Subsistema de Explicación.....	68
2.1.26.5. Interfaz de Usuario.....	68
2.1.26.6. Componentes de un Sistema Experto.....	68
2.1.26.7. El motor de inferencia.....	69
2.1.26.8. La interface.....	69
2.1.26.9. Módulo de Comunicaciones.....	70
2.1.27. El Modelo Entidad-Relación.....	70
2.1.28. BD. Atributos.....	70
2.1.29. MySQL.....	71
2.1.30. Java.....	72
2.1.31. Ficheros .arff.....	76
2.1.31.1. Preparación de los Datos.....	76
2.1.32. Weka.....	78
2.1.33. Multiplayer Perceptron.....	78
2.1.34. Marco Espacial.....	79
<b>3. ESTRATEGIA METODOLOGICA.....</b>	<b>79</b>
3.1. METODOLOGÍA DE INVESTIGACION.....	79
3.2. METODOLOGÍA INFORMÁTICA.....	81

<b>4. RESULTADOS</b> .....	85
4.1. REQUERIMIENTOS FUNCIONALES DEL NEGOCIO.....	85
4.2. DETERMINACIÓN REQUERIMIENTOS DEL SISTEMA.....	90
4.2.1. Requerimientos Funcionales.....	90
4.3. HISTORIA DE USUARIO.....	102
4.4. TAREA DEL MODELO DE SINTOMATOLOGÍA.....	102
4.5. PROTOTIPADO DE PANTALLA INGRESO DE SÍNTOMAS.....	104
4.6. BASE CONOCIMIENTO DEL SISTEMA.....	105
4.7. TAREA DEL DESARROLLO BASE DE CONOCIMIENTO DEL SISTEMA.....	106
4.8. DESARROLLO DEL MODULO DE DIAGNOSTICO DE ENFERMEDADES.....	108
4.9. PROTOTIPOS DE PANTALLAS PARA MODULO DE DIAGNOSTICO DE  ENFERMEDADES.....	110
4.10. DESARROLLO PARA MÓDULO DE CONTRAINDICACIONES Y RECETA DE MEDICAMENTOS.....	112
4.11. PROTOTIPADOS DE PANTALLA DE CONTRAINDICACIONES.....	114
4.12. DESARROLLO PARA MÓDULO DE CONTRAINDICACIONES Y RECETA DE MEDICAMENTOS.....	115
4.13. PROTOTIPOS DE PANTALLAS PARA ADMINISTRACION DE MEDICAMENTOS.....	116
4.14. ARQUITECTURA.....	122
4.14.1. Construcción: Plan de pruebas.....	122
4.14.1.1. Arquitectura cliente servidor de Tres Capa.....	122
4.14.2. Transición: Pruebas.....	123
4.14.2.1. Modelo de pruebas.....	123

<b>5. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>125</b>
5.1. CONCLUSIONES.....	125
5.2. RECOMENDACIONES.....	126
<b>BIBLIOGRAFIA.....</b>	<b>127</b>
<b>ANEXOS.....</b>	<b>129</b>

## LISTA DE FIGURAS

	<b>Página.</b>
<b>Figura 1.</b> Diagnostico	16
<b>Figura 2.</b> Mapa conceptual de Sistema Experto	29
<b>Figura 3.</b> Etapas de un Sistema Experto	36
<b>Figura 4.</b> Ciclo de vida de un Sistema Experto	37
<b>Figura 5.</b> Arquitectura de un Sistema Experto	39
<b>Figura 6.</b> Módulo de Explicaciones	39
<b>Figura 7.</b> Conocimiento y su relación con los datos	41
<b>Figura 8.</b> Componentes del conocimiento	42
<b>Figura 9.</b> Medios del conocimiento	43
<b>Figura 10.</b> Clasificación del conocimiento	48
<b>Figura 11.</b> Multilayer Perceptron	56
<b>Figura 12.</b> Función de Transferencia Lineal (purelin)	62
<b>Figura 13.</b> Función de Transferencia Sigmoidal	64
<b>Figura 14.</b> Arquitectura de una Red de Retro-propagación	65
<b>Figura 15.</b> Prototipado de pantalla ingreso de síntomas	104
<b>Figura 16.</b> Diagnóstico de enfermedades	110
<b>Figura 17.</b> Posibles enfermedades-paciente	111
<b>Figura 18.</b> Prototipados de pantalla de contraindicaciones	114
<b>Figura 19.</b> Crear o editar medicamento	116
<b>Figura 20.</b> Busca de medicamento	117
<b>Figura 21.</b> Loguin Usuario	119
<b>Figura 22.</b> Ingreso al sistema- Pantalla principal	120
<b>Figura 23.</b> Ingreso al sistema	121

<b>Figura 24.</b>	Arquitectura cliente servidor de Tres Capas	122
<b>Figura 25.</b>	Usuario no registrado	123
<b>Figura 26.</b>	Usuario registrado	124

## LISTA DE DIAGRAMAS

<b>Diagrama 1.</b>	Caso de uso de negocio Diagnosticar Paciente	87
<b>Diagrama 2.</b>	Caso de uso de negocio Gestionar contraindicaciones	88
<b>Diagrama 3.</b>	Caso de uso de negocio Recetar medicamento	85
<b>Diagrama 4.</b>	Diagrama de Paquetes	93
<b>Diagrama 5.</b>	Diagrama de Casos de Uso General	97
<b>Diagrama 6.</b>	Modelo de Dominio	98
<b>Diagrama 7.</b>	Modelo Entidad-Relación / Base de Datos	99
<b>Diagrama 8.</b>	Diagrama de Despliegue	100
<b>Diagrama 9.</b>	Modelo de caso de uso	101

## LISTA DE TABLAS

**Página.**

<b>Tabla 1:</b>	Matriz FODA	13
<b>Tabla 2:</b>	Cuadro comparativo Base de Datos	22
<b>Tabla 3:</b>	Cuadro comparativo Lenguaje de Programación	24
<b>Tabla 4:</b>	Costo de equipos servidor	26
<b>Tabla 5:</b>	Costo de software	27
<b>Tabla 6:</b>	Costo de la herramienta desarrollada	27
<b>Tabla 7:</b>	Diferencias entre un Sistema Experto y un Sistema Convencional	33
<b>Tabla 8:</b>	Áreas de Aplicación	38

<b>Tabla 9:</b> Requerimientos funcionales del negocio	85
<b>Tabla 10:</b> Requerimientos no funcionales	86
<b>Tabla 11:</b> Requerimientos funcionales	90
<b>Tabla 12:</b> Requerimientos no funcionales	92
<b>Tabla 13:</b> Reconocimiento de casos de uso	94
<b>Tabla 14:</b> Desarrollo de Modulo de sintomatología para el paciente	102
<b>Tabla 15:</b> Desarrollo de la pantalla para el ingreso de síntomas al sistema	102
<b>Tabla 16:</b> Investigación de los principales síntomas a los que el sistema dará soporte, previamente validados por el médico especialista	103
<b>Tabla 17:</b> Desarrollo de Base conocimientos sistema experto	105
<b>Tabla 18:</b> Investigación Estructura archivos arff de weka, y determinación de la estructura de los archivos necesarios para el experto.	106
<b>Tabla 19:</b> Selección de enfermedades e Investigación de los síntomas y características para la codificación e la base de conocimiento.	106
<b>Tabla 20:</b> Codificación de archivos arff	107
<b>Tabla 21:</b> Codificación de archivos arff (obtener)	107
<b>Tabla 22:</b> Desarrollo de módulo para la lectura escritura de kis archivos arff	107
<b>Tabla 23:</b> Desarrollo del módulo de diagnóstico experto	108
<b>Tabla 24:</b> Comparar algoritmos de búsqueda inteligente soportados por weka	108
<b>Tabla 25:</b> Implantación del algoritmo dentro del sistema experto	109
<b>Tabla 26:</b> Implantación del algoritmo dentro del sistema experto (obtener)	109
<b>Tabla 27:</b> Desarrollo del módulo de manejo de contraindicaciones y recete de medicamentos	112
<b>Tabla 28:</b> Investigación de las contraindicaciones a ser tomadas en cuenta por el sistema	113
<b>Tabla 29:</b> Desarrollo módulo de contraindicaciones y medicamentos para la receta	113
<b>Tabla 30:</b> Desarrollo del módulo de gestión de medicamentos	115
<b>Tabla 31:</b> Desarrollo módulo de administración de medicamentos	115
<b>Tabla 32:</b> Desarrollo control acceso al sistema	118
<b>Tabla 33:</b> Desarrollo módulo de control de acceso al sistema	118

## LISTA DE ANEXOS

	<b>Página.</b>
<b>Anexo 1.</b> Manual del Módulo Sintomatología	129
<b>Anexo 2.</b> Elaboración de archivos arff de las enfermedades del cerco Epidemiológico	140
<b>Anexo 3.</b> Preguntas realizadas para el desarrollo de un sistema médico para el análisis de las diez enfermedades más comunes en la sangre	159
<b>Anexo 4.</b> Reglas de Inferencia	163

## 1. INTRODUCCIÓN

El presente proyecto toma en cuenta una área muy importante como es la medicina, es un espacio en donde se requiere de mucho entrenamiento para ser un especialista, además, cuando existe una amplia diversidad de enfermedades, los síntomas pueden ser confusos cuando se busca determinar rápidamente un diagnóstico oportuno, que puede significar la sobrevivencia o la muerte del paciente.

Se presenta un sistema experto de asistencia médica que es el encargado de tomar decisiones altamente especializadas con base en los conocimientos de expertos sobre un área en particular, para lo que se necesita información sobre los síntomas del paciente, condición general, historial clínico y resultados del laboratorio. Estos datos se obtienen a partir de una serie de preguntas, cada una de las cuales es determinada a partir de la respuesta anterior del paciente utilizando diversas reglas.

La rápida evolución de las tecnologías de software ponen a disposición del informáticos potentes herramientas para el desarrollo de sistemas expertos, frameworks que permiten el desarrollo de plataformas de aprendizaje automático y minería de datos los cuales pueden ser administrados con una alto nivel accesibilidad.

## 1.1. ANTECEDENTES

En la actualidad un sistema experto de asistencia médica es el encargado de tomar decisiones altamente especializadas con base en los conocimientos de expertos sobre un área en particular, por lo que los datos son almacenados de forma estructurada para su recuperación. Además de la capacidad de ofrecer soluciones sobre algún problema, incluye la explicación del porqué se llegó a determinadas medidas.

El origen de los sistemas expertos ocurre entre los años 1960 y 1970 en los Estados Unidos, cuando se desarrolla un sistema para resolución de problemas con base en los conocimientos almacenados a través de algoritmos y métodos de búsqueda en general. En 1960, se crea el primer sistema experto, DENDRAL, cuya rama de experiencia era la química, y permite identificar de manera rápida la estructura molecular: el sistema experto fue desarrollado a través del proyecto de programación heurística de la Universidad de Stanford. Posteriormente, surge MYCIN, cuyo dominio fue el diagnóstico de enfermedades contagiosas de la sangre, lo que da como resultado una lista de recomendaciones para su tratamiento.

La medicina es un área en donde se requiere de mucho entrenamiento para ser un especialista, además, cuando existe una amplia diversidad de enfermedades, los síntomas pueden ser confusos cuando se busca determinar rápidamente un diagnóstico oportuno, que puede significar la sobrevivencia o la muerte del paciente.

En este sentido, el sistema experto sustituye al especialista en un área dominada plenamente por el médico. La parte importante son los recursos que se refieren al conocimiento almacenado adquirido, ya sea con la ayuda de un especialista o bien, a través del sistema que integra un módulo de aprendizaje, donde se construye su propio conocimiento.

Para realizar un diagnóstico se requiere información sobre los síntomas del paciente, condición general, historial clínico y resultados del laboratorio. Estos datos se obtienen a partir de una serie de preguntas, cada una de las cuales es determinada a partir de la respuesta anterior del paciente utilizando diversas reglas o a través de la experiencia (almacenada en la memoria del ser humano experto o bien, del sistema experto). Al principio las preguntas son generadas para reducir el número de enfermedades posibles planteando una hipótesis, y al final se realizan preguntas para soportar el diagnóstico.

Una de las formas comunes de llegar a un diagnóstico es mediante el interrogatorio al paciente, en este sentido, los sistemas expertos son los más aptos para esta tarea. Cuando el interrogatorio al paciente se realiza de forma correcta se podrá elegir el tratamiento adecuado para su problema.

La ventaja de los sistemas expertos es que al tener almacenado el conocimiento en medios electrónicos, nunca se deteriorará, por el contrario, con el módulo de aprendizaje se logran ingresar nuevas reglas para tratar nuevas enfermedades, lo que asegura también que al realizar la prueba en pacientes con los mismos síntomas se diagnostique de la misma forma. A pesar de la precisión de los

sistemas expertos, una parte que hace falta para poder respaldar los resultados o para poder llegar a ellas más rápido es la exploración física.

La rápida evolución de las tecnologías de software ponen a disposición del informáticos potentes herramientas para el desarrollo de sistemas expertos, frameworks que permiten el desarrollo de plataformas de aprendizaje automático y minería de datos los cuales pueden ser administrados con un alto nivel accesibilidad.

## **1.2. PLANTEAMIENTO DEL PROBLEMA**

### **1.2.1. Problema Principal**

“La falta de información necesaria, rápida y concisa sobre el diagnóstico de Enfermedades Hematológicas ha provocado deficiencia en la entrega oportuna del diagnóstico y tratamiento de las mismas en la Provincia de Loja”

### **1.2.2. Problemas Secundarios**

La falta de información de forma ordenada ha provocado la deficiencia en el diagnóstico de las enfermedades Hematológicas.

La inexistencia de un análisis de la información que posee el médico con respecto a las enfermedades hematológicas ha provocado la baja eficacia en la identificación de las mismas.

Inexistencia de un sistema para especificar los medicamentos a ser administrados.

### 1.3. SISTEMATIZACIÓN

#### 1.3.1. Diagnóstico

**Tabla 1:** Matriz FODA

<b>Fortalezas</b>	<b>Debilidades</b>
<ul style="list-style-type: none"> <li>▪ En nuestra ciudad existen grandes profesionales de la salud, expertos reconocidos</li> <li>▪ Gran cantidad de centros hospitalarios y casas de salud</li> <li>▪ Existencia de protocolos y procedimientos para diagnóstico manual</li> </ul>	<ul style="list-style-type: none"> <li>▪ El conocimiento de estos expertos no es compartido de manera adecuada, pocos son los que escriben tratados o libros</li> <li>▪ Ningún centro hospitalario o casa de salud de nuestro medio, hace uso de sistemas expertos para auxiliar a la correcta realización de los procesos</li> <li>▪ Pese a la existencia de protocolos establecidos así como de procedimientos de diagnóstico no existe una homogeneidad en su uso y generalización</li> </ul>

<b>Oportunidades</b>	<b>Amenazas</b>
<ul style="list-style-type: none"> <li>▪ Poder utilizar los conocimientos de médicos expertos de nuestro país para la construcción de bases de conocimiento que den soporte a sistemas expertos de diagnóstico</li> <li>▪ Que nuestros hospitales y casas de salud cuenten con sistemas expertos validados para diagnóstico oportuno.</li> <li>▪ La posibilidad de crear sistemas que diagnostiquen tomando en cuenta el mayor número de síntomas y en base a criterio aceptados por todos nuestros expertos de la medicina</li> </ul>	<ul style="list-style-type: none"> <li>▪ Existencia de un sinnúmero de criterios al momento de diagnosticar debido a que no se toman en cuenta las mismas premisas.</li> <li>▪ Retraso tecnológico en nuestros hospitales y casas de salud</li> <li>▪ Riesgo de dar un mal diagnóstico, de obviar síntomas y perjudicar al paciente</li> </ul>

**Elaborado por: Digna R. Jaramillo P.**

Luego de un análisis de la información recolectada entorno al tema de estudio del presente proyecto he podido determinar la existencia varias problemáticas las cuales enuncio a continuación acompañada de sus correspondientes causas:

- **Efecto:**

Falta sistema experto diagnostico enfermedades Hematológicas.

- **Causas:**

Principalmente se debe a la falta de líneas de investigación en lo que respecta a sistemas inteligentes y sus aplicaciones la solución de problemas de nuestro entorno social, este hecho se ve agravado primeramente a los altos costos de implementación de dichos sistemas, sin embargo existen soluciones de software libre pero la documentación está orientada a desarrolladores y es escasa.

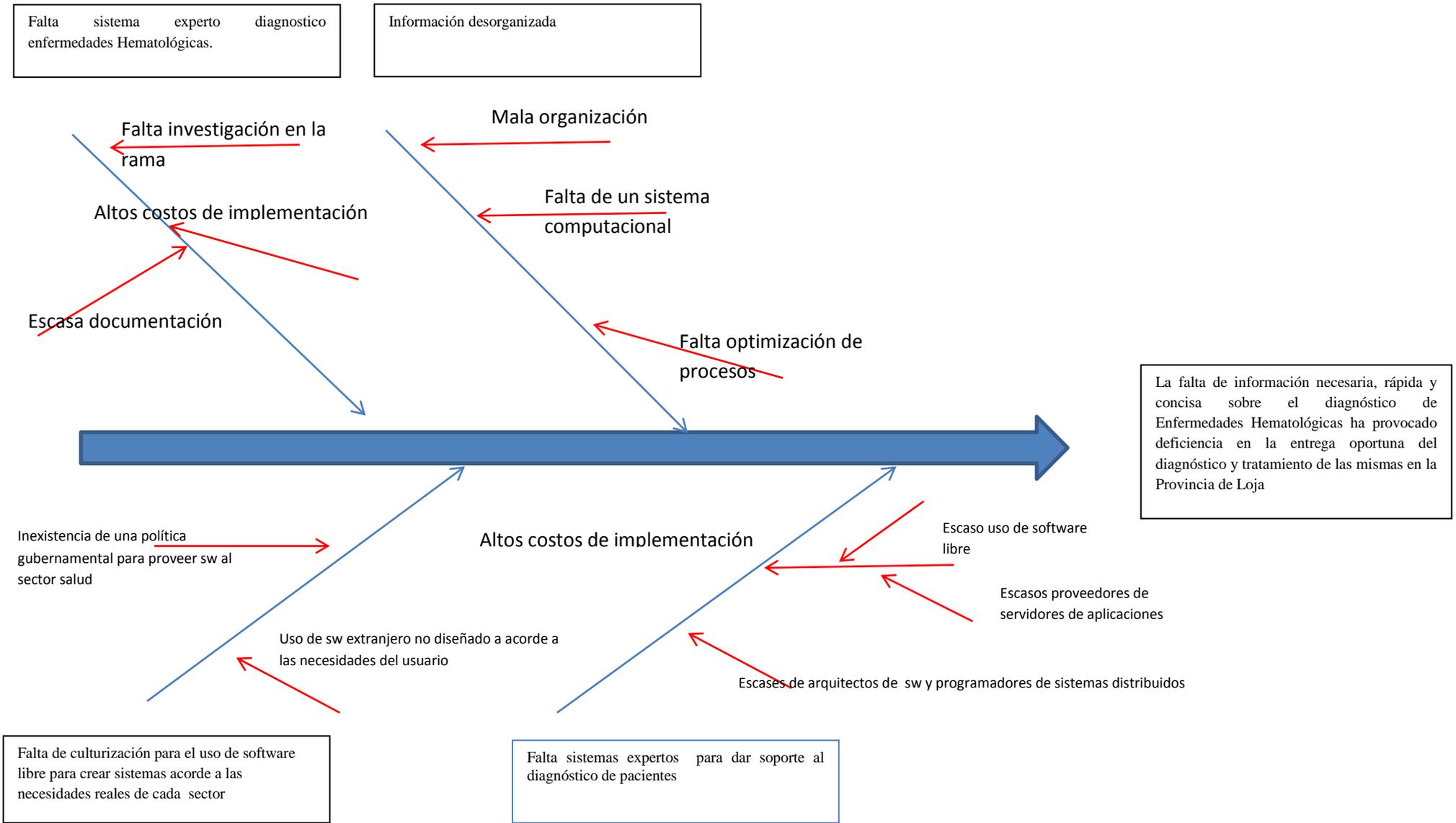
- **Efecto:**

Información desorganizada, inexistencia de bases de conocimiento en torno a diagnósticos exitosos y administración de medicamentos.

- **Causas:**

Se produce por la falta de un sistema computacional diseñado acorde a las necesidades del sector de la salud, el cual debe estar sujeto un diseño que contemple el perfeccionamiento de los procesos para la consecución de un manejo ágil, eficiente y eficaz de la información.

**Figura 1. Diagnostico**



**Elaborado por: Digna R. Jaramillo P.**

### **1.3.2. Control del pronóstico**

La implementación de un sistema experto para dar soporte al diagnóstico de enfermedades hematológicas así como de sugerencias de medicamento a ser administrados a pacientes el siguiente paso en las mejoras y actualización del sector de la salud de la provincia de Loja, ya que de continuar las cosas manejándose de forma manual o con software de propósito general ni podrá garantizarse en correcto almacenamiento de la información ni el acceso oportuno a la misma cuando sea necesario, así como una segunda opinión al experto humano para el diagnóstico individual de cada paciente.

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General:**

- Diseño e implementación de un Sistema Experto para diagnóstico de las enfermedades (enfermedades verificables en analíticas de sangre como: diabetes, anemia, hepatitis) en base al cuadro sintomatológico del paciente, y generar las posibles medicamentos activos para la receta.

### **1.4.2. Objetivos Específicos:**

- Recolectar la información necesaria de las enfermedades Hematológicas más importantes para el módulo de inferencia del sistema experto contrastando la información obtenida con el conocimiento del Doctor especialista.

- Diseñar e implementar un componente de software basado en un sistema experto para diagnóstico de enfermedades (enfermedades verificables en analíticas de sangre como: diabetes, anemia, hepatitis) en base al cuadro sintomatológico del paciente.

## **1.5. JUSTIFICACIÓN**

### **1.5.1. Justificación Teórica**

Gracias al conocimiento adquirido durante el transcurso de la carrera se puede desarrollar proyectos de investigación. La investigación en la Universidad Israel, se la asume como una función prioritaria dentro de cada Área. A través de ella, se propone desarrollar un proceso sostenido de rescate, validación y producción de conocimientos, que den respuestas efectivas a las complejas problemáticas del entorno local y regional; y, cuyos proyectos específicos apoyen los programas de formación y vinculación con la colectividad.

Este proyecto se justifica porque guarda relación con las líneas de investigación que existen en el Área de Estudio de la Informática, ya que se está resolviendo problemas existentes en la sociedad.

### **1.5.2. Justificación Práctica**

Los sistemas expertos se pueden considerar como el primer producto verdaderamente operacional de la inteligencia artificial. Son programas de

ordenador diseñados para actuar como un especialista humano en un dominio particular o área de conocimiento. En este sentido, pueden considerarse como intermediarios entre el experto humano, que transmite su conocimiento al sistema, y el usuario que lo utiliza para resolver un problema con la eficacia del especialista. El sistema experto utilizará para ello el conocimiento que tenga almacenado y algunos métodos de inferencia.

A la vez, el Usuario puede aprender observando el comportamiento del sistema u obtener todas las alternativas posibles en base a la sintomatología ingresada evitando de esta manera omitir resultados por error humano, y seleccionar el diagnóstico más oportuno.

### **1.5.3. Justificación Metodológica**

En todo proyecto de desarrollo de software, se hace necesario adoptar una metodología ágil de desarrollo, la cual por medio de una serie de actividades definidas permita llegar a culminar con éxito el presente proyecto. Luego de un análisis comparativo de las múltiples metodologías existentes, en función de la naturaleza y actividades que se deben realizar para cumplir con los objetivos del presente proyecto, se determinó de forma oportuna y conveniente adoptar al modelo de desarrollo **XP**, el cual se adapta a las necesidades de desarrollo del mismo, permitiendo cumplir con las etapas de análisis, diseño e implementación de forma ágil; minimizando el tiempo de desarrollo y permitiendo a través de las actividades propias de la misma obtener un software de calidad como solución a la problemática planteada.

## 1.6. ALCANCE Y LIMITACIONES

### ▪ **ALCANCE:**

El alcance del presente proyecto abarca la consecución de un sistema experto que como entrada toma el cuadro sintomatológico de un paciente determinado y basado en su base de conocimiento y mediante su motor de inferencias permite realizar un diagnóstico más acertado, a más de sugerir los posibles medicamentos para el tratamiento en base a contraindicaciones propias del paciente. Los sistemas de este tipo abren un mundo infinito de posibilidades para crear herramientas auxiliares a nuestros profesionales de la salud.

### ▪ **LIMITACIONES:**

El uso de tecnologías de software inteligente dentro del campo de la medicina en nuestro medio es una posibilidad aun no contemplada, existe una gran demanda que responde a la necesidad de contar con herramientas auxiliares a la labor del médico, sin embargo el conocimiento humano en esta área es gigantesco, es por ello que el presente proyecto se limita a dar soporte al diagnóstico de las principales enfermedades verificables en la sangre, aunque este proyecto se plantea como piloto para el desarrollo de sistemas completos al presente para así intentar abarcar un mayor número de casos clínicos comúnmente resueltos.

## 1.7. ESTUDIO DE FACTIBILIDAD

### 1.7.1. Factibilidad Técnica

El presente proyecto resulta factible desde un punto de vista técnico ya que existe a disposición todas las tecnologías de software que posibilitan su desarrollo, estas a más de pertenecer a la categoría de software libre cuentan con el respaldo de reconocidos grupos de desarrollo y con la documentación necesaria para aprovecharlas al máximo.

#### a) Desarrollo

- Hardware

Como requerimientos de hardware el presente un computador personal

las siguientes características mínimas de hardware:

- Procesador con una frecuencia superior a 2.4 Gz,
- Memoria RAM de 2 Gb o más.
- Espacio mínimo de 10 Gb de disco duro.
  
- Software
- El lenguaje de programación java proporciona al sistema la versatilidad de ser multiplataforma, sin embargo se recomienda su uso bajo el sistema GNU/LINUX, debido a su estabilidad y por ser software libre.
- El Lenguaje de programación seleccionado es Java debido a la madurez del mismo y a la gran cantidad de librería y frameworks.
- Como servidor de base de datos empleamos Mysql Sever 5.0 debido a su estabilidad, potencia y por formar parte de la comunidad de software libre.

- Como motor de inferencia se ha seleccionado al proyecto Weka el cual pone a nuestro alcance varios algoritmos de busque heurística.

**Tabla 2:** Cuadro Comparativo Base de Datos

MY SQL	ORACLE	POSTGRES
<ol style="list-style-type: none"> <li>1. Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.</li> <li>2. Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.</li> <li>3. Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.</li> <li>4. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.</li> </ol>	<ol style="list-style-type: none"> <li>1. Oracle es el motor de base de datos relacional más usado a nivel mundial.</li> <li>2. Puede ejecutarse en todas las plataformas, desde una Pc hasta un supercomputador.</li> <li>3. Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.</li> <li>4. Oracle es la base de datos con más orientación hacia INTERNET</li> </ol> <p><b>Desventajas</b></p> <ol style="list-style-type: none"> <li>1. El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras, en mi</li> </ol>	<ol style="list-style-type: none"> <li>5. Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).</li> <li>6. Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.</li> <li>7. Tiene la capacidad de comprobar la integridad</li> </ol>

<p style="text-align: center;"><b>Desventajas</b></p> <ol style="list-style-type: none"> <li>1. Carece de soporte para transacciones, rollback's y subconsultas.</li> <li>3. El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.</li> <li>4. No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.</li> </ol>	<p>opinión.</p> <ol style="list-style-type: none"> <li>2. Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes.</li> <li>3. Un Oracle mal configurado puede ser desesperantemente lento.</li> <li>4. También es elevado el coste de la formación, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.</li> </ol>	<p>referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.</p> <p style="text-align: center;"><b>Desventajas</b></p> <ol style="list-style-type: none"> <li>1. Consume gran cantidad de recursos.</li> <li>2. Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.</li> <li>3. Es de 2 a 3 veces más lento que MySQL.</li> </ol>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: [www.todoexpertos.com](http://www.todoexpertos.com)

Elaborado por: Digna R. Jaramillo P.

**Tabla 3:** Cuadro Comparativo Lenguaje de Programación

PHYTON	.NET	JAVA
<ol style="list-style-type: none"> <li>1. Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación interactiva y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.</li> <li>2. Python usa tipado dinámico y conteo de referencias para la administración de memoria.</li> <li>3. Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado ligadura dinámica de métodos).</li> <li>4. Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz</li> </ol>	<ol style="list-style-type: none"> <li>1. .Net es superior en programación web, pero... no es multiplataforma.</li> <li>2. Fácil de aprender. Es fácil aprender la programación punto net para los desarrolladores debido a que tiene una documentación muy completa y soporte de ayuda.</li> <li>3. Movilidad. Las aplicaciones pueden ser ejecutadas en diferentes tipos de dispositivos.</li> <li>4. Basado en objetos. Ofrece un modelo de desarrollo basado en objetos que utilizan un sistema común de datos y estos pueden ser reutilizados solo llamando al objeto a donde lo necesites.</li> <li>5. Herencia de clases, las clases incluidas en el framework pueden ser utilizadas independientemente del lenguaje que utilices, esto gracias al mecanismo de herencia propio de la programación orientada a objetos.</li> </ol>	<ol style="list-style-type: none"> <li>1. <b>Antigüedad:</b> Java tiene bastantes años más que .Net, por lo que es más sencillo encontrar código por internet, manuales y tutoriales</li> <li>2. <b>Licencias:</b> la máquina virtual de Java openjdk tiene licencia libre, y la oficial de Sun tiene CDDL (licencia casi de open source). Hay muchas máquinas virtuales comerciales y libres; desde la de IBM (comercial) hasta las libres de Kaffe o la de GNU. .Net es código cerrado totalmente.</li> <li>3. - <b>Entornos de desarrollo:</b> aunque Visual Studio 2005 es muy profesional, es sencillo y está en español, hay más entornos tanto para Linux como Windows para el lenguaje Java. Además, el sistema de plugins de Eclipse permite que este IDE vaya incrementando su funcionalidad cada vez</li> <li>4. En definitiva, ambos son muy similares pero Java tiene una experiencia difícil de superar y éste último se puede ejecutar sin problemas en Windows, GNU/Linux y MacOS.</li> </ol>

<p>programable. 5. Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional del Lidp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.</p>		
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

Fuente: [es.over-blog.com](http://es.over-blog.com)

Elaborado por: Digna R. Jaramillo P.

### 1.7.2. Factibilidad Económica

El presente capítulo tiene como objetivo realizar una cuantificación de los costos y beneficios que generará el presente proyecto.

En esta sección se analizará toda la información económica que tiene relación con el proyecto, es decir se detalla los costos y beneficios que proporcionará dicho proyecto. Se realizará la relación Beneficio – Costo para garantizar que el proyecto es rentable.

#### 1.7.2.1. Costo de inversión

Estos valores corresponden a costos por estudios, equipos e instalación hasta su funcionamiento. El presente proyecto será financiado por recursos propios.

### 1.7.2.2. Costo de equipos

En el desarrollo del proyecto, se necesitó 2 equipos de computación, con diferentes características, dependiendo de su utilidad en el proyecto. Las características de los equipos y sus costos se los detalla en la siguiente tabla.

**Tabla 4:** Costo de equipos servidor

EQUIPOS			
ITEM	CANTIDAD	DESCRIPCIÓN	COSTO (USD)
1	1	<b>Servidor Web:</b>	533,89
		1. Mainboard Intel 865 GLV Pentium IV bus de 800MHz DDR400/533	
		2. Memoria RAM de 1GB PC400	
		3. Disco duro 80 GB IDE 7200rpm	
		4. Procesador Intel Pentium IV 2,8GHz	
		5. Tarjeta de red 10/100 Mbps	
		6. Monitor LG 15'	
		7. Unidad de CD-rewritable	
		8. Teclado y Mouse	
2	1	<b>Servidor de Base de Datos:</b>	460,49
		1. Mainboard Intel 865 GLV Pentium IV bus de 800MHz DDR400/533	
		2. Memoria RAM de 512MB PC400	
		3. Disco duro 80GB IDE 7200rpm	
		4. Procesador Intel Pentium IV 2,8GHz	
		5. Tarjeta de red 10/100 Mbps	
		6. Monitor LG 15'	
		7. Unidad de CD-rewritable	
		8. Teclado y Mouse	
TOTAL			994,38*

\*Los costos de los equipos fueron provistos por la empresa Gootlek Cia. Ltda

### 1.7.2.3. Costo de Software

El proyecto está basado en la utilización de software libre, por lo que el costo de software es cero. Este costo está detallado en la siguiente tabla.

**Tabla 5:** Costo de software

<b>SOFTWARE</b>			
<b>ITEM</b>	<b>CANTIDAD</b>	<b>DESCRIPCIÓN</b>	<b>COSTO(USD)</b>
1	2	Sistema Operativo: Ubuntu 12.10	0,00
2	1	Herramienta de Programación: Netbeans 7.2	0,00
3	1	Administrador de BDD: MySQL 5.0: 1. MySQL Administrador 2. MySQL Query Browser	0,00
TOTAL			0,00

**1.7.2.4. Costo de la Herramienta Desarrollada**

La aplicación tiene un costo de 0 USD, como se muestra en la siguiente tabla, debido a que se la está desarrollando con software libre.

**Tabla 6:** Costo de la herramienta desarrollada

<b>HERRAMIENTA DESARROLLADA</b>			
<b>ITEM</b>	<b>CANTIDAD</b>	<b>DESCRIPCIÓN</b>	<b>COSTO(USD)</b>
1	1	Aplicación: "EXSYS"	0,00
TOTAL			0,00

**1.7.2.5. Beneficios**

Dadas las características del Proyecto, los beneficios que nos proporcionará son muy difíciles de cuantificar sin embargo se los ha agrupado a continuación:

- Al ser un software gratuito, el uso por parte de profesionales de la medicina es rentable.
- Ayuda a especialista en el análisis de la mayor cantidad de síntomas para obtener los diagnósticos más acertados ahorrando tiempo.
- Sirve como base para investigaciones en la misma línea.

### **1.7.3. Factibilidad Operativa**

Operativamente el presente proyecto tendrá un impacto positivo dentro del área de la salud, ya que es una herramienta de apoyo, contará con una buena aceptación ya que hoy por hoy la mayor parte de profesionales están habituados al uso de tecnología informática, satisfaciendo la demanda existente de este tipo de sistemas.

El sistema está orientado para convertirse en un apoyo para los especialistas de la medicina, por lo que su interface es intuitiva y de fácil manejo.

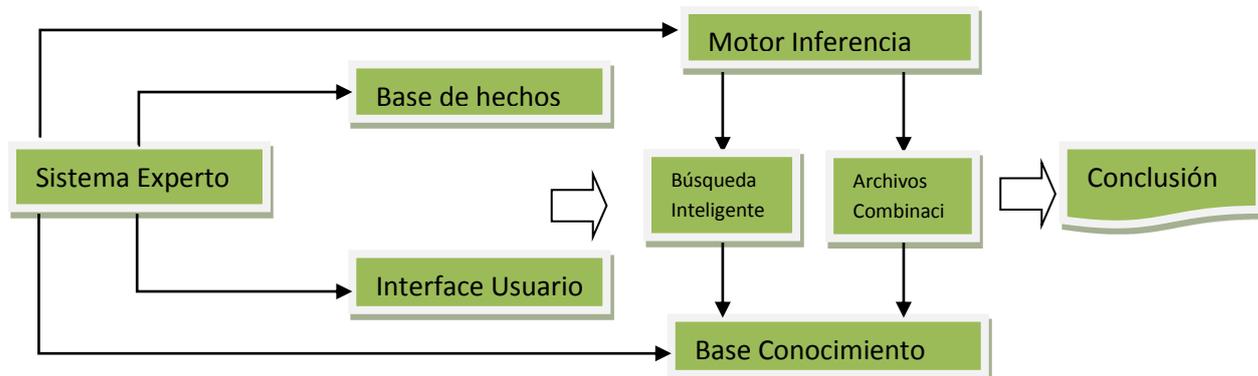
El sistema resulta ser escalable y de fácil mantenimiento, luego del periodo de vida útil de un sistema.

## **2. MARCO DE REFERENCIA**

### **2.1. MARCO TEÓRICO**

En el Marco teórico se puede empezar a definir con el siguiente mapa conceptual.

**Figura 2.** Mapa conceptual de Sistema Experto



Elaborado por: Digna R. Jaramillo P.

### 2.1.1. Sistemas Expertos y El Conocimiento

Sus inicios datan a mediados de los años sesenta. Durante esta década los investigadores Alan Newell y Herbert Simón desarrollaron un programa llamado GPS (General Problem Solver; solucionador general de problemas). Podía trabajar con criptoaritmética, con las torres de Hanoi y con otros problemas similares. Lo que no podía hacer el GPS era resolver problemas del mundo real, tales como un diagnóstico médico.

Es una aplicación informática capaz de solucionar un conjunto de problemas que exigen un gran conocimiento sobre un determinado tema. *Un sistema experto es un conjunto de programas que, sobre una base de conocimientos, posee información de uno o más expertos en un área específica.* Se puede entender como una rama de la inteligencia artificial, donde el poder de resolución de un problema en un programa de computadora viene del conocimiento de un dominio específico. Estos sistemas imitan las actividades de un humano para resolver

problemas de distinta índole (no necesariamente tiene que ser de inteligencia artificial). También se dice que un SE, se basa en el conocimiento declarativo (hechos sobre objetos, situaciones) y el conocimiento de control (información sobre el seguimiento de una acción).

Para que un sistema experto sea herramienta efectiva, los usuarios deben interactuar de una forma fácil, reuniendo dos capacidades para poder cumplirlo:

1. Explicar sus razonamientos o base del conocimiento: los sistemas expertos se deben realizar siguiendo ciertas reglas o pasos comprensibles de manera que se pueda generar la explicación para cada una de estas reglas, que a la vez se basan en hechos.
2. Adquisición de nuevos conocimientos o integrador del sistema: son mecanismos de razonamiento que sirven para modificar los conocimientos anteriores. Sobre la base de lo anterior se puede decir que los sistemas expertos son el producto de investigaciones en el campo de la inteligencia artificial ya que ésta no intenta sustituir a los expertos humanos, sino que se desea ayudarlos a realizar con más rapidez y eficacia todas las tareas que realiza.

Debido a esto en la actualidad se están mezclando diferentes técnicas o aplicaciones aprovechando las ventajas que cada una de estas ofrece para poder tener empresas más seguras. Un ejemplo de estas técnicas sería los agentes que tienen la capacidad de negociar y navegar a través de recursos en línea; y es por eso que en la actualidad juega un papel preponderante en los sistemas expertos.

### **2.1.2. Definición de Sistema Experto**

Es un software que imita el comportamiento de un experto humano en la solución de un problema. Pueden almacenar conocimientos de expertos para un campo determinado y solucionar un problema mediante deducción lógica de conclusiones.

Son SE aquellos programas que se realizan haciendo explícito el conocimiento en ellos, que tienen información específica de un dominio concreto y que realizan una tarea relativa a este dominio.

Programas que manipulan conocimiento codificado para resolver problemas en un dominio especializado en un dominio que generalmente requiere de experiencia humana.

Programas que contienen tanto conocimiento declarativo (hechos acerca de objetos, eventos y/o situaciones) como conocimiento de control (información acerca de los cursos de una acción), para emular el proceso de razonamiento de los expertos humanos en un dominio en particular y/o área de experiencia.

Software que incorpora conocimiento de experto sobre un dominio de aplicación dado, de manera que es capaz de resolver problemas de relativa dificultad y apoyar la toma de decisiones inteligentes en base a un proceso de razonamiento simbólico.

## ▪ **Características**

Para que un sistema actúe como un verdadero experto, es deseable que reúna, en lo posible, lo más importante de las características de un experto humano, esto es:

- Habilidad para adquirir conocimiento.
- Fiabilidad, para poder confiar en sus resultados o apreciaciones.
- Solidez en el dominio de su conocimiento.
- Capacidad para resolver problemas.

Dada la complejidad de los problemas que usualmente tiene que resolver un SE, puede existir cierta duda en el usuario sobre la validez de respuesta obtenida. Por este motivo, es una condición indispensable que un SE sea capaz de explicar su proceso de razonamiento o dar razón del por qué solicita tal o cual información o dato.

### 2.1.3. Diferencias entre un Sistema Experto y un Sistema Convencional

**Tabla 7:** Diferencias entre un Sistema Experto y un Sistema Convencional

Sistema Clásico	Sistema Experto
Conocimiento y procesamiento combinados en un programa	Base de conocimiento separada del mecanismo de procesamiento
No contiene errores	Puede contener Errores
No da explicaciones, los datos sólo se usan o escriben	Una parte del sistema experto consiste en el módulo de explicación
Es difícil efectuar cambios en los conocimientos programados	Los cambios en las reglas son fáciles
El sistema solo opera completo	El sistema puede funcionar con pocas reglas
Se ejecuta paso a paso	La ejecución usa heurísticas y lógica
Necesita información completa para operar	Puede operar con información incompleta
Representa y usa datos	Representa y usa conocimiento
Consiste en varias etapas de desarrollo comienza con la necesidad	Tiene varias fases de desarrollo pero no casi las aplica a todas

de un usuario o cliente	
Depende del problema y elige un tipo de modelo a trabajar, ya sea espiral, etc.	Aplica nuevos mecanismos
Conocimiento y lógica de procesos mezclados en un programa	Base de Conocimientos y Mecanismos de Inferencia separados
Necesitan información completa para operar	Deben ser más tolerantes para operar aún con alguna información desconocida
Manejan datos cuantitativos	Manejan datos cualitativos

Fuente: [www.books.google.com.ec](http://www.books.google.com.ec)

#### 2.1.4. Motivaciones para el uso de Sistemas Expertos

Con su ayuda, personas con poca experiencia pueden resolver problemas que requieren un "conocimiento formal especializado". Se pueden obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos. Estos sistemas razonan pero en base a un conocimiento adquirido y no tienen sitio para la subjetividad. Se ha comprobado que tienen al menos, la misma competencia que un especialista humano.

Su uso es especialmente recomendado en las siguientes situaciones:

- Cuando los expertos humanos en una determinada materia son escasos

- En situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas
- Cuando es muy elevado el volumen de datos que ha de considerarse para obtener una conclusión

## **2.1.5. Problemas y Limitaciones de los Sistemas Expertos**

### **2.1.5.1. Limitaciones**

Es evidente que para actualizar se necesita de reprogramación de estos (tal vez este sea una de sus limitaciones más acentuadas) otra de sus limitaciones puede ser el elevado costo en dinero y tiempo, además que estos programas son poco flexibles a cambios y de difícil acceso a información no estructurada.

Debido a la escasez de expertos humanos en determinadas áreas, los SE pueden almacenar su conocimiento para cuando sea necesario poder aplicarlo. Así mismo los SE pueden ser utilizados por personas no especializadas para resolver problemas. Además si una persona utiliza con frecuencia un SE aprenderá de él.

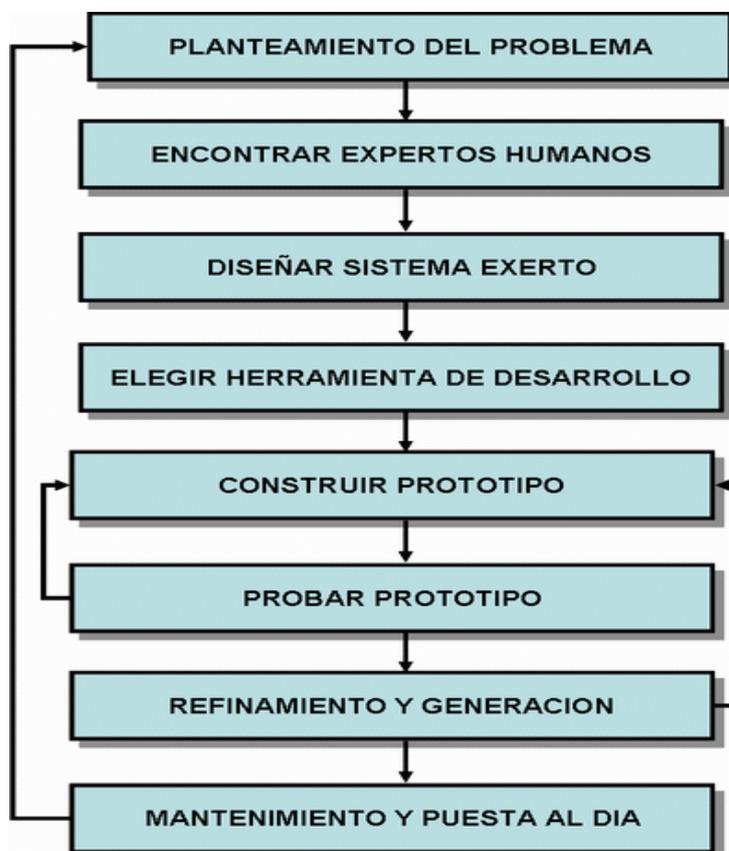
Por otra parte la inteligencia artificial no ha podido desarrollar sistemas que sean capaces de resolver problemas de manera general, de aplicar el sentido común para resolver situaciones complejas ni de controlar situaciones ambiguas.

El futuro de los SE da vueltas por la cabeza de cada persona, siempre que el campo elegido tenga la necesidad y/o presencia de un experto para la obtención de cualquier tipo de beneficio.

- Representación temporal de conocimiento.
- Razonamiento de sentido común.
- Límites de habilidad en el área de aplicación.
- Manipulación de conocimiento inconsistente.

### 2.1.6. Etapas de un Sistema Experto

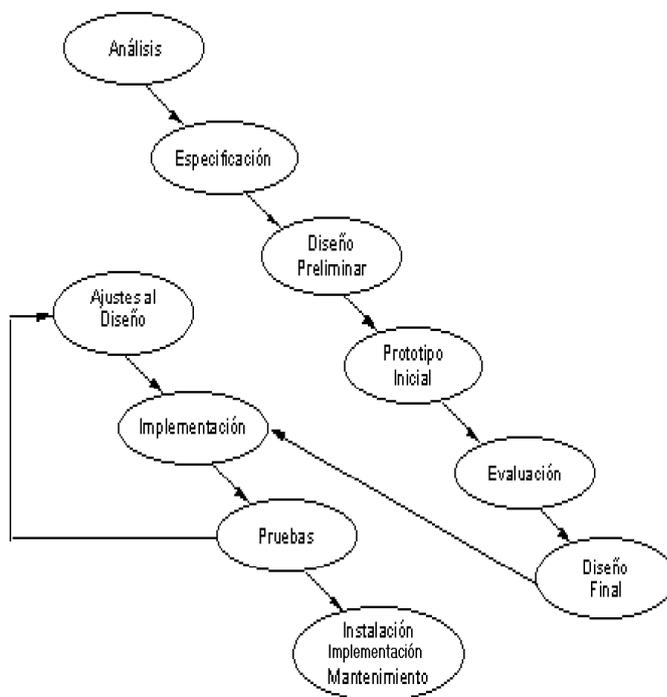
Figura 3. Etapas de un Sistema Experto



Fuente: [www.google.com.ec/imgres?=&Etapas+de+un+Sistema+Experto](http://www.google.com.ec/imgres?=&Etapas+de+un+Sistema+Experto)

### 2.1.7. Ciclo de vida de un Sistema Experto

**Figura 4.** Ciclo de vida de un Sistema Experto



Fuente: [www.google.com.ec/imgres?=&Cliclo+de+vida+de+un+Sistema+Exper](http://www.google.com.ec/imgres?=&Cliclo+de+vida+de+un+Sistema+Exper)

### 2.1.8. Áreas de Aplicación Sistema Experto

Sus principales aplicaciones se dan en las gestiones empresariales debido a que;

a) Casi todas las empresas disponen de un ordenador que realiza las funciones básicas de tratamiento de la información: contabilidad general, decisiones financieras, gestión de la tesorería, planificación, etc.

b) Este trabajo implica manejar grandes volúmenes de información y realizar operaciones numéricas para después tomar decisiones. Esto crea un terreno ideal para la implantación de los SE.

Además los SE también se aplican en la contabilidad en apartados como: Auditoria(es el campo en el que más aplicaciones de SE, se está realizando) Fiscalidad, planificación, análisis financiero y la contabilidad financiera.

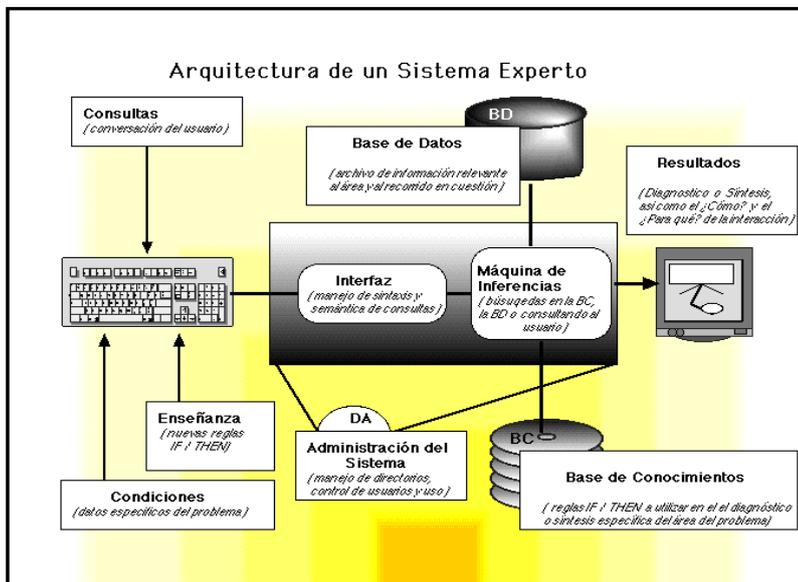
**Tabla 8:** Áreas de Aplicación

Militar	Informática	Telecomunicaciones
Química	Derecho	Aeronáutica
Geología	Arqueología	Agricultura
Electrónica	Transporte	Educación
Medicina	Industria	Finanzas y Gestión

Fuente:[www.google.com.ec/imgres?=&Areas+de+Aplicación+Sistema+Expert](http://www.google.com.ec/imgres?=&Areas+de+Aplicación+Sistema+Expert)

### 2.1.9. Arquitectura de un Sistema Experto

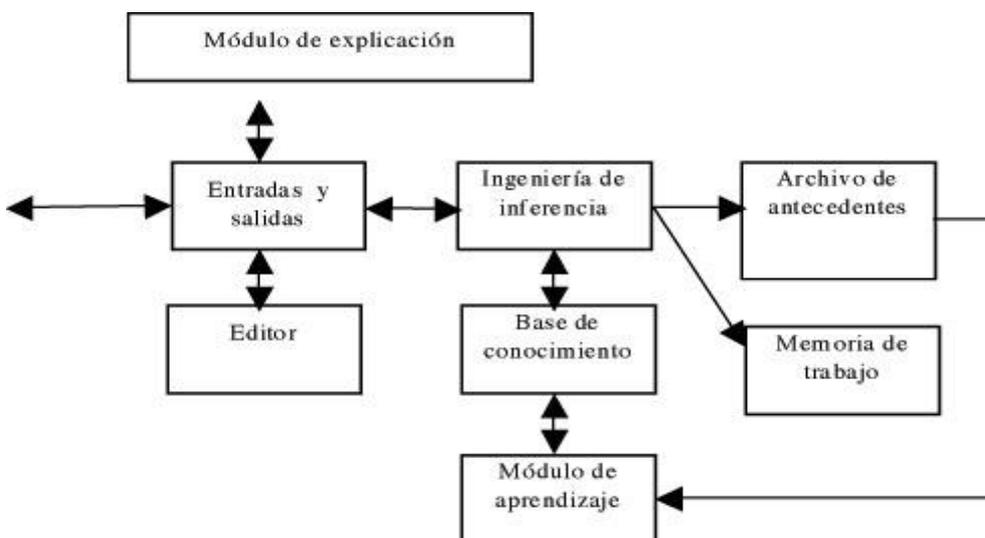
Figura 5. Arquitectura de un Sistema Experto



Fuente: [www.google.com.ec/imgres?=&Arquitectura+de+un+Sistema+Experto](http://www.google.com.ec/imgres?=&Arquitectura+de+un+Sistema+Experto)

### 2.1.10. Módulo de Explicaciones

Figura 6. Módulo de Explicaciones



Fuente: [www.google.com.ec/imgres?=&Modulo+de+Explicaciones](http://www.google.com.ec/imgres?=&Modulo+de+Explicaciones)

### **2.1.11. Módulo de adquisición de conocimiento**

El módulo de adquisición del conocimiento permite que se puedan añadir, eliminar o modificar elementos de conocimiento (en la mayoría de los casos reglas) en el sistema experto. Si el entorno es dinámico es muy necesario, puesto que, el sistema funcionará correctamente sólo si se mantiene actualizado su conocimiento. El módulo de adquisición permite efectuar ese mantenimiento, anotando en la base de conocimientos los cambios que se producen.

### **2.1.12. Conocimiento y su relación con los datos**

El conocimiento es una apreciación de la posesión de múltiples datos interrelacionados que por sí solos poseen menor valor cualitativo. Significa, en definitiva, la posesión de un modelo de la realidad en la mente.

Por su parte el conocimiento tiene como fuente la información misma, por lo que es un producto posterior y surge a partir de ella. Si la información se elabora a partir de objetos materiales para después aparecer como ente ideal; el conocimiento retoma ese ente ideal para construirse. Una distinción semejante la encontramos en Shera, quien expresa: “distinguimos entre información y conocimiento la información es el insumo del conocimiento, y siempre es recibida a través de los sentidos no puede haber conocimiento sin un conocedor.

**Figura 7.** Conocimiento y su relación con los datos



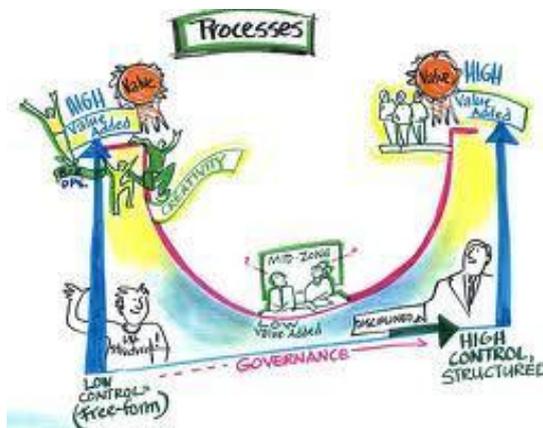
Fuente: [www.google.com.ec/imgres?=&Conocimiento+y+su+relacion+con+los+datos+de+un+sistema+experto](http://www.google.com.ec/imgres?=&Conocimiento+y+su+relacion+con+los+datos+de+un+sistema+experto)

Debemos tomar en cuenta que Información, Dato y conocimiento son muy diferentes.

- **Información:** Es cuando los datos ya han sido sometidos a un proceso el cual los relaciona entre si estos datos ya informan de algo real.
- **Datos:** Son valores recolectados del mundo real, estos no tienen contexto ni relación entre si o por si solos no representan una información valida
- **Conocimiento:** Se genera cuando la información es sometida al uso y hay experiencia gracias a la vivencia.

### 2.1.13. Componentes del conocimiento

**Figura 8.** Componentes del conocimiento



**Fuente:** [www.google.com.ec/imgres?=&Componentes+del+conocimiento+de+un+sistema+experto](http://www.google.com.ec/imgres?=&Componentes+del+conocimiento+de+un+sistema+experto)

**Sujeto:** Es el individuo conocedor. En él se encuentran los estados del espíritu en el que se envuelven o entran la ignorancia, duda, opinión y certeza.

Su función consiste en aprehender el objeto, esta aprehensión se presenta como una salida del sujeto fuera de su propia esfera, una invasión en la esfera del objeto y una captura de las propiedades de éste.

**Objeto:** Es aquello a lo que se dirige la conciencia, a la que se dirige la conciencia ya sea de una manera cognoscitiva, ya sea volitiva. Es lo que se percibe, imagina, concibe o piensa. Su función es ser aprehensible y aprehendido por el sujeto.

Los objetos pueden ser reales o irreales; llamamos real todo lo que nos es dado en la experiencia externa o interna o se infiere algo de ella. Los objetos irreales se presentan, por el contrario, como meramente pensados.

**Medio:** Generalmente hay ausencia de medios especiales, los instrumentos de trabajo son a la vez medios, del conocimiento de la realidad.

#### 2.1.14. Medios del conocimiento

**Figura 9.** Medios del conocimiento



**Fuente:** [www.google.com.ec/imgres?=&Medios+del+conocimiento+de+un+sistema+experto](http://www.google.com.ec/imgres?=&Medios+del+conocimiento+de+un+sistema+experto)

**La Experiencia interna:** Consiste en darnos cuenta de lo que existe en nuestra interioridad. Esta experiencia constituye una certeza primaria: en nuestro interior ocurre realmente lo que experimentamos.

**La Experiencia externa:** Es todo conocimiento o experiencia que obtenemos por nuestros sentidos.

**La Razón:** Esta se sirve de los sentidos, elabora los datos recibidos por ellos los generaliza y los abstrae, transformando la experiencia sensible y singular en conocimientos que valen en cualquier lugar y tiempo.

**La Autoridad:** Muchísimos conocimientos que poseemos nos llegan a través de la comunicación de personas que saben mucho sobre el tema, estas personas tienen autoridad científica y lo que divulgan o enseñan merece toda nuestra adhesión.

#### **2.1.15. Organización**

Los Sistemas de Organización del Conocimiento integran una serie de tecnologías que ayudan a crear, gestionar y visualizar modelos que muestren una perspectiva simplificada de los conceptos que forman determinado dominio y de la estructura semántica subyacente. El objetivo de estos sistemas es principalmente ayudar a la comprensión, gestión y recuperación de los conceptos contenidos en cada dominio. Dado que cualquier disciplina tiene necesidad de mostrarse de forma comprensible y de facilitar mecanismos de gestión y de recuperación conceptual, los Sistemas de Organización del Conocimiento se encuentran en diversos estudios, desde la Pedagogía a la Inteligencia Artificial, pasando por la Documentación y la Ingeniería del Software entre otras. En cada una de estas disciplinas, los Sistemas de Organización del Conocimiento han desarrollado

esquemas organizativos que han recibido distintos nombres, desde *tesauros* en Documentación, *mapas conceptuales* en Pedagogía u *ontologías* en Inteligencia Artificial o Filosofía. En cada disciplina los esquemas tienen diferente capacidad semántica dependiendo de los objetivos, pero básicamente prevalece, como elemento común, un vocabulario para designar los conceptos establecidos en el área, y las relaciones semánticas entre estos conceptos.

La creación de un Sistema de Conocimiento requiere de un elevado esfuerzo intelectual y de un alto grado de consenso, lo cual se traduce en un proceso lento y laborioso. En los últimos años disciplinas como la Documentación, Ingeniería Ontológica y la Ingeniería del Software han reseñado las etapas comunes que existen en la creación de estos modelos. Por lo que se requiere un gran esfuerzo para automatizar en el mayor grado posible estas fases.

Una de las principales fases es la adquisición del conocimiento. Esta etapa propone identificar, mediante diferentes fuentes documentales y expertos en el dominio, los conceptos que lo conforman y las denominaciones alternativas en lenguaje natural, es decir las equivalencias. Posteriormente, se estructuran los conceptos jerárquicamente, esto es, se clasifican según estructuras de generalización-especificación. Otra etapa que frecuentemente aparece es determinar que otros tipos de relaciones existen entre los conceptos, así como las restricciones que estos deben tener.

### **2.1.16. Denominación**

El conocimiento es más que un conjunto de datos, visto solo como datos es un conjunto sobre hechos, verdades o de información almacenada a través de la experiencia o del aprendizaje (a posteriori), o a través de introspección (a priori).

Según Platón, el conocimiento se caracteriza por ser necesariamente verdadero (episteme). De otro modo, la mera creencia y opinión (ignorante de la realidad de las cosas) quedan relegadas al ámbito de lo probable y lo aparente.

El conocimiento comienza por los sentidos, pasa de estos al entendimiento y termina en la razón. Igual que en el caso del entendimiento, hay un uso meramente formal de la misma, es decir un uso lógico ya que la razón hace abstracción de todo un contenido, pero también hay un uso real.

### **2.1.17. Restricciones**

Es importante reconocer que los SE típicos tienen las siguientes restricciones:

- Los conocimientos se obtienen de un número pequeño de especialistas.
- La aplicación está orientada a un área específica limitada o a una pequeña colección de áreas.
- El área de aplicación debe tener poca necesidad de razonamientos temporales o espaciales.

- La tarea no depende del empleo de un gran volumen de conocimientos generales o de sentido común.
- El conocimiento que se requiere para ejecutar la tarea, es razonablemente completo, correcto y estable.

### 2.1.18. Otras relaciones

Se ha escrito sobre datos, información, conocimiento, aprendizaje, inteligencia pero que son estos términos que frecuentemente causan ambigüedad y como se relacionan.

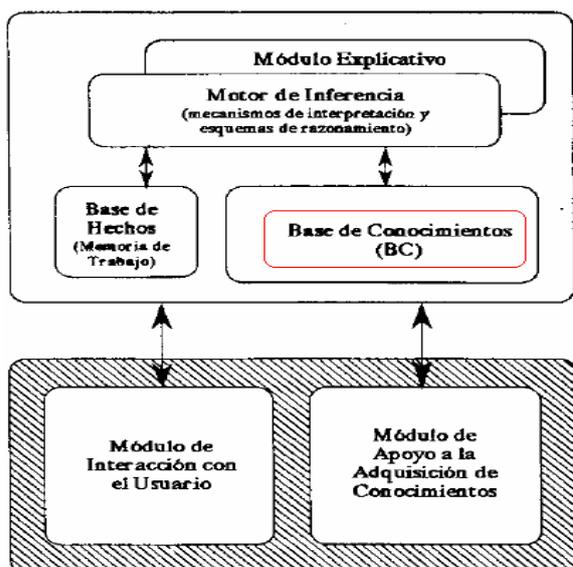
- **Datos:** Valores recolectados del mundo real, estos no tienen contexto ni relación entre si Ej: 90, Matemática, Audie.
- **Información:** Cuando los datos son sometidos a un proceso el cual los relaciona entre si Ej. Audie gano el curso de matemática con 90 puntos
- **Conocimiento:** Se genera cuando la información es sometida al uso y hay experiencia gracias a la vivencia. Ej. Audie es buen estudiante porque saco 90 puntos en matemática, debería estudiar un poco más para llegar a los 100 puntos.
- **Inteligencia:** Nace gracias a que tenemos conocimiento sobre el tema y podemos predecir o imaginarnos con un grado aceptable de certeza acerca de acontecimientos futuros, está relacionada directamente con la capacidad de aprender. Ej. Como Audie estudia matemática los sábados y saca 90, yo voy a estudiar matemática todos los días así saco 100 en el curso.

- **Aprendizaje:** Es cuando tomamos riesgos y les agregamos nuestros errores, es decir, aprender es el resultado entre perder el miedo y arriesgarnos a salir del estado estable en el que nos encontramos y pasar a un estado de incertidumbre de cambios y no tememos a equivocarnos, después de todo el problema no es equivocarse sino el problema es no aprender de los errores.

### 2.1.19. Clasificación del conocimiento

La base de conocimiento es un depósito de las primitivas del conocimiento tales como reglas de procedimiento y reglas heurísticas, y que define la capacidad del SE para actuar como un experto.

**Figura 10.** Clasificación del conocimiento



Fuente: [www.google.com.ec/imgres?=&Modo+explicativo+de+un+sistema+experto](http://www.google.com.ec/imgres?=&Modo+explicativo+de+un+sistema+experto)

### 2.1.20. Técnicas de representación El Conocimiento

La representación del conocimiento y el razonamiento es un área de la inteligencia artificial cuyo objetivo fundamental es representar el conocimiento de una manera que facilite la inferencia (sacar conclusiones) a partir de dicho conocimiento. Analiza cómo pensar formalmente - cómo usar un sistema de símbolos para representar un dominio del discurso (aquello de lo que se puede hablar), junto con funciones que permitan inferir (realizar un razonamiento formal) sobre los objetos. Generalmente, se usa algún tipo de lógica para proveer una semántica formal de como las funciones de razonamiento se aplican a los símbolos del dominio del discurso, además de proveer operadores como cuantificadores, operadores modales, etc. Esto, junto a una teoría de interpretación, da significado a las frases en la lógica.

Cuando diseñamos una representación del conocimiento (y un sistema de representación del conocimiento para interpretar frases en la lógica para poder derivar inferencias de ellas) tenemos que hacer elecciones a lo largo de un número de ámbitos de diseño. La decisión más importante que hay que tomar es la *expresividad* de la representación del conocimiento. Cuanto más expresiva es, decir algo es más fácil y más compacto. Sin embargo, cuanto más expresivo es un lenguaje, más difícil es derivar inferencias automáticamente de él. Un ejemplo de una representación del conocimiento poco expresiva es la lógica proposicional.

Un ejemplo de una representación del conocimiento muy expresiva es la lógica

autoepistémica. Las representaciones del conocimiento poco expresivas pueden ser tanto completas como consistentes (formalmente menos expresivas que la teoría de conjuntos). Las representaciones del conocimiento más expresivas pueden ser ni completas ni consistentes.

El principal problema es encontrar una representación del conocimiento y un sistema de razonamiento que el soporte que pueda hacer las inferencias que necesite tu aplicación dentro de los límites de recursos del problema a tratar. Los desarrollos recientes en la representación del conocimiento han sido liderados por la web semántica, y han incorporado el desarrollo de lenguajes y estándares de representación del conocimiento basados en XML, que incluyen Resource Description Framework (RDF), RDF Schema, DARPA Agent Markup Language (DAML), y Web Ontology Language (OWL).

#### **2.1.21. Weka y Redes Neuronales**

Weka es un conjunto de librerías JAVA para la extracción de conocimientos desde bases de datos. Es un software ha sido desarrollado en la universidad de Waikato (Nueva Zelanda) bajo licencia GPL lo cual ha impulsado que sea una de las suites más utilizadas en el área en los últimos años.

Este documento introduce con sencillos ejemplos algunos de las técnicas más importantes de este software. Para el desarrollo de los ejemplos es necesario utilizar los siguientes conjuntos de datos.

WEKA se distribuye como un fichero ejecutable comprimido de java (fichero "jar"), que se invoca directamente sobre la máquina virtual JVM. En las primeras versiones de WEKA se requería la máquina virtual Java 1.2 para invocar a la interfaz gráfica, desarrollada con el paquete gráfico de Java *Swing*.

En el caso de la última versión, WEKA 3-4, que es la que se ha utilizado para confeccionar estas notas, se requiere Java 1.3 o superior. La herramienta se invoca desde el intérprete de Java, en el caso de utilizar un entorno Windows, bastaría una ventana de comandos para invocar al intérprete Java:

#### **2.1.21.1. Ficheros .arff**

Nativamente Weka trabaja con un formato denominado arff, acrónimo de *Attribute-Relation*

*File Format*. Este formato está compuesto por una estructura claramente diferenciada en tres partes:

1. **Cabecera.** Se define el nombre de la relación. Su formato es el siguiente:

```
@relation <nombre-de-la-relación>
```

Donde <nombre-de-la-relación> es de tipo String\*. Si dicho nombre contiene algún espacio será necesario expresarlo entrecomillado.

2. **Declaraciones de atributos.** En esta sección se declaran los atributos que compondrán nuestro archivo junto a su tipo. La sintaxis es la siguiente:

```
@attribute <nombre-del-atributo> <tipo>
```

\*Entendiendo como tipo String el ofrecido por Java.

3. Donde <nombre-del-atributo> es de tipo String teniendo las mismas restricciones que el caso anterior. Weka acepta diversos tipos, estos son:

- a) **NUMERIC** Expresa números reales\*.
- b) **INTEGER** Expresa números enteros.
- c) **DATE** Expresa fechas, para ello este tipo debe ir precedido de una etiqueta de formato entrecomillada.

La etiqueta de formato está compuesta por caracteres separadores (guiones y/o espacios) y unidades de tiempo:

dd Día.

MM Mes.

yyyy Año.

HH Horas.

mm Minutos.

ss Segundos.

d) **STRING** Expresa cadenas de texto, con las restricciones del tipo String comentadas anteriormente.

e) **ENUMERADO** El identificador de este tipo consiste en expresar entre llaves y separados por comas los posibles valores (caracteres o cadenas de caracteres) que puede tomar el atributo. Por ejemplo, si tenemos un atributo que indica el tiempo podría definirse:

@attribute tiempo {soleado,lluvioso,nublado}

4. **Sección de datos.** Declaramos los datos que componen la relación separando entre comas los atributos y con saltos de línea las relaciones.

@data

Aunque éste es el modo "completo" es posible definir los datos de una forma abreviada (*sparse data*). Si tenemos una muestra en la que hay muchos datos que sean 0 podemos expresar los datos prescindiendo de los elementos que son nulos, rodeando cada una de las filas entre llaves y situando delante de cada uno de los datos el número de atributo.

### **2.1.21.2. Introducción A Weka**

En este capítulo se presenta de forma concisa y práctica la herramienta de minería de datos WEKA. WEKA, acrónimo de *Waikato Environment for Knowledge Analysis*, es un entorno para experimentación de análisis de datos que permite aplicar, analizar y evaluar las técnicas más relevantes de análisis de datos, principalmente las provenientes del aprendizaje automático, sobre cualquier conjunto de datos del usuario. Para ello únicamente se requiere que los datos a analizar se almacenen con un cierto formato, conocido como *ARFF* (*Attribute-Relation File Format*).

WEKA se distribuye como software de libre distribución desarrollado en Java y está constituido por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación, agrupamiento, asociación, y visualización, así como facilidades para su aplicación y análisis de prestaciones cuando son aplicadas a los datos de entrada seleccionados. Estos paquetes pueden ser integrados en cualquier proyecto de análisis de datos, e incluso pueden extenderse con contribuciones de los usuarios que desarrollen nuevos algoritmos. Con objeto de facilitar su uso por un mayor número de usuarios,

WEKA además incluye una interfaz gráfica de usuario para acceder y configurar las diferentes herramientas integradas.

Una de las características más interesantes de Weka es la posibilidad de modificar su código y obtener versiones adaptadas con funcionalidades que no contengan las versiones oficiales. El primer paso para poder modificar Weka es descargarse la última *snapshot* (versión en desarrollo) que se encuentra en el servidor de versiones (CVS).

### 2.1.21.3. Datos de entrenamiento

Los datos de entrada a la herramienta, sobre los que operarán las técnicas implementadas, deben estar codificados en un formato específico, denominado *Attribute-Relation File Format* (extensión "arff"). La herramienta permite cargar los datos en tres soportes: fichero de texto, acceso a una base de datos y acceso a través de internet sobre una dirección URL de un servidor web. En nuestro caso trabajaremos con ficheros de texto. Los datos deben estar dispuestos en el fichero de la forma siguiente: cada instancia en una fila, y con los atributos separados por comas. El formato de un fichero arff sigue la estructura siguiente:

% comentarios

@relation NOMBRE\_RELACION

@attribute r1 real

@attribute r2 real ...

...

@attribute i1 integer

@attribute i2 integer

...

@attribute s1 {v1\_s1, v2\_s1,...vn\_s1}

@attribute s2 {v1\_s1, v2\_s1,...vn\_s1}

...

@data

DATOS

por tanto, los atributos pueden ser principalmente de dos tipos: numéricos de tipo real o entero (indicado con las palabra *real* o *integer* tras el nombre del atributo), y simbólicos, en cuyo caso se especifican los valores posibles que puede tomar entre llaves.

#### 2.1.21.4. Clasificadores

Si queremos realizar una clasificación lo primero será elegir un clasificador y configurarlo a nuestro gusto, para ello pulsaremos sobre el botón **Choose** dentro del área *Classifier*. Una vez pulsado se desplegará un árbol que nos permitirá seleccionar el clasificador deseado. Una vez seleccionado aparecerá, en la etiqueta contigua al botón **Choose**, el filtro seleccionado y los argumentos con los que se ejecutará. Esta información es muy útil si queremos utilizar el interfaz de consola ya que podremos configurar nuestro filtro con la interfaz y luego obtener el resultado apto para línea de mandato.

Para poder acceder a las propiedades de cada clasificador deberemos hacer *double-click* sobre la etiqueta antes mencionada. Al darle aparecerá una nueva ventana con las propiedades junto a una breve explicación del mismo. Una vez

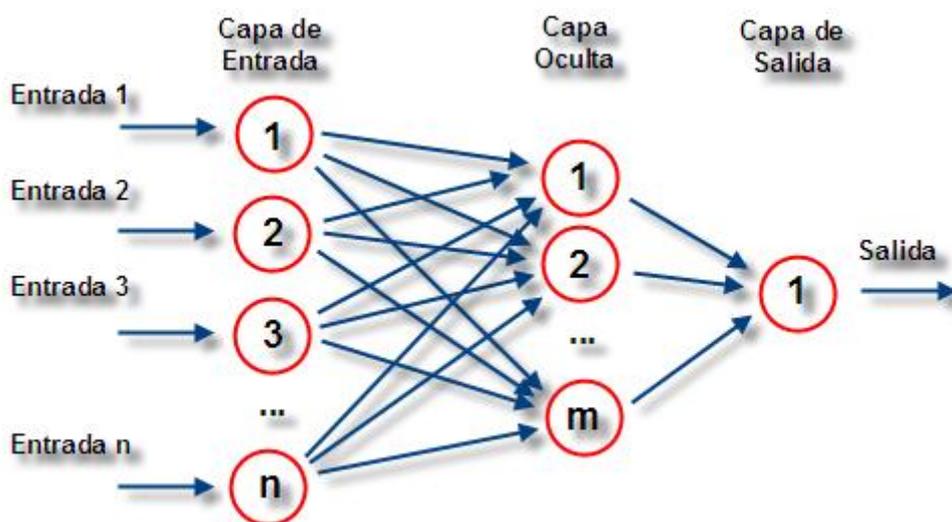
elegido el clasificador y sus características el próximo paso es la configuración del modo de entrenamiento (*Test Options*). Weka proporciona 4 modos de prueba:

#### 2.1.21.5. Use training set

Con esta opción Weka entrenará el método con todos los datos disponibles y luego lo aplicará otra vez sobre los mismos.

#### 2.1.22. Multilayer Perceptron

Figura 11. Multilayer Perceptron



Fuente: [www.google.com.ec/imgres?=&Multilayer+perceptron+de+un+sistema+experto](http://www.google.com.ec/imgres?=&Multilayer+perceptron+de+un+sistema+experto)

### 2.1.22.1. Tipos

Las capas pueden clasificarse en tres tipos:

- Capa de entrada: Constituida por aquellas neuronas que introducen los patrones de entrada en la red. En estas neuronas no se produce procesamiento.
- Capas ocultas: Formada por aquellas neuronas cuyas entradas provienen de capas anteriores y cuyas salidas pasan a neuronas de capas posteriores.
- Capa de salida: Neuronas cuyos valores de salida se corresponden con las salidas de toda la red.

La propagación hacia atrás (también conocido como retropropagación del error o regla delta generalizada), es un algoritmo utilizado en el entrenamiento de estas redes, por ello, el perceptrón multicapa también es conocido como red de retropropagación (no confundir con la red de contrapropagación).

### 2.1.22.2. Historia

En 1969, Minsky y Papert, demuestran que el perceptrón simple y ADALINE no puede resolver problemas no lineales (por ejemplo, XOR). La combinación de varios perceptrones simples podría resolver ciertos problemas no lineales pero no existía un mecanismo automático para adaptar los pesos de la capa oculta. Rumelhart y otros autores, en 1986, presentan la "Regla Delta Generalizada" para adaptar los pesos propagando los errores hacia atrás, es decir, propagar los

errores hacia las capas ocultas inferiores. De esta forma se consigue trabajar con múltiples capas y con funciones de activación no lineales. Se demuestra que el perceptrón multicapa es un aproximador universal. Un perceptrón multicapa puede aproximar relaciones no lineales entre los datos de entrada y salida. Esta red se ha convertido en una de las arquitecturas más utilizadas en el momento.

### **2.1.22.3. Características**

- Las funciones de transferencia de los elementos de procesado (neuronas) han de ser derivables.

### **2.1.22.4. Limitaciones**

- El Perceptrón Multicapa no extrapola bien, es decir, si la red se entrena mal o de manera insuficiente, las salidas pueden ser imprecisas.
- La existencia de mínimos locales en la función de error dificulta considerablemente el entrenamiento, pues una vez alcanzado un mínimo el entrenamiento se detiene aunque no se haya alcanzado la tasa de convergencia fijada.

Cuando caemos en un mínimo local sin satisfacer el porcentaje de error permitido se puede considerar: cambiar la topología de la red (número de capas y número de neuronas), comenzar el entrenamiento con unos pesos iniciales diferentes, modificar los parámetros de aprendizaje, modificar el conjunto de entrenamiento o presentar los patrones en otro orden.

### 2.1.22.5. Aplicaciones

El perceptrón multicapa (de aquí en adelante MLP, MultiLayer Perceptron) se utiliza para resolver problemas de asociación de patrones, segmentación de imágenes, compresión de datos, etc.

### 2.1.23. Redes Neuronales

Las *redes neuronales* son un elemento importante de las denominadas tecnologías de *Inteligencia Artificial* (IA).

La IA es "*la disciplina científica y técnica que se ocupa del estudio de las ideas que permiten ser inteligentes a los ordenadores*" (definición de H. Winston).

Otra posible definición de la IA sería: rama de la computación que se encarga, entre otras cosas, de los problemas de percepción, razonamiento y aprendizaje en relación con sistemas artificiales, y que tiene como áreas de investigación a los sistemas expertos y de conocimiento, la robótica, los lenguajes naturales y las *redes neuronales*.

Y a pesar de que el objetivo final de la IA, dotar de auténtica inteligencia a las máquinas, queda todavía muy lejos (e incluso hay autores que defienden que esto nunca será posible), la ciencia de la *Inteligencia Artificial* ha generado numerosas herramientas prácticas, entre las que se encuentran las redes neuronales.

Las *redes neuronales*, también llamadas "*redes de neuronas artificiales*", son *modelos* bastante simplificados de las redes de neuronas que forman el cerebro. Y, al igual que este, intentan "**aprender**" a partir de los *datos* que se le suministran.

Así, las principales *características* que diferencian a las *redes neuronales* de otras tecnologías de IA son:

- Su capacidad de **aprendizaje** a partir de la experiencia (*entrenamiento*). Normalmente, para la elaboración de un *programa informático* es necesario un *estudio* detallado de la tarea a realizar para después *codificarla* en un lenguaje de programación. Pero, las *redes neuronales* pueden ser *entrenadas* para realizar una determinada tarea sin necesidad de un estudio a fondo ni programarla usando un lenguaje de programación. Además; las *redes neuronales* pueden volver a entrenarse para ajustarse a nuevas necesidades de la tarea que realizan, sin tenerse que reescribir o revisar el código (cosa frecuente en programas tradicionales).
- Su **velocidad** de respuesta una vez concluido el entrenamiento. Se comportan también en este caso de manera similar a como lo hace el cerebro: los seres humanos no necesitamos pensar mucho para identificar un objeto, una palabra,... una vez hemos aprendido a hacerlo.
- Su **robustez**, en el sentido de que el conocimiento adquirido se encuentra repartido por toda la red, de forma que si se lesiona una parte se continúan generando cierto número de respuestas correctas (en este caso también hay cierta analogía con los cerebros parcialmente dañados).

### 2.1.23.1. Componentes de una Red Neuronal

La misma está constituida por neuronas interconectadas y arregladas en tres capas (esto último puede variar). Los datos ingresan por medio de la “*capa de entrada*”, pasan a través de la “*capa oculta*” y salen por la “*capa de salida*”. Cabe mencionar que la capa oculta puede estar constituida por varias capas.

Antes de comenzar el estudio sobre las redes neuronales, se debe aprender algo sobre las neuronas y de cómo ellas son utilizadas por una red neuronal.

Se compara una neurona biológica con una neurona artificial. En la misma se pueden observar las similitudes entre ambas (tienen entradas, utilizan pesos y generan salidas). *comparación entre una neurona biológica (izquierda) y una artificial (derecha).*

Mientras una neurona es muy pequeña en sí misma, cuando se combinan cientos, miles o millones de ellas pueden resolver problemas muy complejos. Por ejemplo el cerebro humano se compone de billones de tales neuronas.

### 2.1.23.2. La Neurona

La neurona artificial fue diseñada para "emular" las características del funcionamiento básico de la neurona biológica. En esencia, se aplica un conjunto de entradas a la neurona, cada una de las cuales representa una salida de otra neurona. Cada entrada se multiplica por su "peso" o ponderación correspondiente análoga al grado de conexión de la sinapsis. Todas las entradas ponderadas se

suman y se determina el nivel de excitación o activación de la neurona. Una representación vectorial del funcionamiento básico de una neurona artificial se indica según la siguiente expresión de la ecuación.

## 2.1.24. Función de Integración

### 2.1.24.1. Función de Transferencia o Activación

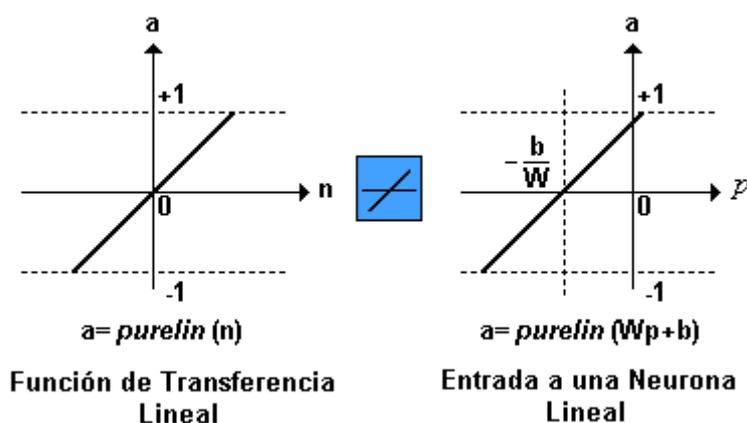
#### 2.1.24.1.1. Función de transferencia lineal (purelin)

La salida de una función de transferencia lineal es igual a su entrada, la cual se representa en la figura 12.

**Figura 12.** Función de Transferencia Lineal (purelin)

$$a = n$$

(1.5)



Fuente: [www.hugo-inc.net16.net](http://www.hugo-inc.net16.net)

En la gráfica del lado derecho de la figura 12, puede verse la característica de la salida  $a$  de la red, comparada con la entrada  $p$ , más un valor de ganancia  $b$ ,

neuronas que emplean esta función de transferencia son utilizadas en la red tipo Adaline.

#### **2.1.24.1.2. Función de transferencia sigmoideal (logsig)**

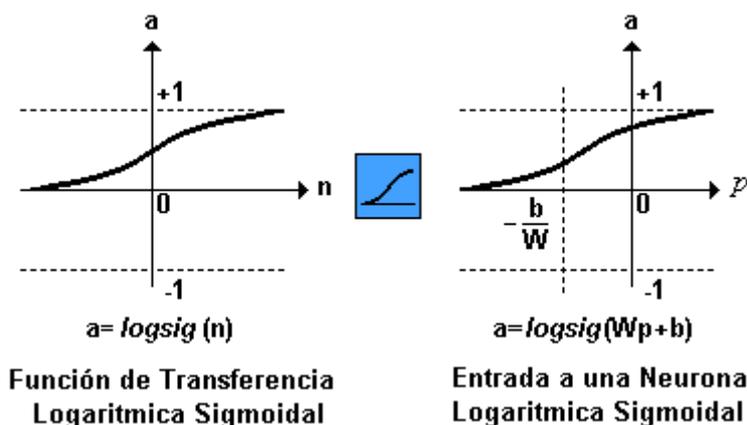
Esta función toma los valores de entrada, los cuales pueden oscilar entre más y menos infinito, y restringe la salida a valores entre cero y uno, de acuerdo a la expresión:

$$a = \frac{1}{1 + e^{-n}}$$

(1.6)

Esta función es comúnmente usada en redes multicapa, como la Backpropagation, en parte porque la función logsig es diferenciable, como se muestra en la Fig. 13.

**Figura 13.** Función de Transferencia Sigmoidal.



**Fuente:** [www.hugo-inc.net16.net](http://www.hugo-inc.net16.net)

La Fig. 13, hace una relación de las principales funciones de transferencia empleadas en el entrenamiento de redes neuronales.

### 2.1.25. Red de Retro-Propagación

Al hablar de redes de retropropagación o redes de propagación hacia atrás hacemos referencia a un algoritmo de aprendizaje más que a una arquitectura determinada. La retropropagación consiste en propagar el error hacia atrás, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas ocultas intermedias y ajustando los pesos de las conexiones con el fin de reducir dicho error. Hay distintas versiones o reglas del algoritmo de retropropagación y distintos arquitecturas conexionistas a las que pueden ser aplicados.

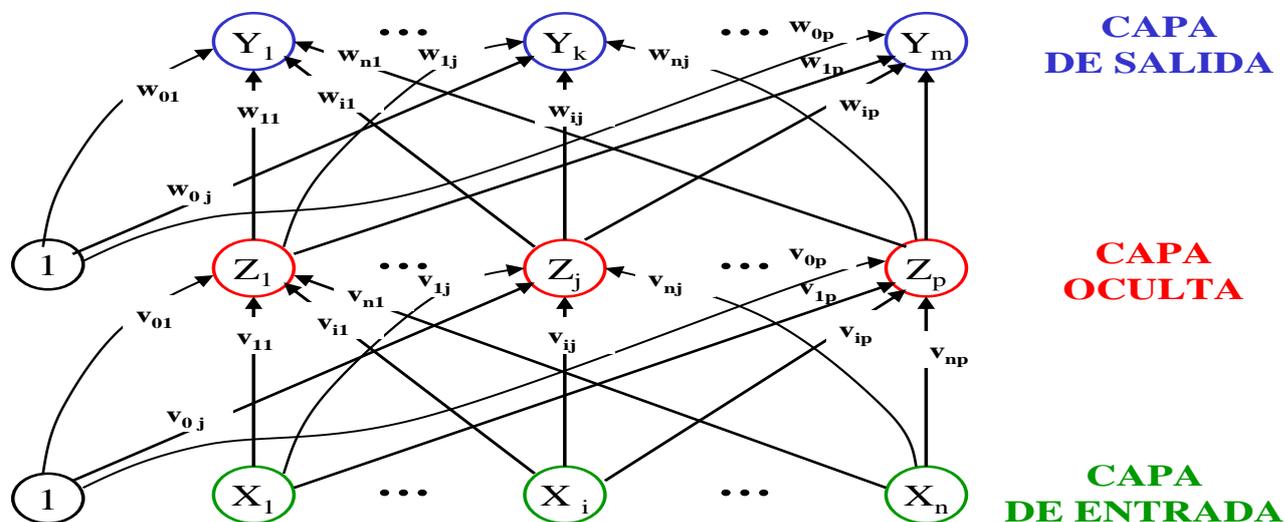
Durante mucho tiempo no se dispuso de algoritmos para entrenar redes multicapa, y como las redes de una capa estaban muy limitadas en cuanto a lo

que eran capaces de representar, el campo de las redes neuronales artificiales estaba estancado. La invención y perfeccionamiento del algoritmo de retropropagación dio un gran impulso al desarrollo de este campo. Tiene un buen fundamento matemático y a pesar de sus limitaciones ha expandido enormemente el rango de problemas donde se aplican las redes neuronales artificiales.

### 2.1.25.1. Arquitectura de una Red de Retro-propagación

A continuación se representa la arquitectura de una red neuronal multicapa con una capa oculta:

**Figura 14.** Arquitectura de una Red de Retro-propagación



Fuente: [www.arantxa.ii.uam.es](http://www.arantxa.ii.uam.es)

Las neuronas de **entrada** son las unidades  $X$ . Su salida es simplemente el patrón con el que se quiere estimular la red. El número de estas unidades es igual al número  $n$  de atributos que tiene cada patrón.

Las neuronas de la **capa oculta**, y de la **capa de salida**, son las unidades  $Z$  e  $Y$  respectivamente. Cada una de estas neuronas tiene un *bias*, cada uno de distinto valor. El bias que recibe la neurona  $Z_j$  se denomina  $v_{0j}$ , y el que recibe la neurona  $Y_k$  se denota  $w_{0k}$ . Estos bias actúan como pesos procedentes de neuronas cuya salida es siempre 1.

El número de neuronas en la capa oculta es arbitrario ( $p$ ), y el de neuronas en la capa de salida es igual a la dimensión del vector de datos que queremos predecir,  $m$ .

La dirección de las flechas en el diagrama indica la dirección del flujo de información para la fase de propagación hacia delante. Durante la fase de retropropagación, la información fluye justo en sentido contrario.

#### **2.1.26. Marco conceptual**

Los sistemas **expertos** incorporan en la base de conocimiento del sistema el conocimiento de un experto e intentan simular el razonamiento humano por medio de un conjunto de programas de computación.

### **2.1.26.1. Base de conocimiento**

Es la parte del sistema experto que contiene el conocimiento sobre el dominio. Hay que obtener el conocimiento del experto y codificarlo en la base de conocimientos. Una forma clásica de representar el conocimiento en un sistema experto son las reglas. Una regla es una estructura condicional que relaciona lógicamente la información contenida en la parte del antecedente con otra información contenida en la parte del consecuente.

### **2.1.26.2. Base de hechos (Memoria de trabajo)**

Contiene los hechos sobre un problema que se han descubierto durante una consulta. Durante una consulta con el sistema experto, el usuario introduce la información del problema actual en la base de hechos. El sistema empareja esta información con el conocimiento disponible en la base de conocimientos para deducir nuevos hechos.

### **2.1.26.3. Motor de inferencia**

El sistema experto modela el proceso de razonamiento humano con un módulo conocido como el motor de inferencia. Dicho motor de inferencia trabaja con la información contenida en la base de conocimientos y la base de hechos para deducir nuevos hechos. Contrasta los hechos particulares de la base de hechos con el conocimiento contenido en la base de conocimientos para obtener conclusiones acerca del problema.

#### **2.1.26.4. Subsistema de Explicación**

Una característica de los sistemas expertos es su habilidad para explicar su razonamiento. Usando el módulo del subsistema de explicación, un sistema experto puede proporcionar una explicación al usuario de por qué está haciendo una pregunta y cómo ha llegado a una conclusión. Este módulo proporciona beneficios tanto al diseñador del sistema como al usuario. El diseñador puede usarlo para detectar errores y el usuario se beneficia de la transparencia del sistema.

#### **2.1.26.5. Interfaz de Usuario**

La interacción entre un sistema experto y un usuario se realiza en lenguaje natural. También es altamente interactiva y sigue el patrón de la conversación entre seres humanos. Para conducir este proceso de manera aceptable para el usuario es especialmente importante el diseño del interfaz de usuario. Un requerimiento básico del interfaz es la habilidad de hacer preguntas. Para obtener información fiable del usuario hay que poner especial cuidado en el diseño de las cuestiones. Esto puede requerir diseñar el interfaz usando menús o gráficos.

#### **2.1.26.6. Componentes de un Sistema Experto**

La Base de conocimiento nos halla la base datos y está compuesta por lenguajes de predicado, esta es uno de los componentes que contiene el conocimiento del

experto o también llamado base de datos, su función es almacenar experiencias, conocimientos, etc. de una determinada área.

Existen dos tipos de base de conocimiento:

**El procedural:**

- Se usa en los lenguajes. estructurados como son Pascal, C, Visual Basic etc.

**El declarativo:**

- Está basado en hechos que vienen a ser acciones que se dan dentro del problema que utilizan los lenguajes Prolog y Lisp.

**2.1.26.7. El motor de inferencia**

- Su función es administrar , como, cuando, y las reglas de producción que se aplicaran para la solución de un determinado problema
- Dirige y controla la implementación del conocimiento, además permite decidir que tipo de técnicas se usaran durante el diseño del sistema experto.

**2.1.26.8. La interface**

- Parte que permite la comunicación con el usuario, en forma bidireccional (ambos lados). Mediante al Interface el Motor de Inferencia reconoce la pregunta y saca datos de la Base de Conocimiento y mediante la Interface responde la pregunta

### 2.1.26.9. Módulo de Comunicaciones

Puesto que los sistemas basados en el conocimiento no viven aislados sino que interactúan con otros sistemas es decir no solo son capaces de interactuar con el experto sino con estos sistemas, para poder recoger información o consultar bases de datos.

### 2.1.27. El Modelo Entidad-Relación

- Se usa para diseñar la BD
- La mayoría de BD actuales son de tecnología **relacional** Conceptos base:
  - entidades (y atributos y elementos) relación de Entidades
- Los objetos que aparece en la vida real, es lo que llamamos **entidad**. Por ejemplo, alumnos, empleados, aviones, coches, alojamientos,

### 2.1.28. BD. Atributos

- Estas entidades están compuestas por varios **atributos**, que vienen a ser sus propiedades. Por ejemplo, la entidad alumnos, tendrá los atributos nombre, DNI, nacionalidad, fecha de nacimiento.
- Los atributos también reciben el nombre de **columnas** en la de Elementos
- Cada entidad tendrá un número ilimitado de elementos. Por ejemplo, un elemento de la entidad alumnos será un alumno en sí; así el alumno Pepe será un elemento, José será otro,
- Cada uno de esos elementos también recibe el nombre de **fila** en la terminología

de BDRminología de BDRelaciones

- Las entidades no están aisladas sino que están relacionadas entre sí.
- Estas **relaciones** pueden ser de tres tipos diferentes:
  - 1 a 1
  - 1 a muchos (1 a N)
  - Muchos a muchos (M a N)

### 2.1.29. MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

Los desarrolladores comentan en la documentación que todas estas carencias no les resultaba un problema, ya que era lo que ellos necesitaban. De hecho, MySQL fue diseñada con estas características, debido a que lo que buscaban era un gestor de bases de datos con una gran rapidez de respuesta. Pero ha sido con la distribución de MySQL por Internet, cuando más y más gente les está pidiendo estas funcionalidades, por lo que serán incluidas en futuras versiones del gestor.

### **2.1.30. Java**

Java es **un lenguaje de programación orientado a objetos** que se popularizó a partir del lanzamiento de su primera versión comercial de amplia difusión, la JDK 1.0 en 1996. Actualmente es uno de los lenguajes más usados para la programación en todo el mundo.

Los antecedentes de Java habría que buscarlos en los lenguajes de programación C y C++. El lenguaje C fue desarrollado en la década de los 70 y constituye un lenguaje muy robusto usado como núcleo del sistema operativo Unix. C no admite la orientación a objetos y está considerado un lenguaje “poco amigable” desde el punto de vista de que su sintaxis, elementos de programación que emplea (manejo directo de memoria) y otras cuestiones hacen que sea un lenguaje difícil de aprender. C++ fue una evolución de C desarrollada en los años 80. Introdujo el diseño orientado a objetos, pero manteniendo la compatibilidad con C. Esto suponía que C++ permitiera tanto el uso de la programación estructurada “tradicional” como la programación orientada a objetos. Además C++ mantuvo ciertas características de C como el manejo directo de la memoria, el uso de variables globales, sentencia goto, etc. que hicieron que la evolución fuera “parcial”.

Como paso final en esta evolución tenemos Java, un lenguaje que evoluciona a partir de C y C++, pero que elimina diversos aspectos de estos lenguajes y se constituye en un lenguaje definitivamente orientado a objetos. El romper con distintos aspectos de C++ cuyo manejo inadecuado por parte de muchos programadores daba lugar a problemas en las aplicaciones ha sido un factor decisivo para convertir a Java en un lenguaje popular y de amplio uso.

Nosotros vamos a quedarnos con el lado práctico de lo que hemos comentado respondiendo esta pregunta: **¿Es necesario saber C ó C++ para programar en Java?** No, no es necesario. Aunque puede suponer una ventaja para aquellas personas que tengan conocimientos previos en estos lenguajes, no

recomendamos de forma explícita su estudio en profundidad como paso previo al aprendizaje de Java.

De modo orientativo, veamos un esquema sobre la evolución de los lenguajes.

Este esquema es meramente orientativo: es imposible reflejar la diversidad de lenguajes y su evolución en un gráfico tan simplificado. También los datos relativos a fechas son orientativos.

No hay que suponer que lo único válido sea la programación orientada a objetos por ser lo más moderno. Al contrario, muchísima programación de la que se hace hoy en día se basa en lenguajes o código no orientado a objetos. Además, la misma programación orientada a objetos se basa en conceptos muy antiguos de programación.

Tener en cuenta que algunos lenguajes que nacieron en los años 60 han perdido vigencia y ya no se usan, mientras que otros se han ido modernizando y continúan usándose más o menos ampliamente, como es el caso de Cobol. Lo que consideramos interesante con este esquema es que se vea que Java es una evolución que por un lado incorpora cosas que se venían usando desde hace mucho tiempo en programación, y por otro introduce ciertas novedades que lo convierten en un lenguaje moderno.

Java es un lenguaje útil para casi todo tipo de problemas. Podemos citar como funcionalidades de Java varias:

**Aplicaciones “cliente”:** son las que se ejecutan en un solo ordenador (por ejemplo el portátil de tu casa) sin necesidad de conectarse a otra máquina. Pueden servirte por ejemplo para realizar cálculos o gestionar datos.

**Aplicaciones “cliente/servidor”:** son programas que necesitan conectarse a otra máquina (por ejemplo un servidor de datos) para pedirle algún servicio de forma más o menos continua, como podría ser el uso de una base de datos. Pueden servir por ejemplo para el teletrabajo: trabajar desde casa pero conectados a un ordenador de una empresa.

**Podemos hablar también de “aplicaciones web”**, que son programas Java que se ejecutan en un servidor de páginas web. Estas aplicaciones reciben “solicitudes” desde un ordenador y envían al navegador (Internet Explorer, Firefox, Safari, etc.) que actúa como su cliente páginas de respuesta en HTML.

Éstos son sólo algunos ejemplos de todo el potencial que hay detrás de Java como lenguaje para aprender y obtener muchos beneficios con su uso. Obviamente por determinados términos empleados (cliente, cliente/servidor, base de datos, HTML...), te darás cuenta de que el lenguaje Java tiene mucha potencialidad, pero también de que su conocimiento a fondo requeriría mucho tiempo. Nosotros en este curso vamos a estudiar únicamente los aspectos más básicos de Java.

### 2.1.31. Ficheros .arff

Nativamente Weka trabaja con un formato denominado arff, acrónimo de *Attribute-Relation*

*File Format*. Este formato está compuesto por una estructura claramente diferenciada en tres partes:

1. **Cabecera.** Se define el nombre de la relación. Su formato es el siguiente:

@relation <nombre-de-la-relación>

Donde <nombre-de-la-relación> es de tipo String\*. Si dicho nombre contiene algún espacio será necesario expresarlo entrecorillado.

#### 2.1.31.1. Preparación de los Datos

Nativamente Weka trabaja con un formato denominado *arff*, acrónimo de *Attribute-Relation File Format*.

Este formato está compuesto por una estructura claramente diferenciada en tres partes:

**Cabecera:** Se define el nombre de la relación. Su formato es el siguiente:

@relation <nombre-de-la-relación>

- **Declaraciones de atributos:** En esta sección se declaran los atributos que compondrán el archivo junto a su tipo. La sintaxis es la siguiente:

@attribute <nombre-del-atributo> <tipo>

Donde:

*<nombre-del-atributo> es de tipo string.*

*<tipo> acepta diversos tipos, estos son:*

— *NUMERIC Expresa números reales.*

— *INTEGER Expresa números enteros.*

— *DATE Expresa fechas.*

— *STRING Expresa cadenas de texto.*

- **Sección de datos:** Se declaran los datos que componen la relación separando con comas los atributos y con saltos de línea las relaciones.

La sintaxis es la siguiente:

@data

Una vez conocido el formato de los datos soportado por el Weka, se pasará al confeccionado del archivo con extensión arff.

Los mismos se pueden convertir ficheros de texto conteniendo un registro por línea con los atributos separados por comas (formato csv) a ficheros *arff* mediante el uso de un filtro convertidor.

Con la información recolectada a través de la *EPH (Encuesta Permanente de Hogares)* se han generado una base de datos *Microsoft Access*.

La información será recabada en una planilla de hoja de cálculos *Microsoft Excel*, luego se la convertirá a un documento de texto plano (.txt, .doc, etc.) para su posterior transformación a un archivo de formato específico de datos legible por el *Weka*, el formato *.arff*.

### **2.1.32. Weka**

Corresponde a las siglas en inglés de *Waikato Environment for Knowledge Analysis*. Es una herramienta de software desarrollada en Java por la Universidad de Wakaito (Nueva Zelanda). Esta herramienta contiene principalmente algoritmos de *machine Learning* usados en el proceso de la minería de datos. WEKA incluye herramientas para el reprocesamiento de los datos (filtros), clasificación (árboles, tablas), *clustering*, reglas de asociación y adicionalmente, diversas formas de visualización de los datos, tanto en el inicio del proceso de carga de datos, como después de haber aplicado un algoritmo. Al estar cubierta con la licencia GNU, es posible usar directamente las implementaciones en Java de los algoritmos y unirlos a otros tipos de software, para crear procesos de minería semi-automatizados.

### **2.1.33. Multiplayer Perceptron**

El **perceptrón multicapa** es una red neuronal artificial (RNA) formada por

múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón (también llamado perceptrón simple). El perceptrón multicapa puede ser totalmente o localmente conectado. En el primer caso cada salida de una neurona de la capa "i" es entrada de todas las neuronas de la capa "i+1", mientras que en el segundo cada neurona de la capa "i" es entrada de una serie de neuronas (región) de la capa "i+1".

#### **2.1.34. Marco Espacial**

El presente proyecto se desarrollará El Consultorio del Dr. Carlos Celi durante el periodo junio 2012 – Diciembre 2012, contando con la ayuda comprometida en la parte investigativa y de validación de la base de conocimiento: del Dr. Juan Carlos Delgado y Dr. Carlos Celi.

### **3. ESTRATEGIA METODOLOGICA**

#### **3.1. METODOLOGÍA DE INVESTIGACIÓN**

La Metodología es el conjunto de métodos empleados para el desarrollo de sistemas automatizados. Una metodología completa es algo más que una notación, un proceso, y herramientas. Además de una "notación, de un proceso, y de herramientas," estas "metodologías completas" proporcionan:

- Guías para estimar costos,
- Manejo del proyecto en las tareas y entregas,
- Medidas y métricas,
- Formas definidas y dirección en las entregas de la construcción,
- Políticas y procedimientos para garantizar la calidad del software,
- Descripciones de los roles y programas de entrenamiento detallados,
- Ejemplos totalmente trabajados,
- Ejercicios de entrenamiento,
- Técnicas para adaptar el método, y
- Técnicas definidas.

Para el desarrollo del presente proyecto, se aplicó diferentes métodos investigativos, herramientas técnicas que nos facilitaron la obtención de la información con un orden secuencial y lógico de sus partes, con el fin de cumplir los objetivos planteados en la investigación.

Como técnicas de investigación científica se manejó el **Método Deductivo e Inductivo**, que permitió realizar un diagnóstico general de la problemática entorno al hospital, para llegar a los procesos particulares generados por cada una de las áreas que lo conforman.

**Método Descriptivo**, el cual permitió realizar el análisis e identificación de cómo se realizan los procesos de administración de historias clínicas, generación de turnos y médicos, diagnóstico de enfermedades hematológicas, así como envió de emails de recordatorio.

**Método Cuantitativo**, que nos facilitó el análisis de cuadros estadísticos, tabulación de datos obtenidos mediante la aplicación de las técnicas de recolección de información y así validar que el sistema cumple con los requerimientos de la institución.

**Método Bibliográfico**, este método es fundamental dentro del desarrollo de un proyecto, ya que por medio de este obtendremos los conocimientos suficientes y necesarios para poder desarrollar con éxito un proyecto. Se constituye en la base teórica sobre la cual tenemos que trabajar para obtener los resultados deseados.

### **3.2. METODOLOGÍA INFORMÁTICA**

Desde que el desarrollo de aplicaciones informáticas se empezó a considerar un proceso de ingeniería, muchas metodologías de desarrollo han ido naciendo con el fin de dar soporte al ciclo de desarrollo del proyecto.

Es así que, siendo el presente proyecto un sistema informático que maneja datos personales y clínicos de un paciente, requiere que los procesos y datos a manejarse tengan seguridad, confidencialidad, legibilidad, secuencialidad, duración, y garantía.

Por lo cual he creído conveniente utilizar como metodología para el desarrollo del presente proyecto con **XP** ya que mejorara la comunicación, busca la simplicidad, busca retroalimentación en que tan bien va nuestro trabajo.

Dentro de las ventajas que esta metodología ofrece podemos enunciar las siguientes:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.

La metodología de XP se basa en:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- Pruebas unitarias **continuas**, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Véase, por ejemplo, las herramientas de prueba JUnit orientada a Java, DUnit orientada a Delphi, NUnit para la plataforma.NET o PHPUnit para PHP. Estas tres últimas inspiradas en JUnit, la cual, a su vez, se inspiró en SUnit, el primer framework orientado a realizar tests, realizado para el lenguaje de programación Smalltalk.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. La mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- Frecuente **integración del equipo de programación con el cliente** o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.

- **Corrección de todos los errores** antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización **del código**, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartida**: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad en el código**: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos tendrá que comunicar sobre éste, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

- **Actividades de Xp**

- **Codificar**

Es necesario codificar y plasmar nuestras ideas a través del código. En programación, el código expresa la interpretación del problema, así podemos utilizar el código para comunicar, para hacer comunes las ideas, y por tanto para aprender y mejorar.

- **Hacer pruebas**

Las características del software que no pueden ser demostradas mediante pruebas simplemente no existen. Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Las pruebas nos indican que nuestro trabajo funciona, cuando no podemos pensar en ninguna prueba que pudiese originar un fallo en nuestro sistema, entonces habremos acabado por completo.

- **Escuchar**

Nos menciona en una frase, "Los programadores no lo conocemos todo, y sobre todo muchas cosas que las personas de negocios piensan que son interesantes. Si ellos pudieran programarse su propio software ¿para qué nos querrían?".

Si vamos a hacer pruebas tenemos que preguntar si lo obtenido es lo deseado, y tenemos que preguntar a quién necesita la información. Tenemos que escuchar a nuestros clientes cuáles son los problemas.

## 4. RESULTADOS

### 4.1. REQUERIMIENTOS FUNCIONALES DEL NEGOCIO

El negocio permite:

**Tabla 9.** Requerimientos funcionales del negocio

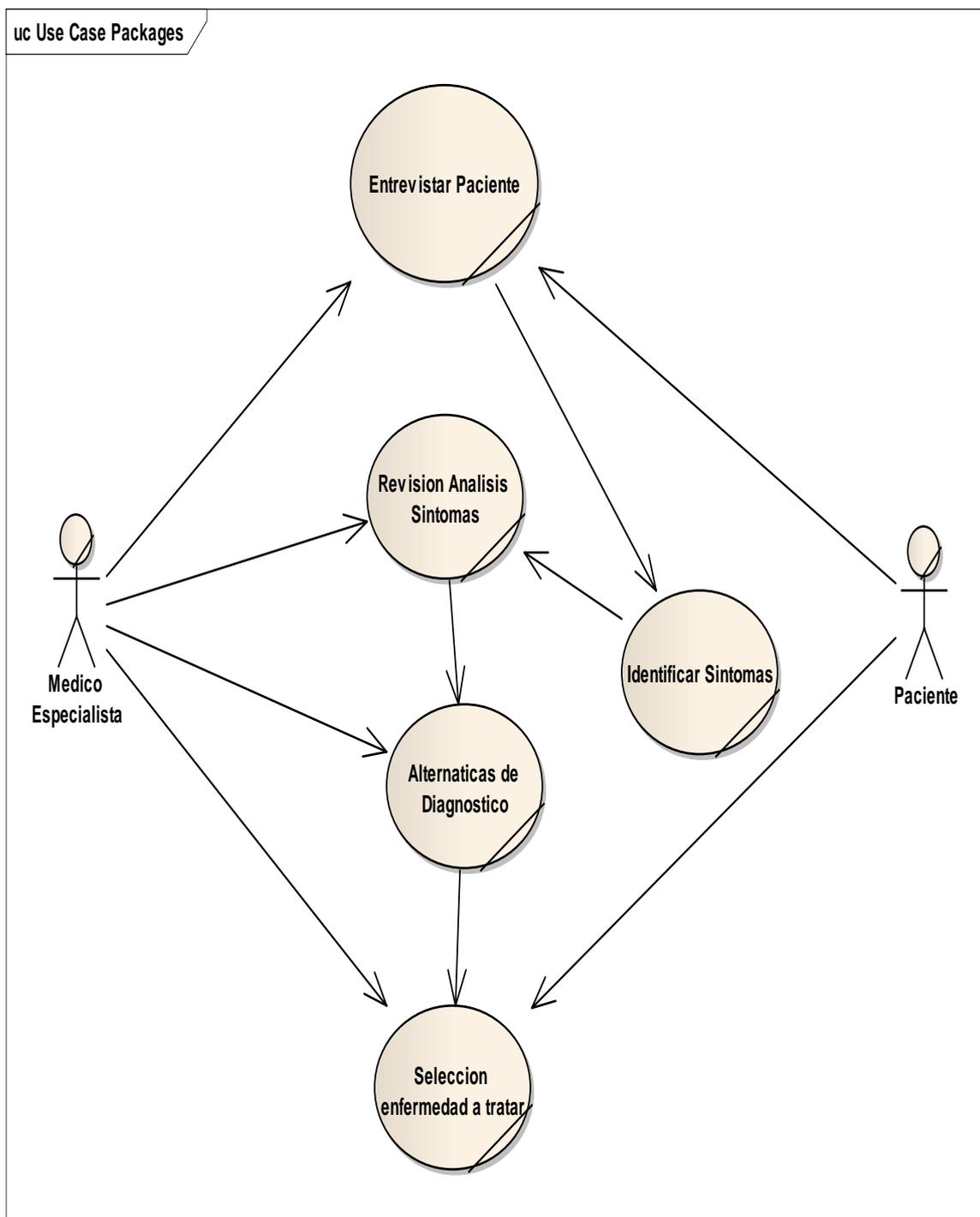
Código	Requerimiento	Tipo	Dificultad
RF001	Al Doctor especialista recolecta síntomas de un paciente determinado(niño, mujer, embarazada)	Evidente	Media
RF002	Al Doctor especialista analiza, investiga causas para dar diagnóstico	Evidente	Media
RF003	Al Doctor, busca, visualiza y edita síntomas ingresados al sistema	Evidente	Media
RF004	Al doctor especialista busca el medicamento apropiado	Evidente	Media
RF005	Al Doctor especialista analiza las contraindicaciones si desconoce algo investiga	Evidente	Alta

**Elaborado por: Digna R. Jaramillo P.**

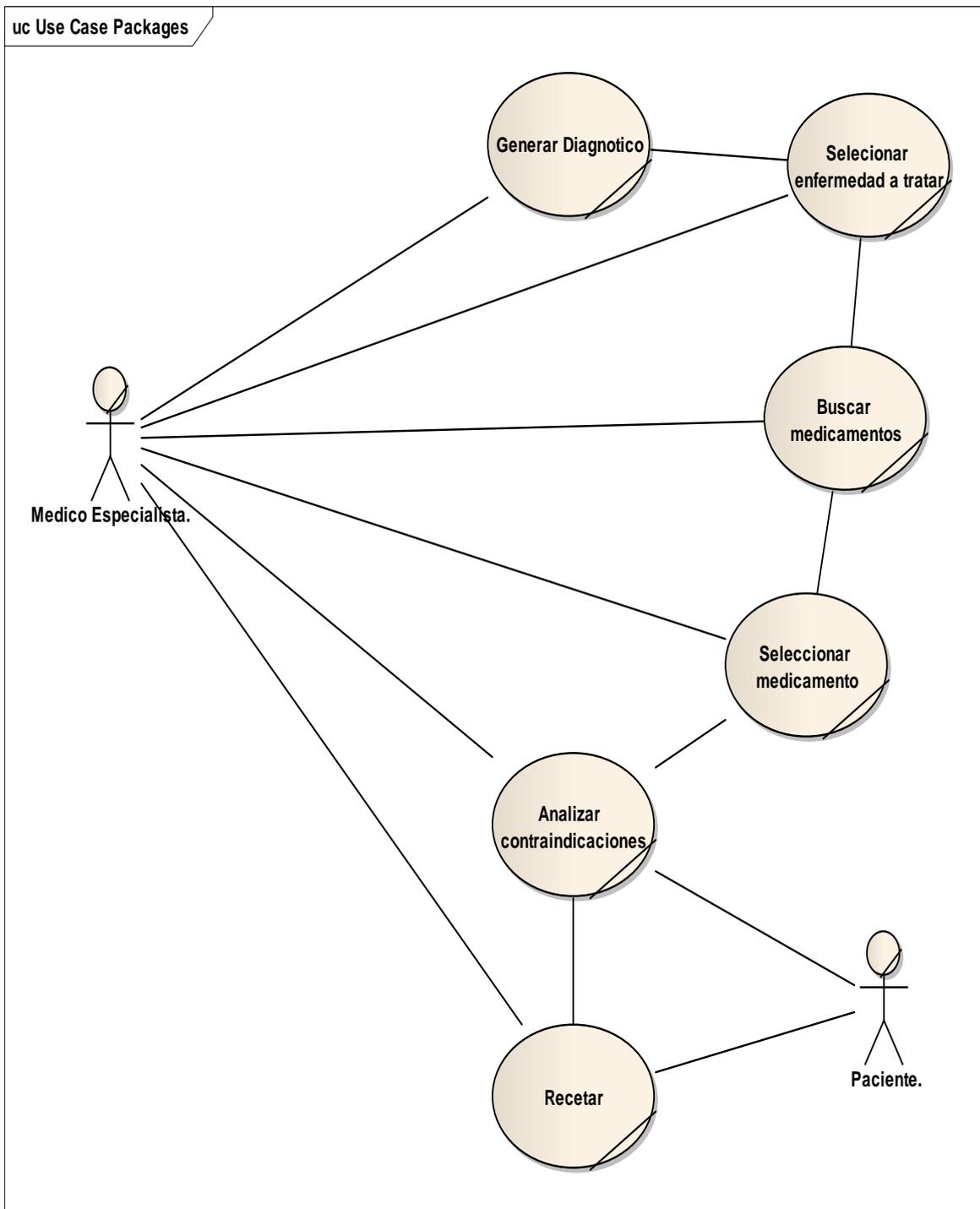
**Tabla 10.** Requerimientos no funcionales

<b>Código</b>	<b>Requerimiento</b>	<b>Observaciones</b>
RNF001	Sistema Operativo	Multiplataforma(Windows xp/ vista/7 ),GNU/LINUX(Fedora 17,Ubuntu 12.10)MAC OS (LEOPARD O SUPERIOR)
RNF002	Lenguaje de Programación	Java JSE ORACLE MICROSYSTEMS (JSE7.0 UX)
RNF003	Motor de Base de Datos	Mysql Server 5.X ORACLE MICROSYSTEMS

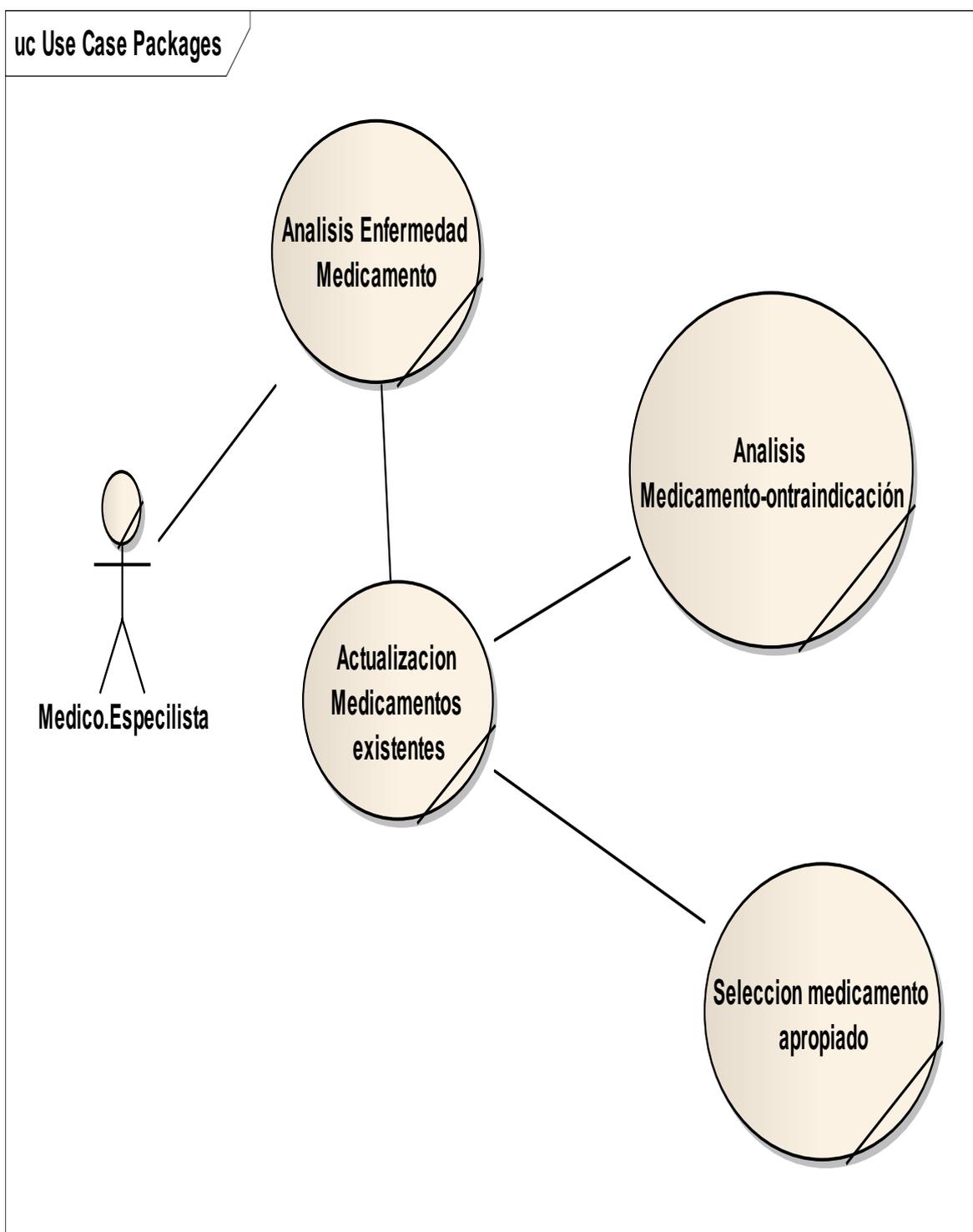
**Elaborado por: Digna R. Jaramillo P.**

**Diagrama 1.** Caso de uso de negocio Diagnosticar Paciente

Elaborado por: Digna R. Jaramillo P.

**Diagrama 2.** Caso de uso de negocio Gestionar contraindicaciones

Elaborado por: Digna R. Jaramillo P.

**Diagrama 3.** Caso de uso de negocio Recetar medicamento

Elaborado por: Digna R. Jaramillo P.

## 4.2. DETERMINACIÓN REQUERIMIENTOS DEL SISTEMA

### 4.2.1. Requerimientos Funcionales

El sistema permitirá:

**Tabla 11:** Requerimientos funcionales

Código	Requerimiento	Tipo	Dificultad
RF001	Al Doctor especialista ingresar al sistema de forma segura usando su usuario y contraseña	Evidente	Media
RF002	Al Doctor especialista crear, modificar, buscar y elimina los medicamentos ingresados en el sistema	Evidente	Media
RF003	Al Doctor, busca, visualiza y edita síntomas ingresados al sistema	Del sistema	Media
RF004	Al doctor especialista selecciona los síntomas presentados por el paciente desde un listado pudiendo agregar nuevos en caso de no existir	Evidente	Media
RF005	Al Doctor especialista obtiene un listado de posibles diagnósticos basado en los síntomas previamente seleccionados, desde un motor de inferencia, y basados en una base de conocimientos previamente preparadas	Del sistema	Alta

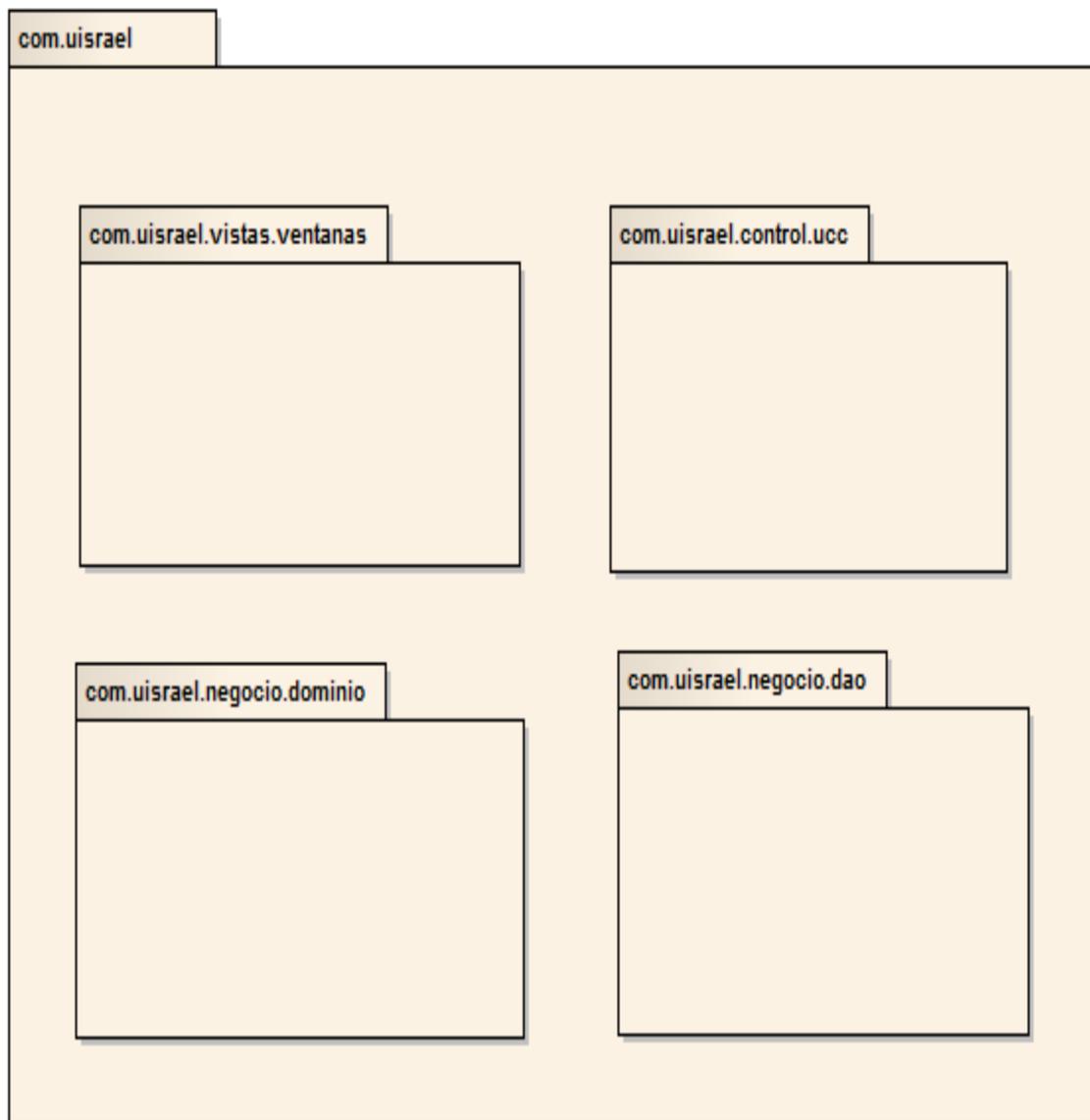
RF006	Al Doctor especialista obtener una base de conocimiento modificable y fácil de respaldar	Evidente	Baja
RF007	Al Doctor especialista ingresar las contraindicaciones para cada uno de los medicamentos presentes en el sistema	Evidente	Media
RF008	Al Doctor especialista previo seleccionar de un listado el diagnóstico más acertado según su criterio experto	Evidente	Alta
RF009	Al Doctor especialista seleccionar las contraindicaciones presentadas por el paciente desde un listado.	Evidente	Media
RF010	El Doctor especialista obtener un listado de medicamentos sugeridos por el sistema basado en el diagnóstico y contraindicaciones seleccionadas por el mismo	Evidente	Media
RF011	Al Doctor especialista generar la receta acorde a los datos del medicamento seleccionado	Evidente	Media
RF012	Al doctor especialista obtener como resultado una enfermedad vinculada al diagnostico	Del Sistema	Media

**Elaborado por: Digna R. Jaramillo P.**

**Tabla 12:** Requerimientos no funcionales

<b>Código</b>	<b>Requerimiento</b>	<b>Observaciones</b>
RNF001	Sistema Operativo	Multiplataforma (Windows xp/vista/7),GNU/LINUX (Fedora 17,Ubuntu 12.10) MAC OS (LEOPARD O SUPERIOR)
RNF002	Lenguaje de Programación	Java JSE ORACLE MICROSYSTEMS (JSE7.0 UX)
RNF003	Motor de Base de Datos	Mysql Server 5.X ORACLE MICROSYSTEMS
RNF004	Motor de Inferencias	Weca version 3.6.3 The University of Waikato
RNF005	Equipo computacional	Computador basado en procesador Intel I3 primera generación, o superior, 2 gb de ram, 10gb d espacio en disco duro
RNF00	IDE de desarrollo	NetBeans 6.9.1- ORACLE MICROSYSTEMS

**Elaborado por: Digna R. Jaramillo P.**

**Diagrama 4.** Diagrama de Paquetes

Elaborado por: Digna R. Jaramillo P.

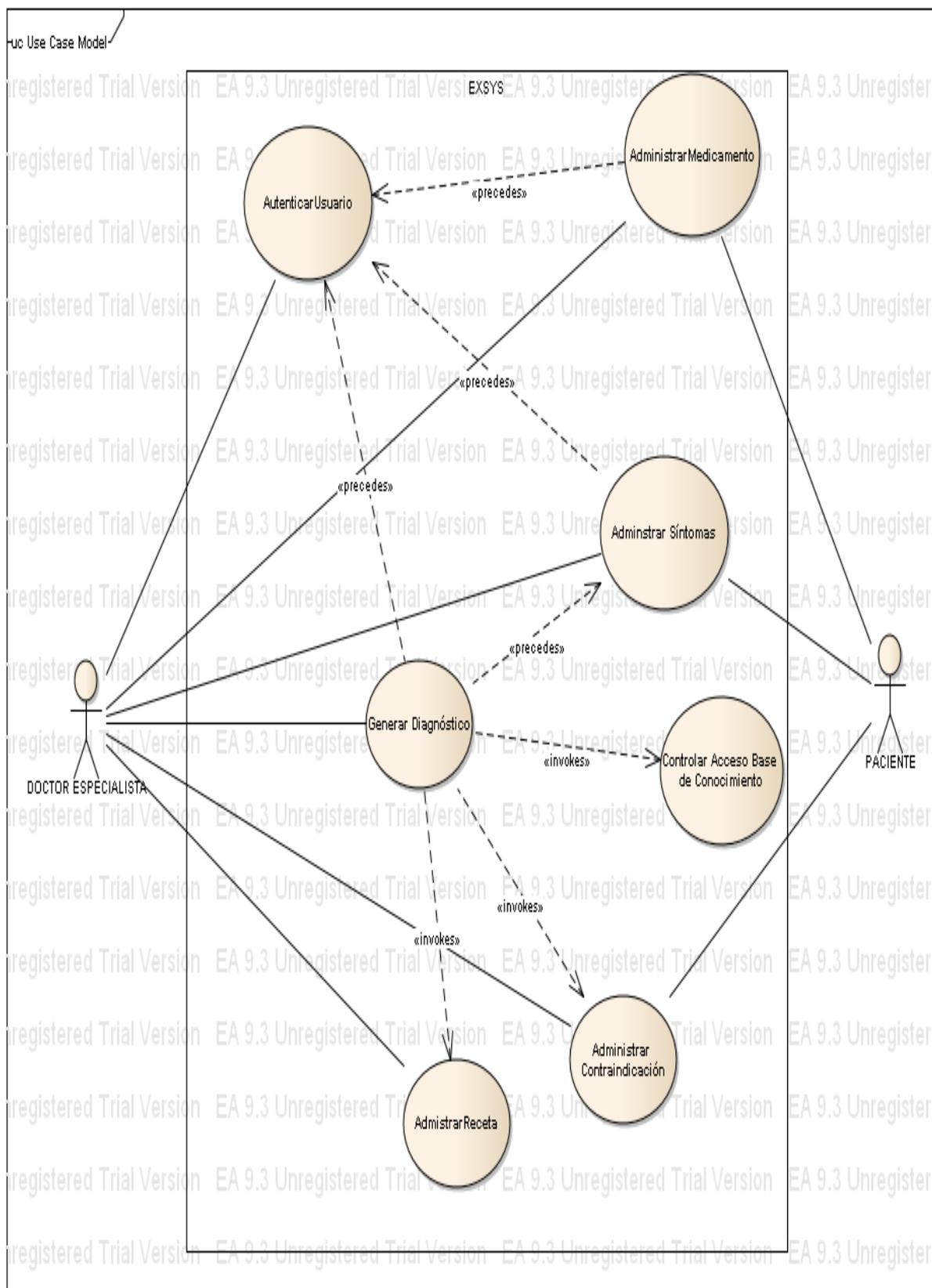
**Tabla 13:** Reconocimiento de casos de uso

Cod.	Actor	Paquete	Metas	Caso de Uso
UC001	 Doctor Especialista	Com.uisr ael.contro l.ucc	- ingresar al sistema de forma segura usando su usuario y contraseña	Cod. requ. RF001 
Cod.	Actor	Paquete	Metas	Caso de Uso
UC002	 Doctor Especialista   Paciente	Com.uisr ael.contro l.ucc	- Crear, modificar, buscar y eliminar los medicamentos - Seleccionar medicamentos adecuado	Cod. requ. RF002, RF010 
Cod.	Actor	Paquete	Metas	Caso de Uso
UC003	 Doctor Especialista   Paciente	Com.uisr ael.contro l.ucc	- buscar, visualizar y editar síntomas	Cod. requ. RF003, RF004 

Cod.	Actor	Paquete	Metas	Caso de Uso	
US004	 Doctor Especialista	com.uisra el.control. ucc	- Obtener un listado de posibles diagnósticos basado en los síntomas previamente seleccionados - Seleccionar diagnóstico más acertado	Cod. requ.	
				RF005, RF008	
	 Paciente				
Cod.	Actor	Paquete	Metas	Caso de Uso	
UC005		Com.uisr ael.contro l.ucc	- Controlar acceso a base de conocimiento.	Cod. requ.	
				RF006	
Cod.	Actor	Paquete	Metas	Caso de Uso	
UC006	 Doctor Especialista	Com.uisr ael.contro l.ucc	- Crear, modificar, buscar y eliminar las contraindicaciones.	Cod. requ.	
				RF007, RF009	
	 Paciente				

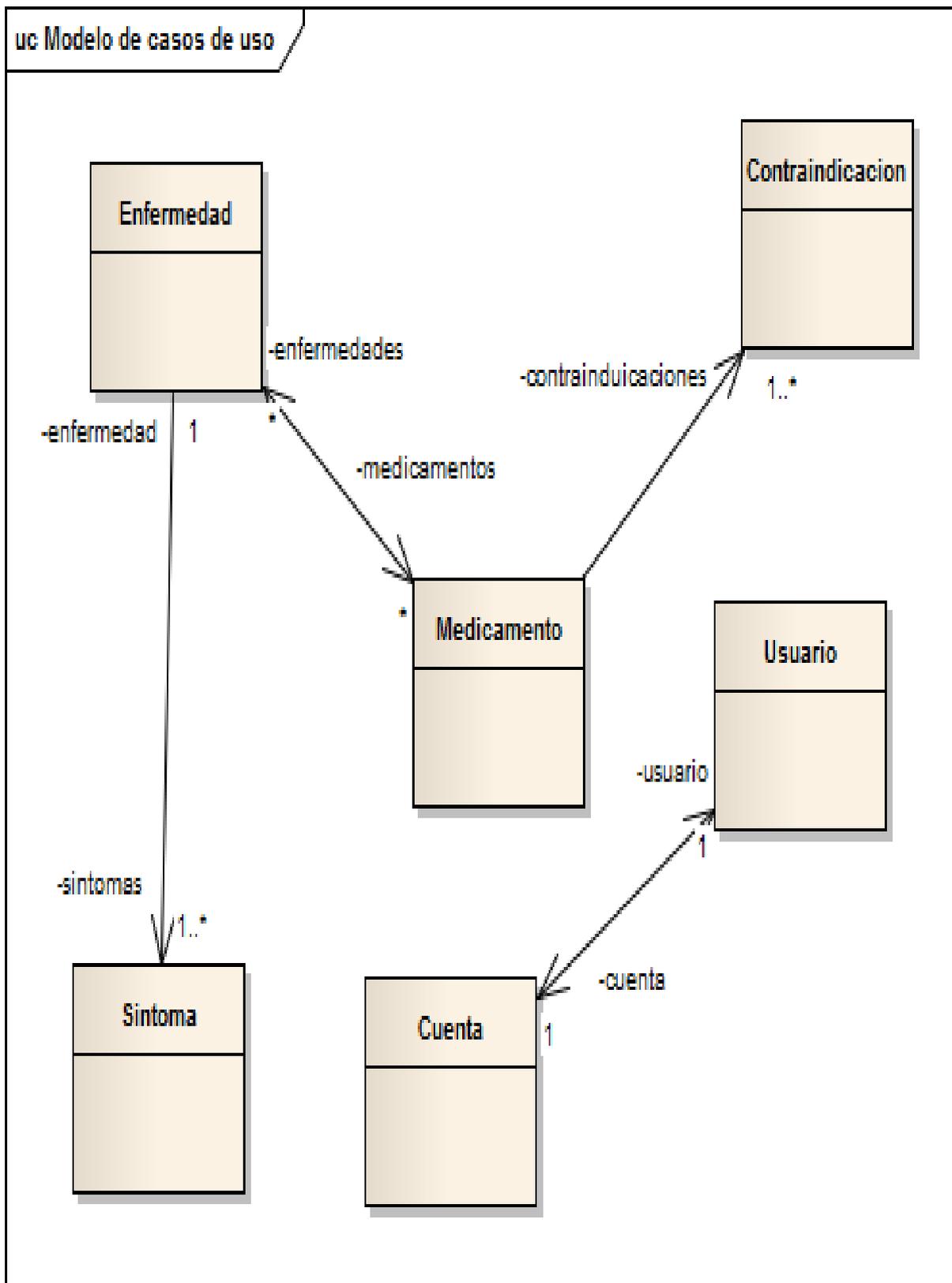
			- Ingresar contraindicacion es previo selección de medicamento		
Cod.	Actor	Paquete	Metas		Caso de Uso
UC008	 Doctor Especialista	Com.uisr ael.contro l.ucc	- Generar Receta	Cod. requ. RF011	Administrar Receta
Cod.	Actor	Paquete	Metas		Caso de Uso
UC009	 Doctor Especialista   Paciente	Com.uisr ael.contro l.ucc	- Obtener como resultado el diagnóstico de la enfermedad	Cod. requ. RF012	Administrar Enfermedad

Elaborado por: Digna R. Jaramillo P.

**Diagrama 5.** Diagrama de Casos de Uso General

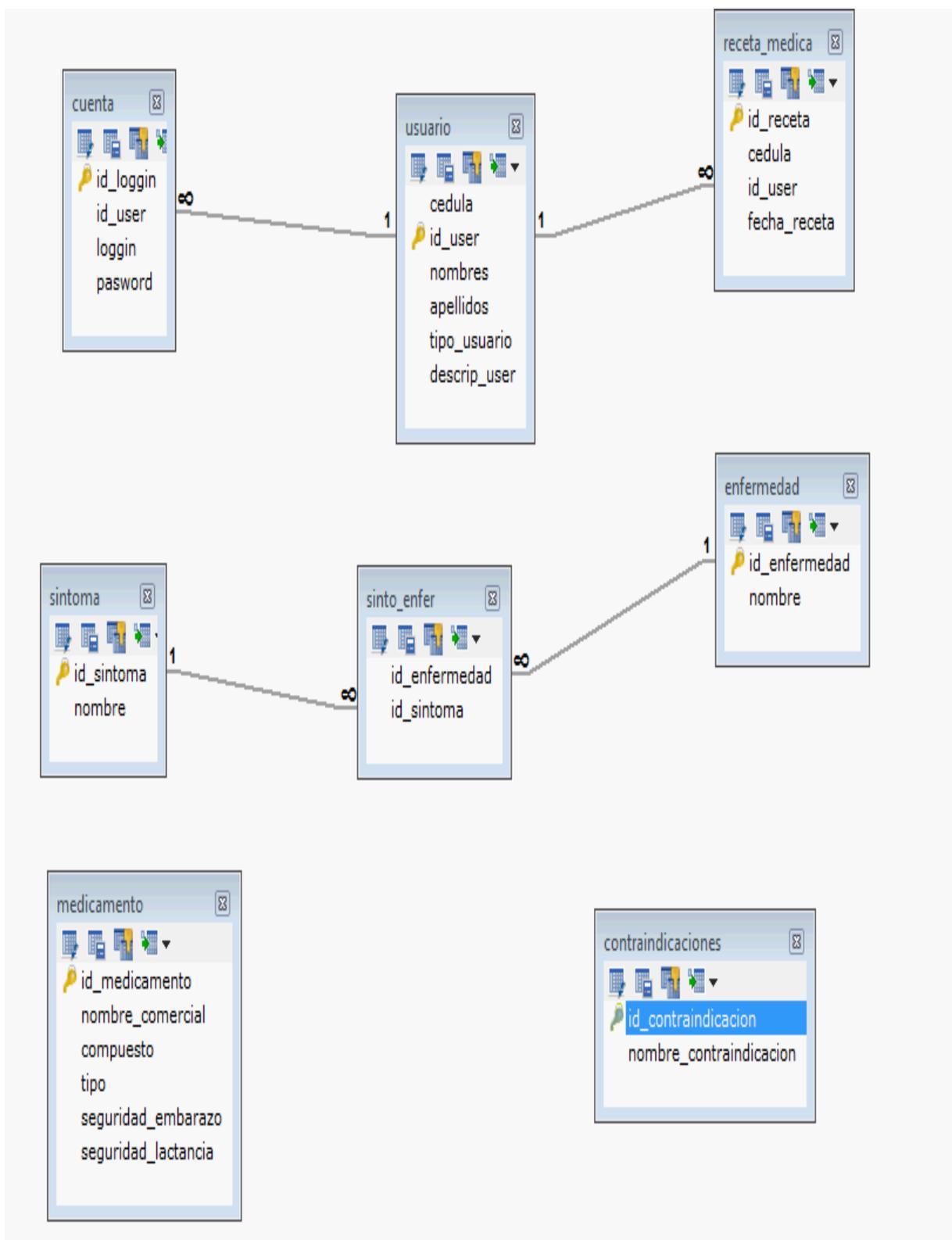
Elaborado por: Digna R. Jaramillo P.

Diagrama 6. Modelo de Dominio

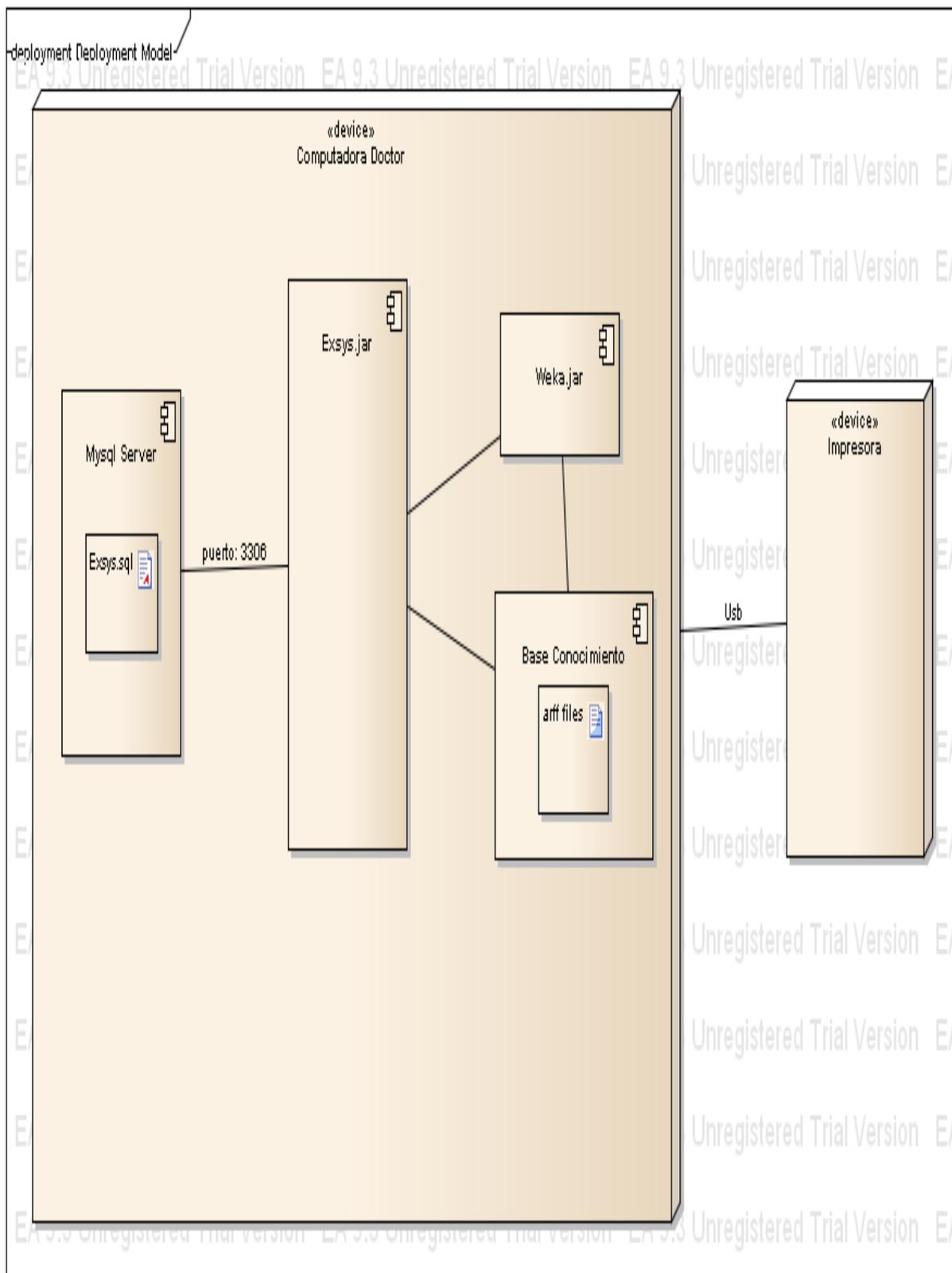


Elaborado por: Digna R. Jaramillo P.

**Diagrama 7.** Modelo Entidad-Relación / Base de Datos

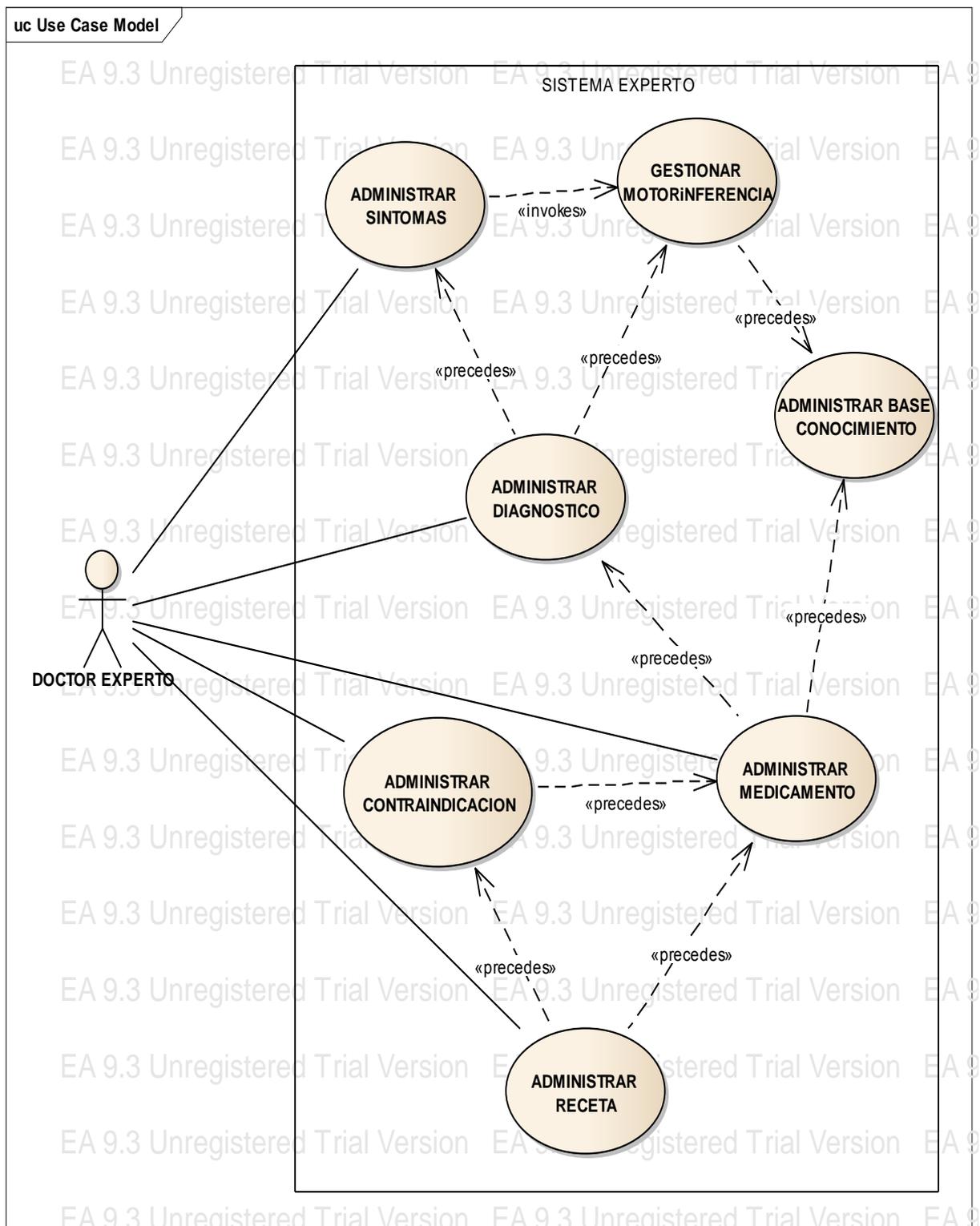


Elaborado por: Digna R. Jaramillo P.

**Diagrama 8.** Diagrama de Despliegue

Elaborado por: Digna R. Jaramillo P.

Diagrama 9. Modelo de caso de uso



Elaborado por: Digna R. Jaramillo P.

### 4.3. HISTORIA DE USUARIO

**Tabla 14:** Desarrollo de Modulo de sintomatología para el paciente

<b>Historia de usuario</b>	
<b>Numero:</b> 1	<b>Usuario:</b> Doctor Carlos Celi
<b>Nombre de la historia</b>	Desarrollo de Modulo de sintomatología para el paciente
<b>Prioridad en negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Bajo
<b>Programador Responsable:</b> Raquel Jaramillo	
<b>Descripción:</b> Presentar al usuario una pantalla en donde pueda seleccionar los síntomas que desea ingresar al sistema	
<b>Observación:</b> Esta pantalla es de vital importancia porque es el medio para ingresar los síntomas válidos para el sistema	
<b>Confirmado:</b> Aceptado por el usuario	

Elaborado por: Digna R. Jaramillo P.

### 4.4. TAREA DEL MODELO DE SINTOMATOLOGÍA

**Tabla 15:** Desarrollo de la pantalla para el ingreso de síntomas al sistema

<b>Tarea</b>	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 1
<b>Nombre de la tarea</b>	Desarrollo de la pantalla para el ingreso de síntomas al sistema
<b>Tipo de tarea:</b> Desarrollo	
Fecha de inicio: 22-12-2012	<b>Fecha fin:</b> 24-12-2012
<b>Descripción:</b> esta ventana nos va permitir mostrar al usuario los síntomas soportados por el sistema previo a la petición de diagnostico	
<b>Programador Responsable:</b> Raquel Jaramillo	

Elaborado por: Digna R. Jaramillo P.

**Tabla 16:** Investigación de los principales síntomas a los que el sistema dará soporte, previamente validados por el médico especialista

<b>Tarea</b>	
<b>Número de tarea:</b> 2	<b>Numero de historia:</b> 1
<b>Nombre de la tarea</b>	Investigación de los principales síntomas a los que el sistema dará soporte, previamente validados por el médico especialista
<b>Tipo de Tarea:</b> investigación	
<b>Fecha de inicio:</b> 22-12-2012	<b>Fecha fin:</b> 23-12-2012
<b>Descripción:</b> encontrar los síntomas a los que se dará soporte en la aplicación, identificando y definiéndolos con el nombre previamente validado por el médico especialista	
<b>Programador responsable:</b> Raquel Jaramillo	

**Elaborado por:** Digna R. Jaramillo P.

#### 4.5. PROTOTIPADO DE PANTALLA INGRESO DE SÍNTOMAS

Figura 15. Prototipado de pantalla ingreso de síntomas

Tesis Raquel Jaramillo

Archivo Visualizar Ayuda

Sintomatología Cerrar sesión

DOCTOR

Medicamentos

Buscar Diagnosticar Salir

Información del Paciente:

Fecha de ingreso:  Hora de ingreso:

Historia Clínica:

Nombres:

Edad:  años Sexo:  Estado Civil:

Tipo de paciente:

Especialidad:

Identificación:

Buscar Síntomas:

Posibles Síntomas:

- Orina frecuente, y en grandes cantidades
- Sed excesiva
- Hambre excesiva
- Pérdida de peso repentino sin causa aparente
- Debilidad
- Somnolencia
- Visión borrosa
- Náuseas
- Vómitos
- Sensación de cansancio
- Infecciones frecuentes, encías u orina
- Hormigueo en manos y pies
- Entumecimiento en manos y pies
- Picazón en la piel y genitales

Seleccionar o agregar nuevo sintoma

- Orina frecuente, y en grandes cantidades
- Sed excesiva
- Hambre excesiva
- Pérdida de peso repentino sin causa aparente
- Debilidad
- Somnolencia
- Visión borrosa
- Náuseas
- Vómitos
- Sensación de cansancio
- Infecciones frecuentes, encías u orina
- Hormigueo en manos y pies
- Entumecimiento en manos y pies
- Picazón en la piel y genitales
- Cortaduras y heridas que tardan en cicatrizar
- Piel seca
- Disminución de la cantidad de orina
- Ausencia de gasto Urinario
- Micción excesiva durante la noche
- Hinchazón de tobillos, pies y piernas
- Hinchazón generalizada por retención de líquido
- Disminución de la sensibilidad, especialmente en
- Inapetencia
- Sabor metálico en la boca
- Hipo persistente
- Agitación
- Delirio o confusión
- Coma

Elaborado por: Digna R. Jaramillo P.

DESCRIPCIÓN: esta pantalla permite al doctor especialista seleccionar los síntomas presentados por el paciente, seleccionándolos uno a uno previo a una petición de diagnóstico.

#### 4.6. BASE CONOCIMIENTO DEL SISTEMA

**Tabla 17:** Desarrollo de Base conocimientos sistema experto

<b>Historia de usuario</b>	
<b>Numero:</b> 2	<b>Usuario:</b> Doctor Carlos Celi
<b>Nombre de la historia</b>	Desarrollo de Base conocimientos sistema experto
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alto
<b>Programador Responsable:</b> Raquel Jaramillo	
<b>Descripción:</b> Provee al sistema de la base de conocimiento inicial en base a la cual hará la toma de decisiones.	
<b>Observación:</b> Como resultado de este proceso se obtiene los archivos ARFF para weka.	
<b>Confirmado:</b> Aceptado por el usuario	

**Elaborado por:** Digna R. Jaramillo P.

#### 4.7. TAREA DEL DESARROLLO BASE DE CONOCIMIENTO DEL SISTEMA

**Tabla 18:** Investigación Estructura archivos arff de weka, y determinación de la estructura de los archivos necesarios para el experto.

Tarea	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 2
<b>Nombre de la tarea</b>	Investigación Estructura archivos arff de weka, y determinación de la estructura de los archivos necesarios para el experto.
<b>Tipo de tarea:</b> Análisis y diseño	
Fecha de inicio:	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de esta actividad se debe conocer la estructura y la cantidad de archivos necesarios para la base de conocimiento del experto	
<b>Programador Responsable:</b> Raquel Jaramillo	

**Elaborado por: Digna R. Jaramillo P.**

**Tabla 19:** Selección de enfermedades e Investigación de los síntomas y características para la codificación e la base de conocimiento.

Tarea	
<b>Número de tarea:</b> 2	<b>Numero de historia:</b> 2
<b>Nombre de la tarea</b>	Selección de enfermedades e Investigación de los síntomas y características para la codificación e la base de conocimiento.
<b>Tipo de Tarea:</b> investigación	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de esta tarea esperamos obtener la información necesaria para la construcción de los archivos arff	
<b>Programador responsable:</b> Raquel Jaramillo	

**Elaborado por: Digna R. Jaramillo P.**

**Tabla 20:** Codificación de archivos arff

Tarea	
<b>Número de tarea:</b> 3	<b>Numero de historia:</b> 2
<b>Nombre de la tarea</b>	Codificación de archivos arff
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de esta tarea esperamos obtener los archivos arff para el sistema experto	
<b>Programador responsable:</b> Raquel Jaramillo	
<b>Elaborado por:</b> Digna R. Jaramillo P.	

**Tabla 21:** Codificación de archivos arff (obtener)

Tarea	
<b>Número de tarea:</b> 4	<b>Numero de historia:</b> 2
<b>Nombre de la tarea</b>	Codificación de archivos arff
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de esta tarea se obtiene los archivos arff para el sistema experto	
<b>Programador responsable:</b> Raquel Jaramillo	
<b>Elaborado por:</b> Digna R. Jaramillo P.	

**Tabla 22:** Desarrollo de módulo para la lectura escritura de kis archivos arff

Tarea	
<b>Número de tarea:</b> 5	<b>Numero de historia:</b> 2
<b>Nombre de la tarea</b>	Desarrollo de módulo para la lectura escritura de kis archivos arff
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de esta tarea esperamos los archivos arff para el sistema experto	
<b>Programador responsable:</b> Raquel Jaramillo	
<b>Elaborado por:</b> Digna R. Jaramillo P.	

#### 4.8. DESARROLLO DEL MODULO DE DIAGNOSTICO DE ENFERMEDADES

**Tabla 23:** Desarrollo del módulo de diagnóstico experto

Historia de usuario	
<b>Numero:</b> 3	<b>Usuario:</b> Doctor Carlos Celi
<b>Nombre de la historia</b>	Desarrollo del módulo de diagnóstico experto
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alta
<b>Programador Responsable:</b> Raquel Jaramillo	
<b>Descripción:</b> Como resultado de este proceso se obtiene el módulo que permitirá en base a la base de conocimiento obtener un diagnóstico.	
<b>Observación:</b> La selección e implementación de un algoritmo de búsqueda inteligente con el que se implemente una capa de igual a naturaleza en el sistema	
<b>Confirmado:</b> Aceptado por el usuario	

**Elaborado por:** Digna R. Jaramillo P.

**Tabla 24:** Comparar algoritmos de búsqueda inteligente soportados por weka

<b>Tarea</b>	
<b>Número de tarea:</b> 1	<b>Numero de historia:</b> 3
<b>Nombre de la tarea</b>	Comparar algoritmos de búsqueda inteligente soportados por weka
<b>Tipo de Tarea:</b> Análisis	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de esta tarea se determinó que el algoritmo más idóneo para un sistema es el multilayer perceptron, el cual posibilita la implementación de un red neuronal.	
<b>Programador responsable:</b> Raquel Jaramillo	

**Elaborado por:** Digna R. Jaramillo P.

**Tabla 25:** Implantación del algoritmo dentro del sistema experto

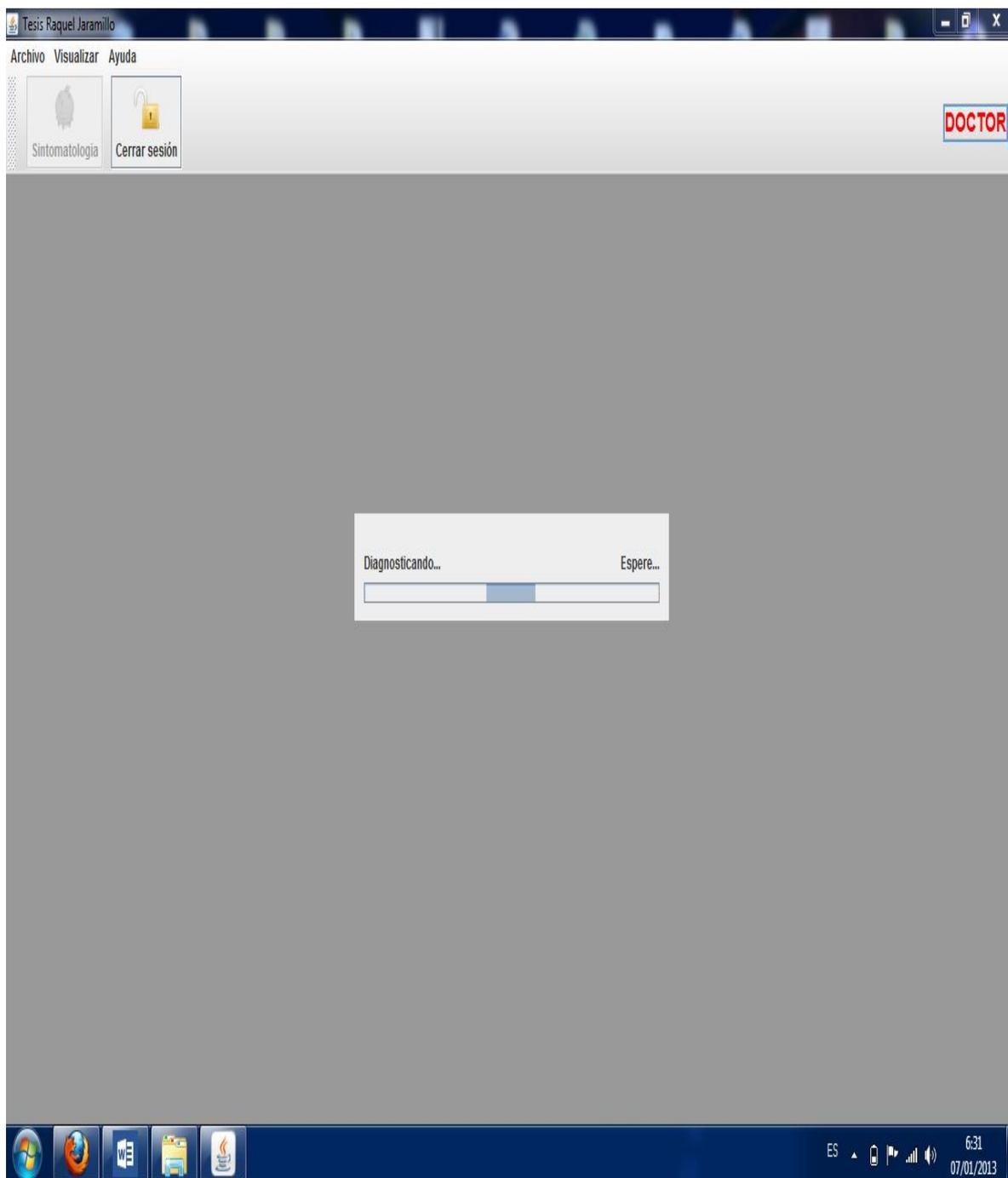
<b>Tarea</b>	
<b>Número de tarea:</b> 2	<b>Numero de historia:</b> 3
<b>Nombre de la tarea</b>	Implantación del algoritmo dentro del sistema experto
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de esta tarea se obtiene la red neuronal implementada en el sistema experto	
<b>Programador responsable:</b> Raquel Jaramillo	
<b>Elaborado por:</b> Digna R. Jaramillo P.	

**Tabla 26:** Implantación del algoritmo dentro del sistema experto

<b>Tarea</b>	
<b>Número de tarea:</b> 3	<b>Numero de historia:</b> 3
<b>Nombre de la tarea</b>	Implantación del algoritmo dentro del sistema experto
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de esta tarea se obtiene la red neuronal para en el sistema experto	
<b>Programador responsable:</b> Raquel Jaramillo	
<b>Elaborado por:</b> Digna R. Jaramillo P.	

#### 4.9. PROTOTIPOS DE PANTALLAS PARA MODULO DE DIAGNOSTICO DE ENFERMEDADES

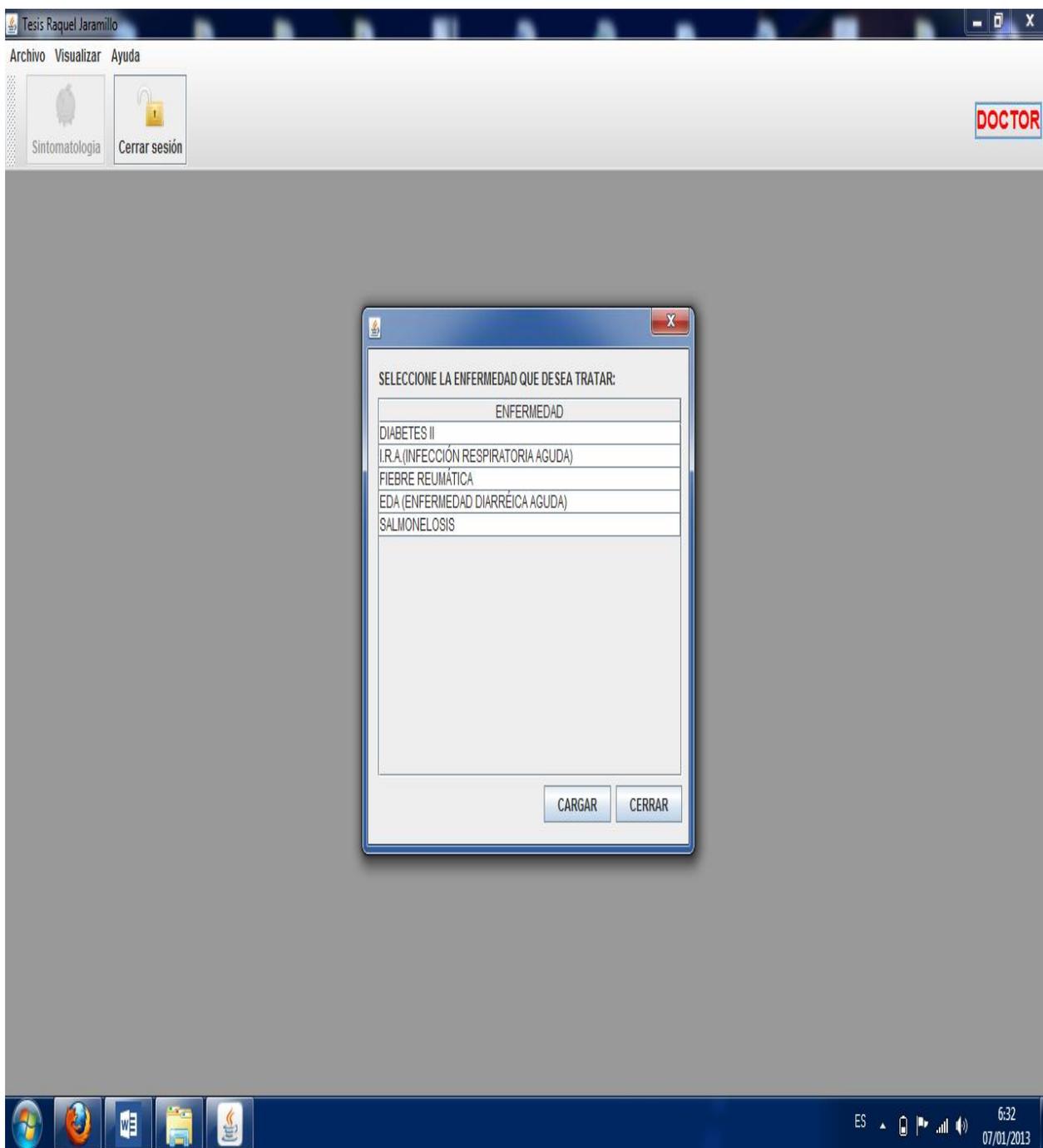
**Figura 16.** Diagnóstico de enfermedades



Elaborado por: Digna R. Jaramillo P.

Seleccionamos Los síntomas y el sistema automáticamente da inicio a las consultas para dar un diagnostico respectivo.

**Figura 17.** Posibles enfermedades-paciente.



Elaborado por: Digna R. Jaramillo P.

Al momento que seleccionamos los síntomas el sistema nos arroja las opciones de las posibles enfermedades que tiene el paciente.

#### 4.10. DESARROLLO PARA MÓDULO DE CONTRAINDICACIONES Y RECETA DE MEDICAMENTOS

**Tabla 27:** Desarrollo del módulo de manejo de contraindicaciones y receta de medicamentos.

<b>Historia de usuario</b>	
<b>Numero:</b> 4	<b>Usuario:</b> Doctor Carlos Celi
<b>Nombre de la historia</b>	Desarrollo del módulo de manejo de contraindicaciones y receta de medicamentos.
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alta
<b>Programador Responsable:</b> Raquel Jaramillo	
<b>Descripción:</b> Como resultado de este proceso se obtiene el módulo que permitirá el ingreso de las contraindicaciones presentadas por el paciente y en base a estas poder obtener una lista de medicamentos sugeridos para la receta.	
<b>Observación:</b> Algoritmo de búsqueda entre los medicamentos existentes en la base de datos del sistema usando como restricciones las contraindicaciones presentadas por el paciente.	
<b>Confirmado:</b> Aceptado por el usuario	

**Elaborado por:** Digna R. Jaramillo P.

**Tabla 28:** Investigación de las contraindicaciones a ser tomadas en cuenta por el Sistema

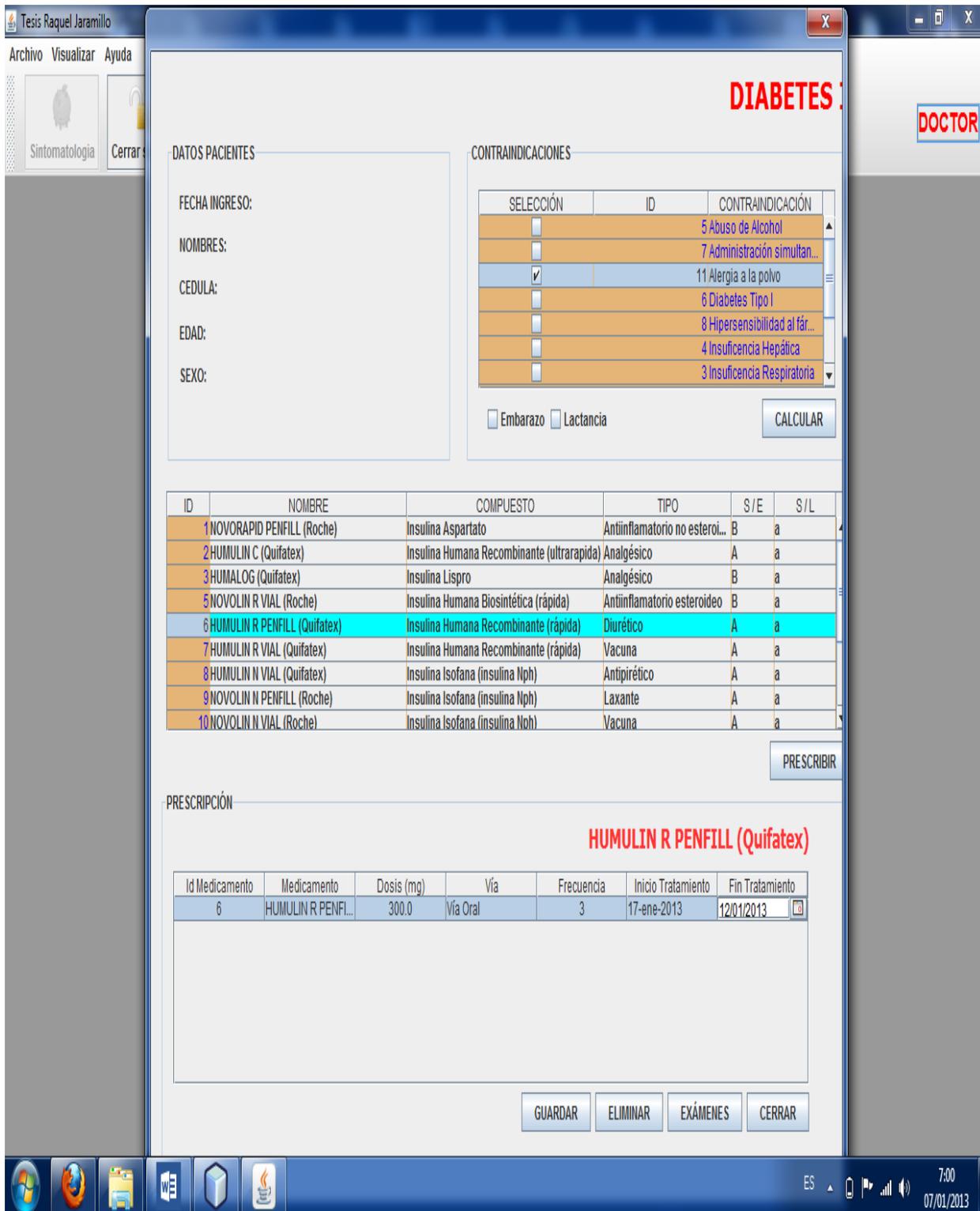
<b>Tarea</b>	
<b>Número de tarea:</b> 1	<b>Numero de historia:</b> 4
<b>Nombre de la tarea</b>	Investigación de las contraindicaciones a ser tomadas en cuenta por el sistema
<b>Tipo de Tarea:</b> Análisis	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Como resultado de este proceso se obtiene el listado de las contraindicaciones válidas para el sistema.	
<b>Observación:</b> Contraindicaciones validadas por el experto Humano.	
<b>Elaborado por:</b> Digna R. Jaramillo P.	

**Tabla 29:** Desarrollo módulo de contraindicaciones y medicamentos para la receta

<b>Tarea</b>	
<b>Número de tarea:</b> 2	<b>Numero de historia:</b> 4
<b>Nombre de la tarea</b>	Desarrollo módulo de contraindicaciones y medicamentos para la receta
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Permite al usuario experto la selección de las contraindicaciones a ser tomadas en cuenta por un algoritmo de búsqueda entre los medicamentos presentes en el sistema	
<b>Observación:</b> Los medicamentos deben ser ingresados previamente en el sistema.	
<b>Elaborado por:</b> Digna R. Jaramillo P.	

### 4.11. PROTOTIPADOS DE PANTALLA DE CONTRAINDICACIONES

Figura 18. Prototipados de pantalla de contraindicaciones



Elaborado por: Digna R. Jaramillo P.

Seleccionamos el medicamento apropiado para la enfermedad seleccionada y también las contraindicaciones que puede tener el mismo.

#### 4.12. DESARROLLO PARA MÓDULO DE CONTRAINDICACIONES Y RECETA DE MEDICAMENTOS

**Tabla 30:** Desarrollo del módulo de gestión de medicamentos

<b>Historia de usuario</b>	
<b>Numero:</b> 5	<b>Usuario:</b> Doctor Carlos Celi
<b>Nombre de la historia</b>	Desarrollo del módulo de gestión de medicamentos
<b>Prioridad en negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Media
<b>Programador Responsable:</b> Raquel Jaramillo	
<b>Descripción:</b> Permite la creación, modificación y eliminación de medicamentos en el sistema, así como la asignación de contraindicaciones y enfermedades a las que se trata con los mismos.	
<b>Observación:</b> Los medicamentos deben ser validados por el experto humano.	
<b>Confirmado:</b> Aceptado por el usuario	

**Elaborado por:** Digna R. Jaramillo P.

**Tabla 31:** Desarrollo módulo de administración de medicamentos.

<b>Tarea</b>	
<b>Número de tarea:</b> 1	<b>Numero de historia:</b> 5
<b>Nombre de la tarea</b>	Desarrollo módulo de administración de medicamentos.
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Permite la creación, modificación y eliminación de medicamentos en el sistema, así como la asignación de contraindicaciones y enfermedades a las que se trata con los mismos.	
<b>Observación:</b> Los medicamentos deben ser validados por el experto humano.	

**Elaborado por:** Digna R. Jaramillo P.

#### 4.13. PROTOTIPOS DE PANTALLAS PARA ADMINISTRACION DE MEDICAMENTOS

Figura 19. Crear o editar medicamento

The screenshot shows a web application window titled "Tesis Raquel Jaramillo" with a menu bar (Archivo, Visualizar, Ayuda) and navigation buttons (Sintomatología, Cerrar sesión). A "DOCTOR" button is visible in the top right. The main content is a modal window titled "MEDICAMENTO (NUEVO)" with the following sections:

- MEDICAMENTO:**
  - Nombre Comercial:
  - Compuesto:
  - Tipo de fármaco:
  - Seguridad Embarazo:
  - Seguridad Lactancia:
- POSOLOGIA:**
  - Presentación:
  - Cantidad (mg):  mg
  - Dosis:
- REL. ENFERMEDAD:**
  - Diabetes I  Fiebre Reumática  I.R.A  EDA
  - Rubeola  Salmonelosis  Diabetes II
  - Hepatitis B  Hipertensión  Hepatitis Virica
- CONTRAINDICACIONES:**

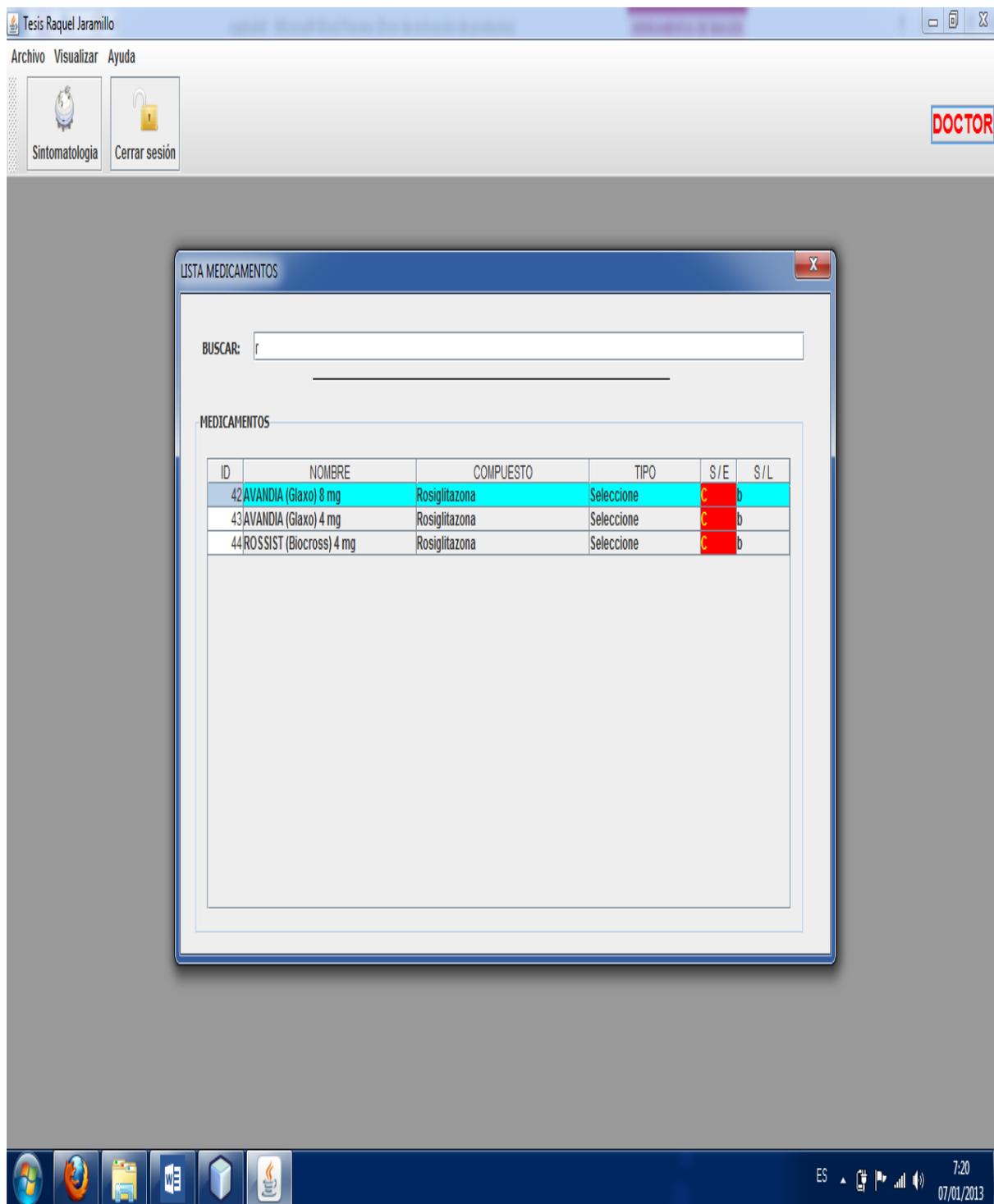
SELECCIÓN	ID	NOMBRE

The Windows taskbar at the bottom shows the system tray with the date 07/01/2013 and time 7:20.

Elaborado por: Digna R. Jaramillo P.

Esta pantalla permite crear o editar un nuevo medicamento que no se encuentre en la base de datos.

**Figura 20.** Busca de medicamento



The screenshot shows a web application interface for a doctor. At the top, there is a navigation bar with 'Archivo', 'Visualizar', and 'Ayuda' menus. Below this are icons for 'Sintomatología' and 'Cerrar sesión', and a 'DOCTOR' button. The main content area displays a 'LISTA MEDICAMENTOS' window with a search field containing the letter 'r'. Below the search field is a table of medications.

ID	NOMBRE	COMPUESTO	TIPO	S/E	S/L
42	AVANDIA (Glaxo) 8 mg	Rosiglitazona	Seleccione	C	b
43	AVANDIA (Glaxo) 4 mg	Rosiglitazona	Seleccione	C	b
44	ROSSIST (Biocross) 4 mg	Rosiglitazona	Seleccione	C	b

The Windows taskbar at the bottom shows the system tray with the time 7:20 and date 07/01/2013.

Elaborado por: Digna R. Jaramillo P.

En esta pantalla permitirá buscar el medicamento que queremos encontrar se puede hacer ingresando la primera letra o silaba del medicamento a buscar.

**Tabla 32:** Desarrollo control acceso al sistema

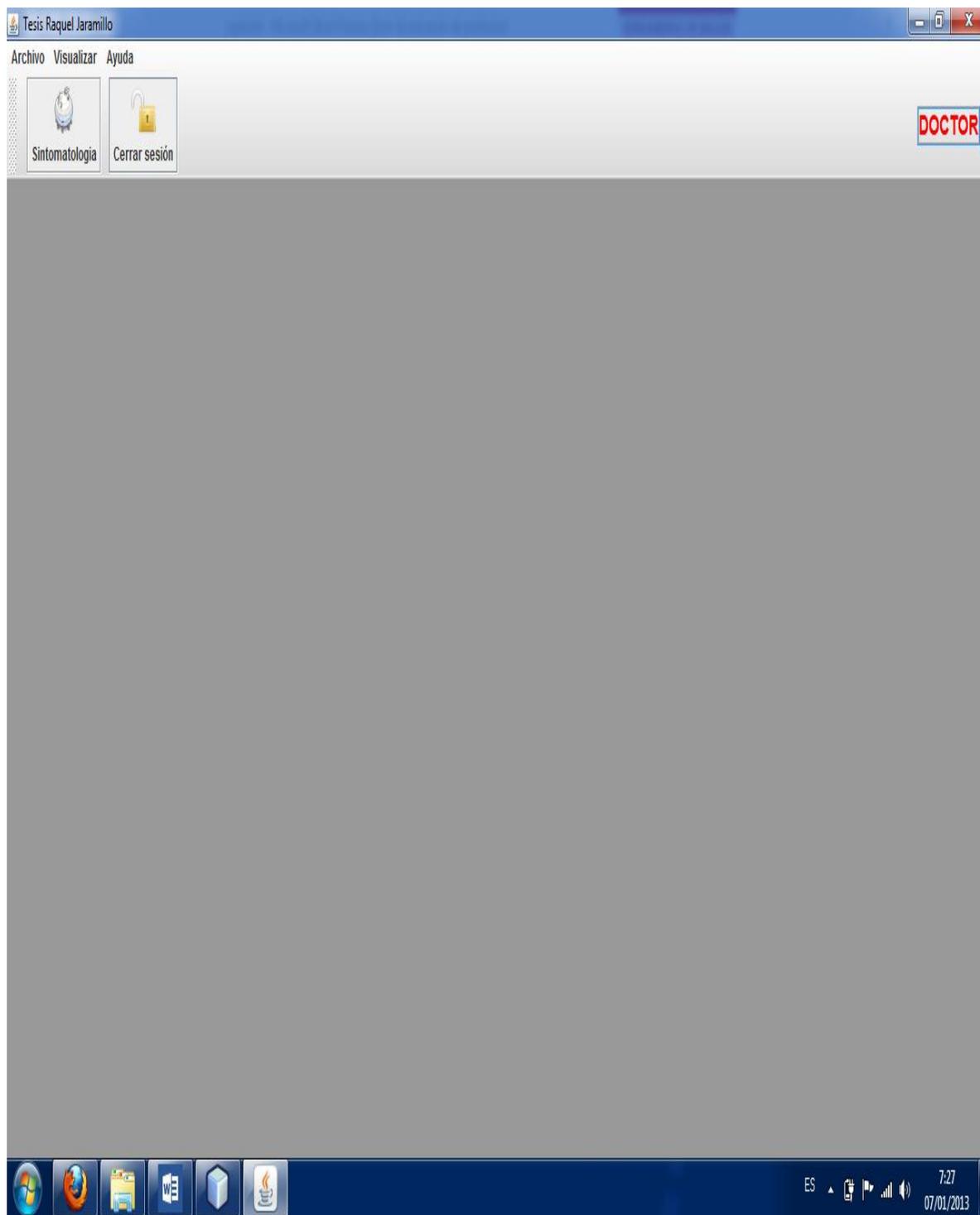
<b>Historia de usuario</b>	
<b>Numero:</b> 6	<b>Usuario:</b> Doctor Carlos Celi
<b>Nombre de la historia</b>	Desarrollo de control de acceso al sistema
<b>Prioridad en negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Media
<b>Programador Responsable:</b> Raquel Jaramillo	
<b>Descripción:</b> Permite el ingreso al sistema previo usuario y contraseña definidos en la cuenta "DOCTOR" EN LA BASE DE DATOS	
<b>Observación:</b> El acceso debe ser restringido.	
<b>Confirmado:</b> Aceptado por el usuario	

**Elaborado por:** Digna R. Jaramillo P.

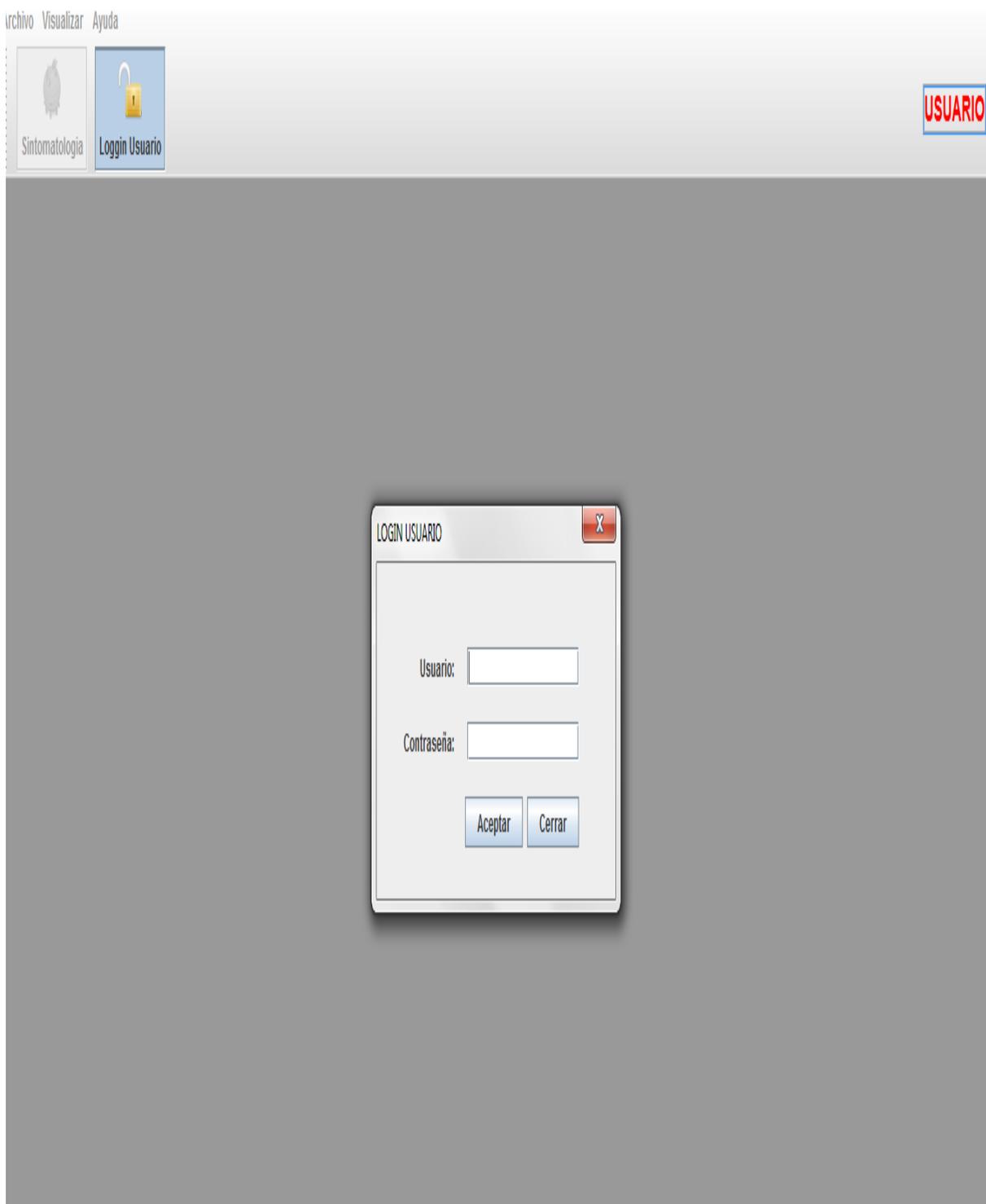
**Tabla 33:** Desarrollo módulo de control de acceso al sistema

<b>Tarea</b>	
<b>Número de tarea:</b> 1	<b>Numero de historia:</b> 6
<b>Nombre de la tarea</b>	Desarrollo módulo de control de acceso al sistema
<b>Tipo de Tarea:</b> Desarrollo	
<b>Fecha de inicio:</b>	<b>Fecha fin:</b>
<b>Descripción:</b> Permite el ingreso al sistema previo usuario y contraseña definidos en la cuenta "DOCTOR" EN LA BASE DE DATOS	
<b>Observación:</b> El acceso debe ser restringido.	

**Elaborado por:** Digna R. Jaramillo P.

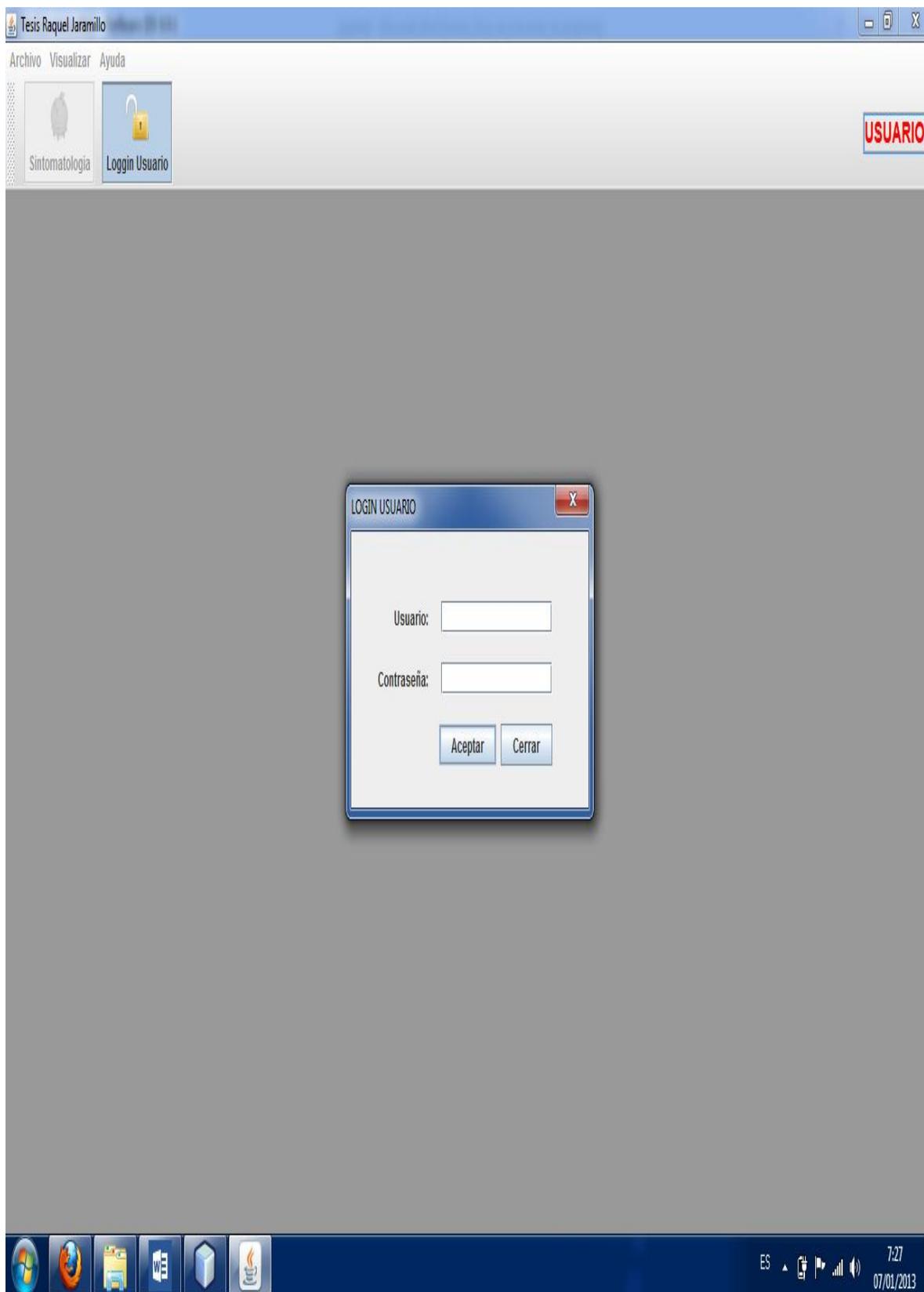
**Figura 21.** Loguin Usuario

**Elaborado por: Digna R. Jaramillo P.**

**Figura 22.** Ingreso al sistema- Pantalla principal

Elaborado por: Digna R. Jaramillo P.

Esta es la primera pantalla que aparecerá al momento de ingresar al sistema (pantalla principal)

**Figura 23.** Ingreso al sistema

Elaborado por: Digna R. Jaramillo P.

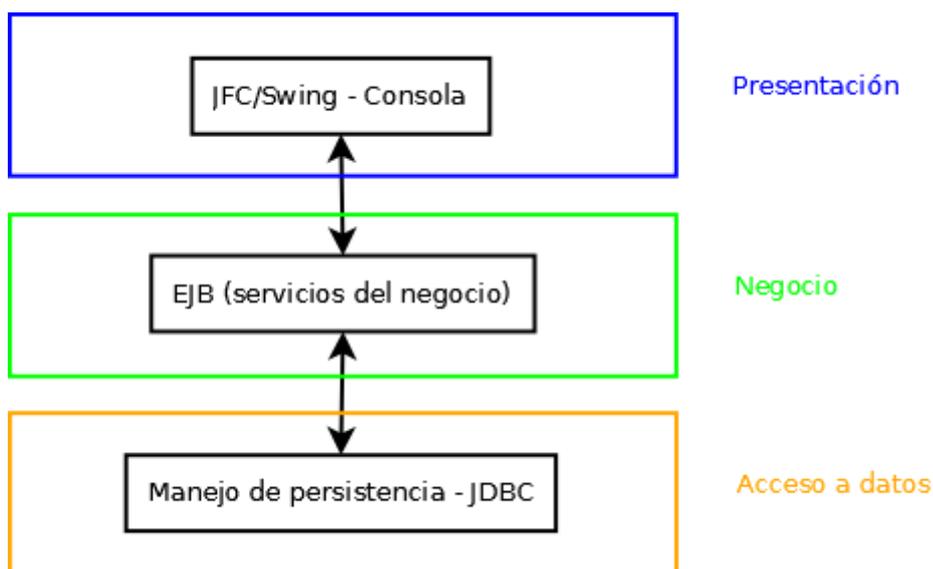
## 4.14. ARQUITECTURA

### 4.14.1. Construcción: Plan de pruebas

#### 4.14.1.1. Arquitectura cliente servidor de Tres Capas

- **Capa de presentación o interfaz de usuario:** consiste de hardware como un PC o una estación de trabajo. En esta capa puede haber cualquier número de máquinas cliente.
- **Capa Lógica de funcionalidad o del negocio:** proporciona la funcionalidad a los usuarios y es el puente entre la primera y la tercera capa.
- **Capa de Datos:** incluye la base de datos y puede colocarse cualquier número de servidores en esta capa, dependiendo del volumen de transacciones y de la cantidad de datos.

**Figura 24.** Arquitectura cliente servidor de Tres Capas



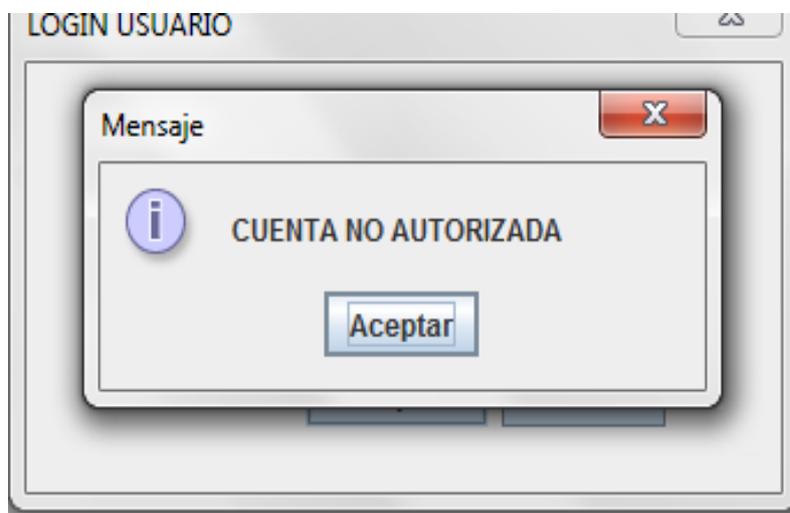
Elaborado por: Digna R. Jaramillo P.

#### 4.14.2. Transición: Pruebas

##### 4.14.2.1. Modelo de pruebas

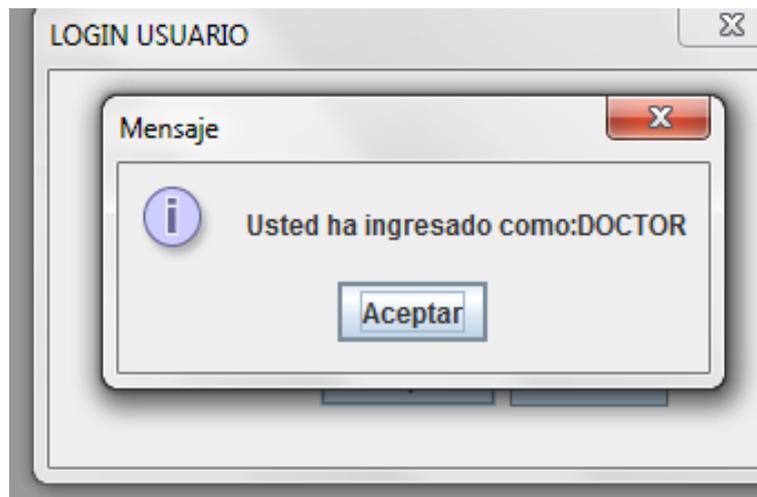
En esta etapa solo se han desarrollado pruebas de análisis basadas principalmente en los casos de uso iniciales que se han propuesto.

**Figura 25.** Usuario no registrado



**Elaborado por: Digna R. Jaramillo P.**

Ventana de acceso al sistema cuando el usuario no está registrado o cuando ha colocado una clave no es la correcta

**Figura 26.** Usuario registrado

**Elaborado por: Digna R. Jaramillo P.**

Si se ingresa datos correctamente esta será la ventana que se observa, la persona que está ingresando es la indicada.

## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1. CONCLUSIONES

- La información obtenida por parte de los expertos médicos con respecto al cerco epidemiológico fundamentalmente se implementó exitosamente en la base del conocimiento del sistema experto mediante el uso de archivos arff.
- El módulo inteligente de EXSYS es un factor importante en el ahorro de tiempo para ayuda en toma de decisiones importantes dentro de la decisión hacia cualquier paciente.
- El uso metodologías ágiles como lo es XP son orientados a la gente y no orientados al proceso. De manera que se pudo definir un proceso que funcionará bien con cualquiera que lo use.
- La librería WEKA es robusta, excelente a la hora de realizar implementaciones de sistemas inteligentes y minería de datos con sus diferentes técnicas y múltiples algoritmos que tiene.
- Con el uso de WEKA el grado de concordancia que tienen los resultados arrojados por un estudio basado en un problema de la vida cotidiana con la realidad dependen en un alto grado de la calidad y fiabilidad de los datos empleados en la fase de entrenamiento.
- Con WEKA logra solución sobre los diferentes problemas o situaciones presentadas en la realidad y luego de haber obtenido un respetivo resultado se puede tomar decisiones para el mejoramiento

## 5.2. RECOMENDACIONES

- Para el desarrollo de sistemas inteligentes se recomienda hacer el uso de la librería WEKA, para poder optimizar el tiempo de desarrollo ya que se puede obtener conclusiones y aproximaciones adecuadas y que sean útiles para el profesional a desarrollar.
- Se recomienda al hacer uso de WEKA mantener una buena fase de entrenamiento con datos el mismo que deben poseer un alto grado de calidad y confiabilidad por parte del programador.
- Trabajar con tecnologías que puedan mitigar el problema de desajuste de impedancia en las base de datos relacionales para impedir el problema de escribir demasiado código SQL.
- Trabajar constantemente con el usuario final para poder tener un software que tanto el desarrollador como el usuario pueda quedar satisfecho y seguro de la herramienta que va a utilizar de apoyo.
- Se recomienda la implementación de EXSYS en hospitales o clínicas de atención general para poder apoyarse y optar por una herramienta de ayuda en el desarrollo cotidiano de actividades de atención hospitalaria.

## BIBLIOGRAFÍA Y REFERENCIAS

- **Fuentes primarias (Libros)**

1. NEBENDAHL, Dieter. “Sistemas Expertos”, Centro de Desarrollo de SW Cornellá, 1991, Barcelona, España.
2. NILSSON, Nills J.. “Inteligencia Artificial Una Nueva Síntesis”, Editorial McGraw Hill, 2001, España.
3. Ruiz del solar., Javier. **Sistemas inteligentes, procesamiento avanzado de imágenes**, universidad de chile Prof.
4. Agentes multiagentes inteligentes: conceptos, arquitecturas y aplicaciones.
5. Salazar serrudo, Carla. Cochabamba, febrero 2003.
6. Stuart j. Russell y Peter norving, **inteligencia artificial un enfoque moderno**, segunda edición

- **Fuentes terciarias (Páginas Web)**

7. ALEGRE LOPEZ, Anita del C. “Simulación de redes neuronales artificiales”, Licenciatura en Sistemas Corrientes – Argentina, 15-12-2009  
[<http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/SimulacionRNA.PDF>]
8. ALONSO GONZALEZ, Carlos J. “C.WEKA: Waitako Environment for Knowledge Analysis”. Introducción básica. Departamento de Informática Universidad de Valladolid, 17-212-2009  
[<http://www.infor.uva.es/~calonso/IAII/Aprendizaje/weka/IntroduccionWeka>].

9. ARREDONDO VIDAL, Tomas. "Introducción a las Redes Neuronales", 17-12-2009, [<http://profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/Introduccion%20a%20las%20redes%20neuronales.pdf>]
10. Documentación Weka. Universidad de Waikato, 17-12-2009 [<http://www.cs.waikato.ac.nz/ml/weka/>]168
11. FERRI C. "Mi página de Weka", 17-12-2009, [<http://www.dsic.upv.es/~cferrri/weka/>]
12. FERNÁNDEZ LORENZANA, Ramón. ASENJO CHACÓN, Manuel. "Inteligencia en redes de Comunicaciones: Práctica Final: Predicción Meteorológica en Weka", 17-12-2009, [<http://www.it.uc3m.es/jvillena/irc/practicass/04-05/10mem.pdf>]
13. GARCIA MORATE, Diego. "Manual de WEKA" ,15-12-2009, [<http://metaemotion.com/diego.garcia.morate/>]
14. SAMPER MÁRQUEZ, Juan José. "Sistemas Expertos. El conocimiento al poder", 15-12-2009, [<http://www.psycologia.com/articulos/arjsamper01.htm>].
15. [http://eisc.univalle.edu.co/cursos/web/material/750061M/1/guia\\_weka.pdf](http://eisc.univalle.edu.co/cursos/web/material/750061M/1/guia_weka.pdf)
16. <http://www.dataprix.com/732-preparacion-datos>
17. <http://www.metaemotion.com/diego.garcia.morate/download/weka.pdf>
18. [http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=368:que-es-java-concepto-de-programacion-orientada-a-objetos-vs-programacion-estructurada-cu00603b&catid=68:curso-aprender-programacion-java-desde-cero&Itemid=188](http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=368:que-es-java-concepto-de-programacion-orientada-a-objetos-vs-programacion-estructurada-cu00603b&catid=68:curso-aprender-programacion-java-desde-cero&Itemid=188)

## **ANEXOS**

### **Anexo 1. Manual del Módulo Sintomatología.**

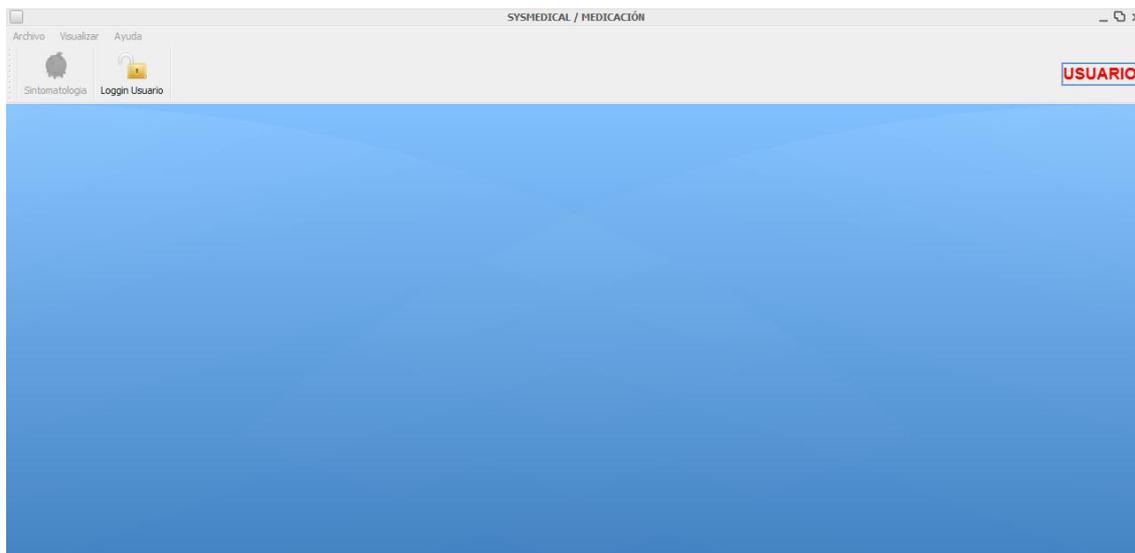
#### **Manual del Módulo Sintomatología.**

Nos permite colaborar con el diagnóstico de los medicamentos del paciente al que el programa brinde de manera oportuna la medicación para el o los pacientes, de manera que sirva de apoyo en las decisiones de los doctores. De manera rápida explicaremos resumidamente lo que se puede lograr con el siguiente módulo.

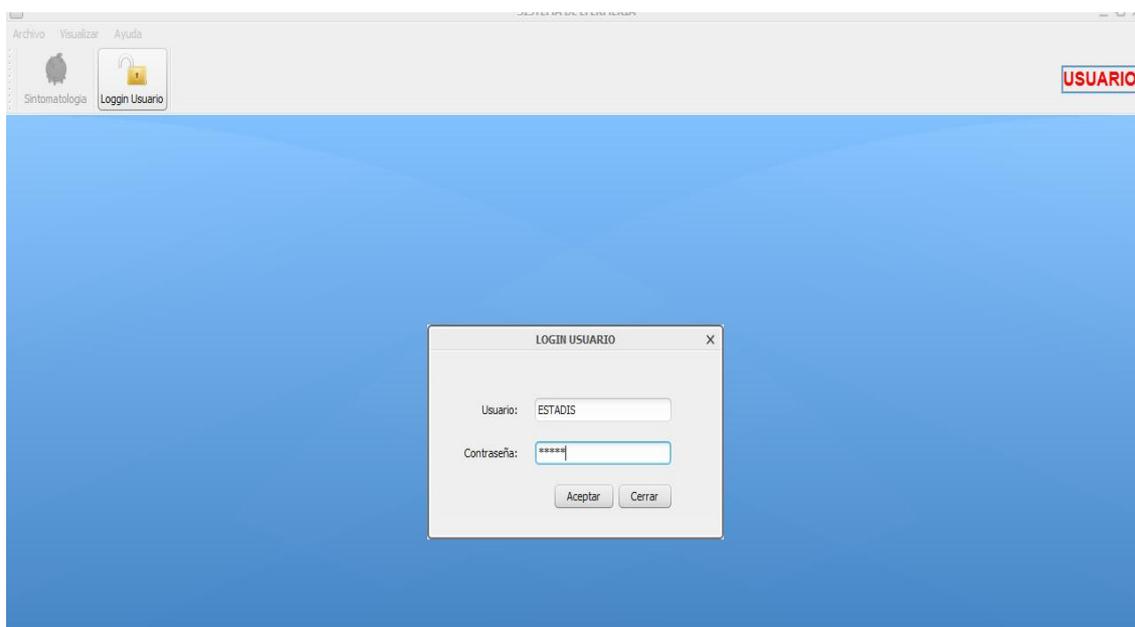
Crear, Modificar, Eliminar, visualizar Medicamentos, Síntomas, Enfermedades, y realizar a su vez la prescripción de la receta médica.

#### **1. Inicio**

Una vez iniciado el programa procedemos a registrarnos como usuario del mismo y hacemos clic en el botón [Login Usuario].

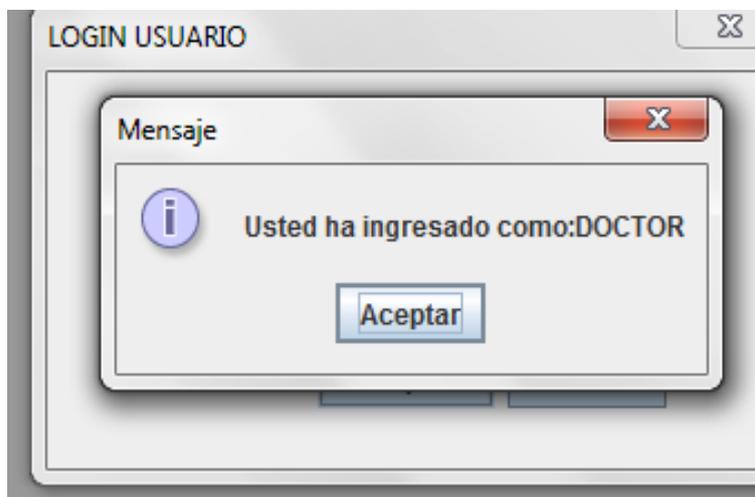
**FIGURA 1.** Pantalla General del módulo de Sintomatología.

Una vez que se haya realizado esta opción podemos proseguir y se nos muestra a continuación el siguiente diálogo para poder introducir los datos proporcionados por el administrador del Sistema. En el cual nos presenta el nombre de Usuario {Proporciona el administrador} y Contraseña {también dada por el administrador del Sistema}.

**Figura 2.** Pantalla de Registro del módulo de Sintomatología.

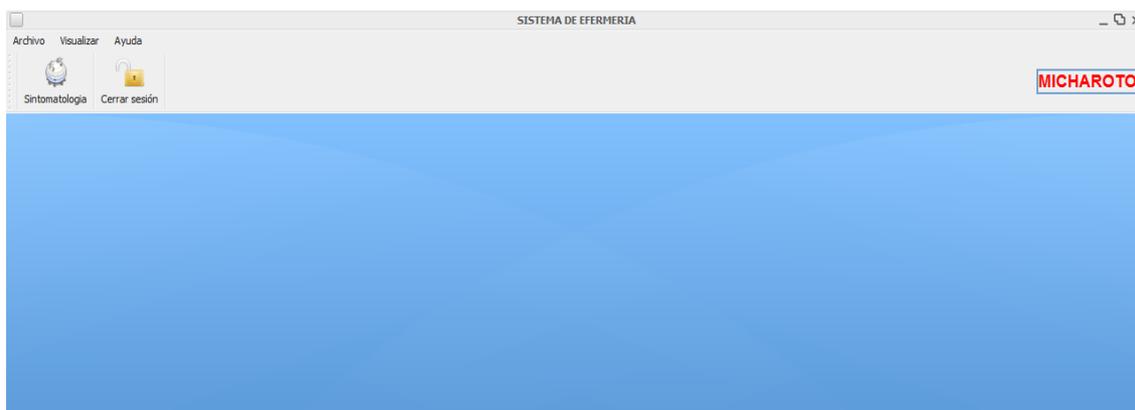
Una vez cumplido los pasos mencionados anteriormente y asumiendo que los datos son correcto lo que tenemos a continuación es el siguiente mensaje.

**Figura 3.** Mensaje del registro exitoso del módulo de sintomatología.



Listo nos muestra que hemos ingresado los datos correctamente por lo que a continuación tendremos una perspectiva general del programa.

**Figura 4.** Pantalla del Módulo de Sintomatología.



Esta es la perspectiva inicial donde tenemos acceso a todas las funcionalidades de nuestro Sistema.

Empecemos.

### a) Uso de Sintomatología.



Muy bien aplicaremos un ejemplo sobre el uso de la Parte de Sintomatología que ayuda con la prescripción de los medicamentos a un paciente de nuestro hospital iniciamos dando clic en el botón.

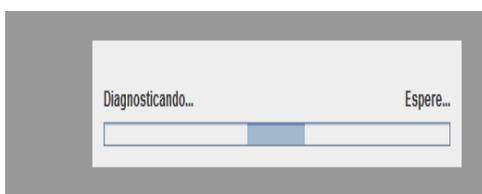
Luego del paso anterior procedemos a marcar los síntomas que el paciente presenta a la visita del doctor y se procede a ser señalados como lo presentamos en la ventana anterior, y una vez que acabemos de seleccionar todos los que se nos han presentado procedemos a dar clic en el botón de [Diagnosticar]. El cual procederá a cargar la posible enfermedad de la que padece.

**Figura 5.** Menú de Diagnóstico de enfermedad de Pacientes.

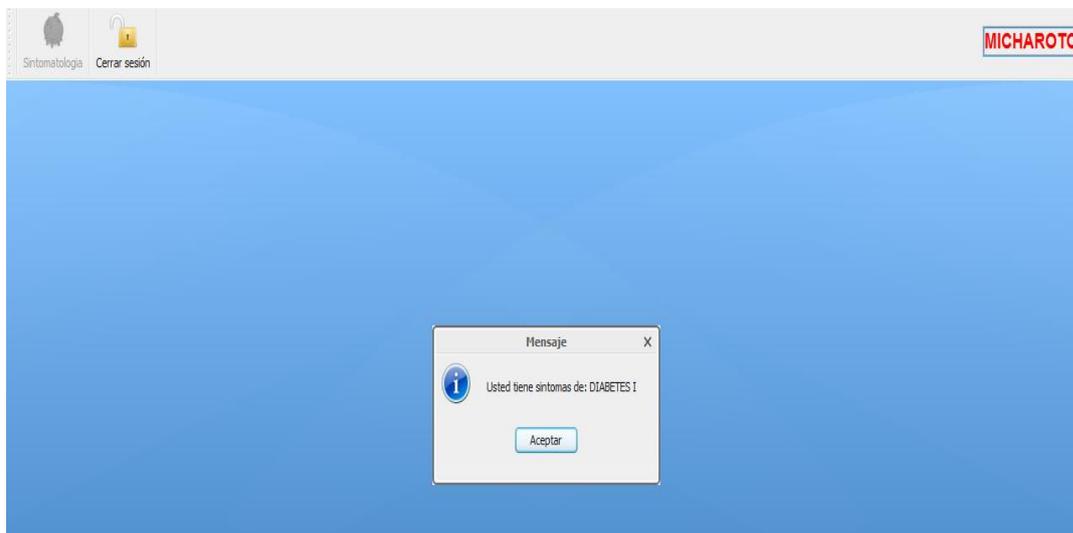


Listo esperamos a que cargue el paciente los datos.

**Figura 6.** Dialogo de carga del diagnóstico de enfermedad de Pacientes.

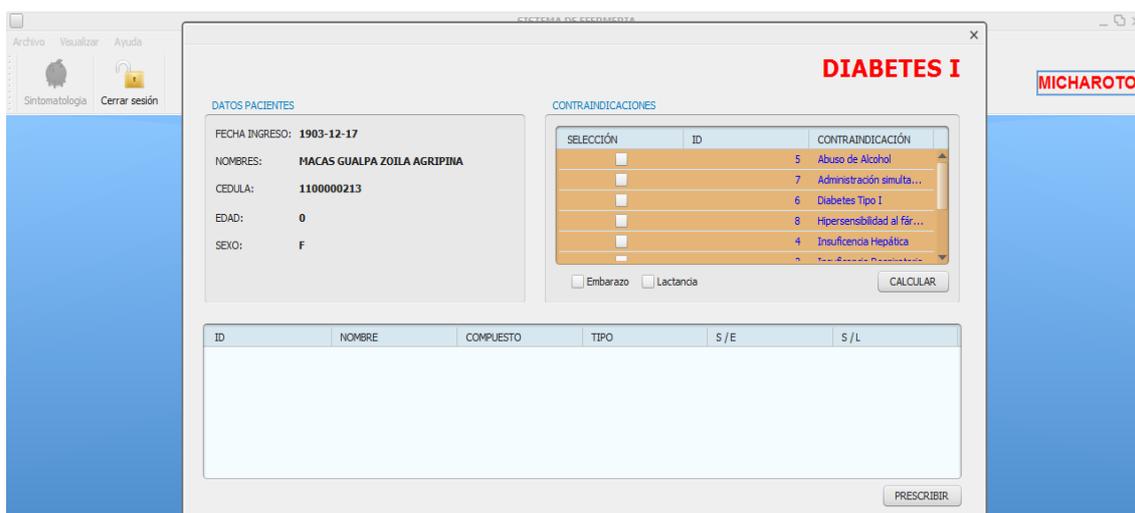


**Figura 7.** Mensaje informativo de enfermedad de Pacientes.



Damos clic sobre el botón [Aceptar] del mensaje que nos apareció con la enfermedad que posiblemente este padeciendo el paciente, ver figura nro. 7 y nos aparecerá la siguiente ventana, ver figura nro. 8:

**Figura 8.** Pantalla de las contraindicaciones al diagnóstico de enfermedad de Pacientes.



De aquí en adelante vamos a proceder a la sección donde vamos a seleccionar las Contraindicaciones de acuerdo al paciente y a su estudio por parte de la historia del paciente.

De manera que seleccionamos las diferentes Contraindicaciones que presenta el paciente.

**Figura 9.** Ejemplo de las contraindicaciones al diagnóstico de enfermedad de Pacientes.

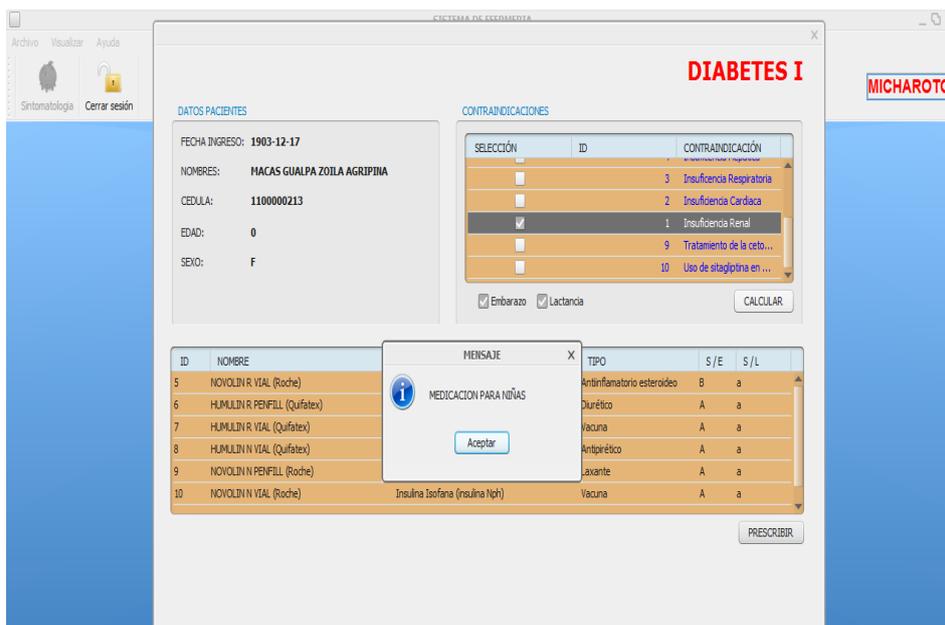
The screenshot shows a software window titled 'SISTEMA DE SEGUIMIENTO' with a patient profile for 'DIABETES I'. The patient's name is 'MACAS GUALPA ZOILA AGRIPPIA'. The 'CONTRAINDICACIONES' section contains a table with the following data:

SELECCIÓN	ID	CONTRAINDICACIÓN
<input type="checkbox"/>	3	Insuficiencia Respiratoria
<input type="checkbox"/>	2	Insuficiencia Cardíaca
<input checked="" type="checkbox"/>	1	Insuficiencia Renal
<input type="checkbox"/>	9	Tratamiento de la ceto...
<input type="checkbox"/>	10	Uso de sitagliptina en ...

Below the table are checkboxes for 'Embarazo' and 'Lactancia', and a 'CALCULAR' button. At the bottom of the window is a 'PRESCRIBIR' button.

Vamos a poner clic en el botón [**CALCULAR**], ver figura nro. 10 y arrojará el informativo de los datos de los posibles medicamento en base a la edad de todos estos datos.

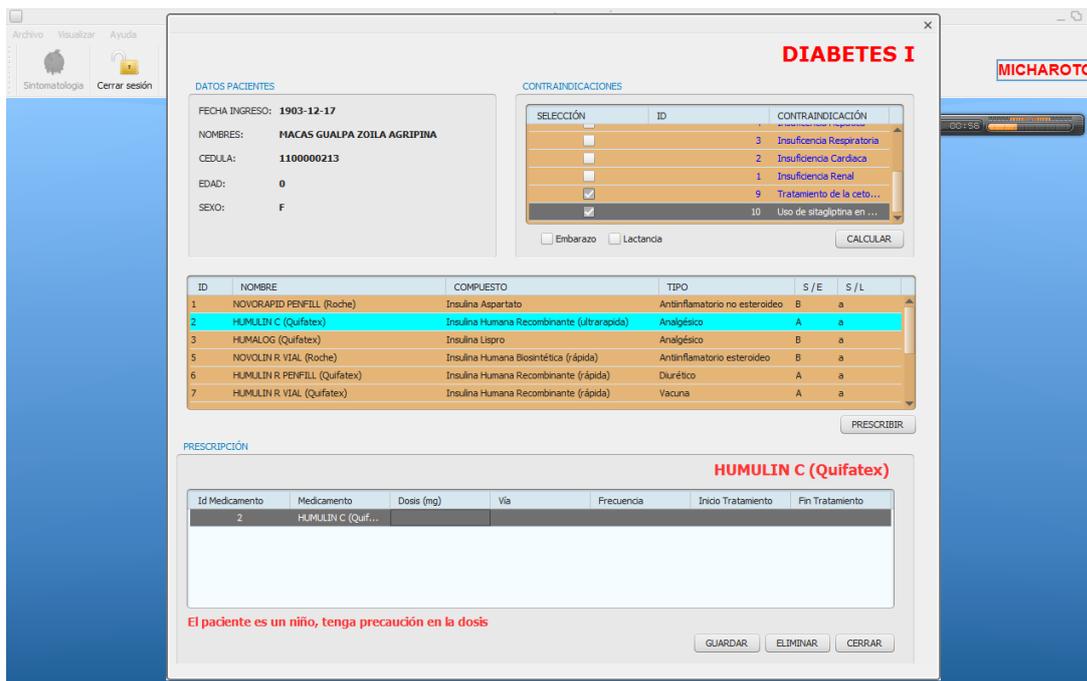
**Figura 10.** Mensaje informativo acerca de las contraindicaciones al diagnóstico de enfermedad de Pacientes.



Una vez lista esta edición de todos los medicamentos le damos clic en el botón [Aceptar] Este mensaje es solamente informativo por lo que el doctor podrá ver desplegada una gran cantidad de medicamentos que se han de desplegar para poder recetarle de acuerdo a la receta por lo que para que pueda como doctor precibir el medicamento en la receta del paciente.

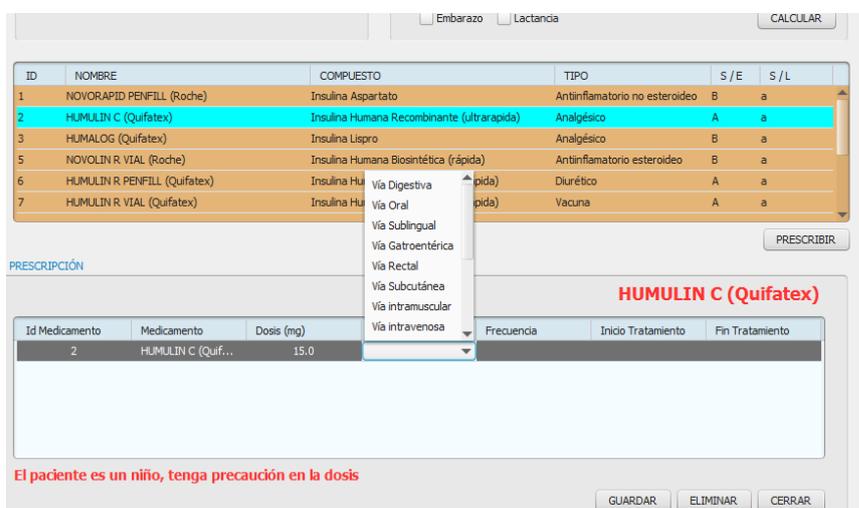
Debemos señalar el medicamento a convenir y luego le damos clic en el botón [PRESCRIBIR] como el ejemplo presentado a continuación.

Figura 11. Pantalla de la prescripción médica.



Al hacer el paso anterior se nos cargará la prescripción del medicamento final para suministrarle al paciente y el doctor pondrá la dosificación especial dependiendo de las características y de acuerdo al aviso que se le presenta con rojo en la parte inferior.

Figura 12. Proceso de la prescripción médica.



Ahora procedemos a ingresar la vía la forma de suministro al paciente por parte de la enfermera netamente puesto por el doctor, de aquí que se pondrá la frecuencia del mismo.

**Figura 13.** Proceso de la prescripción médica en turnos de dosis.

The screenshot displays a medical prescription interface. At the top, there is a list of medications with a dropdown menu open over the frequency field. The main area shows a table for 'HUMULIN C (Quifatex)' with columns for medication ID, name, dose, route, start, and end of treatment. A warning message at the bottom states 'El paciente es un niño, tenga precaución en la dosis'.

Id Medicamento	Medicamento	Dosis (mg)	Vía	Inicio Tratamiento	Fin Tratamiento
2	HUMULIN C (Quif...	15.0	Vía Oral		

El paciente es un niño, tenga precaución en la dosis

La frecuencia varía de acuerdo a las horas que rigen en el reloj que lleva el departamento de enfermería que es bastante conocido por el personal de doctores residentes y es la que va a regir para que se generen los turnos del suministro de medicamentos de los mismos.

Figura 14. Proceso de la prescripción médica en fecha de atención.

**DIABETES I**

**DATOS PACIENTES**

FECHA INGRESO: 1903-12-17  
 NOMBRES: MACAS GUALPA ZOILA AGRIPINA  
 CEDULA: 1100000213  
 EDAD: 0  
 SEXO: F

**CONTRAINDICACIONES**

SELECCIÓN	ID	CONTRAINDICACIÓN
<input type="checkbox"/>	3	Insuficiencia Respiratoria
<input type="checkbox"/>	2	Insuficiencia Cardíaca
<input type="checkbox"/>	1	Insuficiencia Renal
<input checked="" type="checkbox"/>	9	Tratamiento de la ceto...
<input checked="" type="checkbox"/>	10	Uso de sitagliptina en ...

Embarazo  Lactancia CALCULAR

ID	NOMBRE	COMPUESTO	TIPO	S / E	S / L
1	NOVORAPID PENFILL (Roche)	Insulina Aspartato	Antiinflamatorio no esteroideo	B	a
2	HUMULIN C (Quifatex)	Insulina Humana Recombinante (ultrarápida)	Analgésico	A	a
3	HUMALOG (Quifatex)	Insulina Lispro	Analgésico	B	a
5	NOVOLIN R VIAL (Roche)	Insulina Humana Biosintética (rápida)	Antiinflamatorio esteroideo	B	a
6	HUMULIN R PENFILL (Quifatex)	Insulina Humana Recombinante (rápida)	Diurético	A	a
7	HUMULIN R VIAL (Quifatex)	Insulina Humana Recombinante (rápida)	Vacuna	A	a

PRESCRIBIR

**PRESCRIPCIÓN**

**HUMULIN C (Quifatex)**

Id Medicamento	Medicamento	Dosis (mg)	Vía	Frecuencia	Inicio Tratamiento	Fin Tratamiento
2	HUMULIN C (Quif...	15.0	Vía Oral	3	03/11/2011	

El paciente es un niño, tenga precaución en la dosis

MINIMAR CERRAR

Luego procedemos a poner la fecha de inicio del tratamiento y la fecha del fin del tratamiento.

Figura 15. Ejemplo del proceso de la prescripción médica.

**DIABETES I**

**DATOS PACIENTES**

FECHA INGRESO: 1903-12-17  
 NOMBRES: MACAS GUALPA ZOILA AGRIPINA  
 CEDULA: 1100000213  
 EDAD: 0  
 SEXO: F

**CONTRAINDICACIONES**

SELECCIÓN	ID	CONTRAINDICACIÓN
<input type="checkbox"/>	3	Insuficiencia Respiratoria
<input type="checkbox"/>	2	Insuficiencia Cardíaca
<input type="checkbox"/>	1	Insuficiencia Renal
<input checked="" type="checkbox"/>	9	Tratamiento de la ceto...
<input checked="" type="checkbox"/>	10	Uso de sitagliptina en ...

Embarazo  Lactancia CALCULAR

ID	NOMBRE	COMPUESTO	TIPO	S / E	S / L
1	NOVORAPID PENFILL (Roche)	Insulina Aspartato	Antiinflamatorio no esteroideo	B	a
2	HUMULIN C (Quifatex)	Insulina Humana Recombinante (ultrarápida)	Analgésico	A	a
3	HUMALOG (Quifatex)	Insulina Lispro	Analgésico	B	a
5	NOVOLIN R VIAL (Roche)	Insulina Humana Biosintética (rápida)	Antiinflamatorio esteroideo	B	a
6	HUMULIN R PENFILL (Quifatex)	Insulina Humana Recombinante (rápida)	Diurético	A	a
7	HUMULIN R VIAL (Quifatex)	Insulina Humana Recombinante (rápida)	Vacuna	A	a

PRESCRIBIR

**PRESCRIPCIÓN**

**HUMULIN C (Quifatex)**

Id Medicamento	Medicamento	Dosis (mg)	Vía	Frecuencia	Inicio Tratamiento	Fin Tratamiento
2	HUMULIN C (Quif...	15.0	Vía Oral	3	16/11/2011	17/03/2012

El paciente es un niño, tenga precaución en la dosis

GUARDAR ELIMINAR CERRAR

Listo hasta que al final le damos [Enter] y listo seguimos y finalizamos el tratamiento y de la misma manera proseguimos con los medicamentos que crea como Dr. Conveniente.

**Figura 16.** Ejemplo del proceso de la prescripción médica final.

DIABETES I

**DATOS PACIENTES**

FECHA INGRESO: 1903-12-17

NOMBRES: MACAS GUALPA ZOILA AGRIPINA

CEDULA: 1100000213

EDAD: 0

SEXO: F

**CONTRAINDICACIONES**

SELECCIÓN	ID	CONTRAINDICACIÓN
<input checked="" type="checkbox"/>	3	Insuficiencia Respiratoria
<input type="checkbox"/>	2	Insuficiencia Cardíaca
<input type="checkbox"/>	1	Insuficiencia Renal
<input type="checkbox"/>	9	Tratamiento de la ceto...
<input type="checkbox"/>	10	Uso de sitagliptina en ...

Embarazo  Lactancia

CALCULAR

ID	NOMBRE	COMUESTO	TIPO	S/E	S/L
8	HUMULIN N VIAL (Quifatex)	Insulina Isofana (insulina Nph)	Antipirético	A	a
9	NOVOLIN N PENFILL (Roche)	Insulina Isofana (insulina Nph)	Laxante	A	a
10	NOVOLIN N VIAL (Roche)	Insulina Isofana (insulina Nph)	Vacuna	A	a
11	LANTUS (Aventis)	Insulina Glargina	Antiséptico	C	d
12	APIDRA (Aventis)	Insulina Gluslina	Diurético	C	d
46	ASPIRINA	ASPIRINA	Analgésico	A	a

PRESCRIBIR

**PRESCRIPCIÓN**

NOVOLIN N PENFILL (Roche)

Id Medicamento	Medicamento	Dosis (mg)	Vía	Frecuencia	Inicio Tratamiento	Fin Tratamiento
2	HUMULIN C (Quif...	10.0	Vía Sublingual	3	05/01/2012	03/03/2012
9	NOVOLIN N PENF...	5.0	Vía Oral	4	14/01/2012	10/03/2012

El paciente es un niño, tenga precaución en la dosis

Luego para guardar le damos clic en el botón [GUARDAR], si queremos eliminar nuestra receta damos clic en [ELIMINAR] y para poder cerrar en el botón [CERRAR] en el caso de nuestro ejemplo hemos podido Guardar la prescripción al paciente.

Y listo hemos finalizado la prescripción médica.

## **ANEXO 2. Elaboración de archivos arff de las enfermedades del cerco epidemiológico**

### **1. HIPERTENSION ARTERIAL (HTA)**

La hipertensión arterial (HTA) es elevación de los niveles de presión arterial de forma continua o sostenida. Para entenderlo mejor es importante definir la presión arterial. El corazón ejerce presión sobre las arterias para que éstas conduzcan la sangre hacia los diferentes órganos del cuerpo humano [16].

#### **1.1. SÍNTOMAS Y SIGNOS DE LA HTA**

La mayor parte del tiempo, no hay síntomas. Los síntomas que pueden ocurrir abarcan:

- Dolor torácico
- Confusión
- Zumbido o ruido en el oído
- Fatiga
- Dolor de cabeza
- Latidos cardíacos irregulares
- Hemorragia nasal
- Cambios en la visión

Si presenta dolor de cabeza fuerte o cualquiera de los síntomas mencionados anteriormente, consulte con el médico de inmediato [16].

### 1.1.1. ELABORACIÓN DEL ARCHIVO ARFF

**TABLA 1. PEUDOCÓDIGO DEL ARCHIVO .ARFF DESARROLLADO PARA DIAGNÓSTICO DE LA HTA**

<pre> @relation hipertensión  @attribute perdida_sensibilidad_extremidades NUMERIC @attribute hemorragias_nasales NUMERIC @attribute dolor_cabeza NUMERIC @attribute mareos NUMERIC @attribute tos_sanguinolenta NUMERIC @attribute perdida_vision NUMERIC @attribute hemorragia_conjuntiva NUMERIC @attribute caloradas NUMERIC @attribute palpitaciones_corazon NUMERIC @attribute taquicardias NUMERIC @attribute hipertension {0, 1}  @data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0 1,0,0,0,0,0,0,0,0,0,0,0,1,0,1----- hasta completar y se comienza la fase de entrenamiento. </pre>
Elaborado por:

## 2. DIABETES MELLITUS TIPO 1

La diabetes mellitus tipo I o también conocida como diabetes juvenil o diabetes mellitus dependiente, es una enfermedad metabólica caracterizada por una destrucción selectiva de las Célula beta del páncreas causando una deficiencia

absoluta de insulina. Se diferencia de la diabetes mellitus tipo 2 porque es un tipo de diabetes caracterizada por darse en época temprana de la vida, generalmente antes de los 30 años. Sólo 1 de cada 20 personas diabéticas tiene diabetes tipo I, la cual se presenta más frecuentemente en jóvenes y niños. La administración de insulina en estos pacientes es esencial. La diabetes tipo 1 se clasifica en casos autoinmunes—la forma más común—y en casos idiopáticos. La diabetes tipo 1 se encuentra entre todos los grupos étnicos, pero su mayor incidencia se encuentra entre poblaciones del norte de Europa y en Cerdeña. La susceptibilidad a contraer diabetes mellitus tipo 1 parece estar asociada a factores genéticos múltiples, aunque solo el 15-20% de los pacientes tienen una historia familiar positiva [24].

### **2.1. SÍNTOMAS Y SIGNOS DE LA DIABETES 1**

Algunas personas no tendrán ningún síntoma antes de que se les diagnostique la diabetes. Otras pueden notar estos síntomas como los primeros signos de diabetes tipo 1 o cuando la glucemia está alta:

- Sentirse cansado o fatigado
- Sentirse hambriento
- Estar muy sediento
- Orinar con mayor frecuencia
- Perder peso sin proponérselo
- Tener visión borrosa
- Perder la sensibilidad o sentir hormigueo en los pies

Para otras, los síntomas de advertencia de que se están poniendo muy enfermos pueden ser los primeros signos de diabetes tipo 1, o pueden suceder cuando la glucemia está muy alta (ver cetoacidosis diabética):

- Respiración profunda y rápida
- Boca y piel seca
- Cara enrojecida
- Aliento con olor a fruta
- Náuseas o vómitos, incapacidad para retener los líquidos
- Dolor de estómago

La glucemia baja (hipoglucemia) [24] se puede desarrollar rápidamente en personas con diabetes que estén tomando insulina. Los síntomas aparecen típicamente cuando el nivel de glucemia cae por debajo de 70 mg/dL. Tenga cuidado con:

- Dolor de cabeza
- Hambre
- Nerviosismo
- Latidos cardíacos rápidos (palpitaciones)
- Temblores
- Sudoración
- Debilidad.

## 2.2. ELABORACIÓN DEL ARCHIVO ARFF

**TABLA 2. PEUDOCÓDIGO DEL ARCHIVO .ARFF DESARROLLADO PARA DIAGNÓSTICO DE LA DIABETES I**

<p>@relation diabetesI</p> <p>@attribute orina_frecuente NUMERIC</p> <p>@attribute sed_excesiva NUMERIC</p>
-------------------------------------------------------------------------------------------------------------

@attribute hambre\_excesiva NUMERIC

@attribute perdida\_peso NUMERIC

@attribute debilidad NUMERIC

@attribute somnolencia NUMERIC

@attribute vision\_borrosa NUMERIC

@attribute nausea NUMERIC

@attribute vomito NUMERIC

@attribute ojito NUMERIC

@attribute diabetesl {0, 1}

@data

0,0,0,0,0,0,0,0,0,0,0

1,1,0,0,0,0,0,0,0,0,1

1,0,1,0,0,0,0,0,0,0,1

1,0,0,1,0,0,0,0,0,0,1

1,0,0,0,0,0,0,0,0,0,1

1,0,0,0,1,0,0,0,0,0,1

1,0,0,0,0,1,0,0,0,0,1

1,0,0,0,0,0,1,0,0,0,1

1,1,0,0,1,0,0,0,0,0,1

1,1,0,0,0,1,0,0,0,0,1

1,1,0,0,0,0,1,0,0,0,1

1,1,0,0,0,0,0,1,0,0,1

1,1,0,0,0,0,0,0,1,0,1

1,1,1,1,0,0,0,0,0,0,1

1,1,1,0,1,0,0,0,0,0,1----- hasta completar y se comienza la fase de

entrenamiento.
Elaborado por:

### **3. DIABETES MELLITUS TIPO 2**

Conocida anteriormente como diabetes no-insulinodependiente— es una enfermedad metabólica que se caracteriza por altos niveles de glucosa en la sangre, debido a una resistencia celular a las acciones de la insulina, combinada con una deficiente secreción de insulina por el páncreas [26].

#### **3.1. SÍNTOMAS Y SIGNOS DE LA DIABETES 2**

Con frecuencia, las personas con diabetes tipo 2 no presentan síntoma alguno.

En caso de presentarse síntomas, éstos pueden ser:

- Visión borrosa
- Disfunción eréctil
- Fatiga
- Infecciones frecuentes o de curación lenta
- Aumento del apetito
- Aumento de la sed
- Aumento de la micción.

### 3.1.1.1. ELABORACIÓN DEL ARCHIVO ARFF

**TABLA 3. PEUDOCÓDIGO DEL ARCHIVO .ARFF DESARROLLADO PARA DIAGNÓSTICO DE LA DIABETES II**

<pre> @relation diabetesII  @attribute orina_frecuente NUMERIC @attribute sed_excesiva NUMERIC @attribute hambre_excesiva NUMERIC @attribute vision_borrosa NUMERIC @attribute nauseas NUMERIC @attribute vomito NUMERIC @attribute sensacion_cansancio NUMERIC @attribute infecciones_frecuentes NUMERIC @attribute hormigueo_manos_pies NUMERIC @attribute entumecimiento_manos_pies NUMERIC @attribute picazon_piel_genitales NUMERIC @attribute demora_cicatrizar NUMERIC @attribute piel_seca NUMERIC @attribute diabetesII {0, 1}  @data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 1,0,1,0,0,0,0,0,0,0,0,0,0,0,0 1,0,0,0,0,0,0,0,0,0,0,1,0,0----- hasta completar y se comienza la fase de entrenamiento. </pre>
Elaborado por:

## **4. HEPATITIS B**

La hepatitis B es una enfermedad del hígado causada por el virus de la hepatitis B, perteneciente a la familia Hepadnaviridae (virus ADN hepatotrópico). Es una enfermedad infecciosa del hígado causada por este virus y caracterizada por necrosis hepatocelular e inflamación [27].

### **4.1. SÍNTOMAS Y SIGNOS DE LA HEPATITIS B.**

Después de que usted resulte infectado por primera vez con el virus de la hepatitis B:

- Puede que no tenga ningún síntoma
- Puede sentirse enfermo durante un periodo de días o semanas
- Puede resultar muy enfermo (llamada hepatitis fulminante)

Los síntomas pueden no aparecer hasta 6 meses después del momento de la infección. Los síntomas iniciales pueden abarcar:

- Inapetencia
- Fatiga
- Febrícula
- Dolores musculares y articulares
- Náuseas y vómitos
- Piel amarilla y orina turbia debido a la ictericia



## 5. FIEBRE REUMÁTICA

La fiebre reumática es una enfermedad inflamatoria, no supurativa y recurrente producida por la respuesta del sistema inmunitario de algunas personas predispuestas, a los antígenos de la bacteria estreptococo del grupo A beta hemolítico [10].

### 5.1. SÍNTOMAS Y SIGNOS DE LA FIEBRE REUMÁTICA.

Entre los síntomas más comunes tenemos los siguientes:

- Dolor abdominal.
- Fiebre.
- Problemas del corazón (cardíacos) que pueden ser asintomáticos o pueden ocasionar dificultad respiratoria y dolor torácico.
- Dolor articular, artritis (principalmente en las rodillas, los codos, los tobillos y las muñecas).
- Inflamación articular, enrojecimiento o calor.
- Hemorragias nasales (epistaxis).
- Erupción en la piel (eritema marginado).
- Erupción cutánea en el tronco y en la parte superior de brazos o piernas.
- Erupciones que pueden tener apariencia de anillo o parecido a una serpiente

### 5.2. ELABORACIÓN DEL ARCHIVO ARFF PARA LA FIEBRE REUMÁTICA

**TABLA 5. PEUDOCÓDIGO DEL ARCHIVO .ARFF DESARROLLADO PARA DIAGNÓSTICO DE LA FIEBRE REUMÁTICA**

@relation fiebreReumatica
@attribute sensacion_cansancio NUMERIC
@attribute hinchazon_generalizado_retencion_liquididos NUMERIC
@attribute inapetencia NUMERIC

<p>@attribute inflamacion_articular NUMERIC</p> <p>@attribute fiebre NUMERIC</p> <p>@attribute palidez NUMERIC</p> <p>@attribute malestar_general NUMERIC</p> <p>@attribute dolor_abdominal NUMERIC</p> <p>@attribute dolor_pecho NUMERIC</p> <p>@attribute erupciones_cutaneas NUMERIC</p> <p>@attribute bultos_codos_rodillas NUMERIC</p> <p>@attribute respiracion_jadeante NUMERIC</p> <p>@attribute pulso_rapido NUMERIC</p> <p>@attribute movimientos_incontrolables_brazos_piernas NUMERIC</p> <p>@attribute fiebreReumatica {0, 1}</p> <p>@data</p> <p>0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</p> <p>1,0,0,0,0,0,0,0,0,0,1,0,0,0,1----- hasta completar y se comienza la fase de entrenamiento.</p>
Elaborado por:

### 5.2.1. EDA (ENFERMEDAD DIARREICA AGUDA)

La enfermedad diarreica aguda (EDA) se puede definir como un cambio súbito en el patrón de evacuación intestinal normal del individuo, caracterizado por aumento en la frecuencia o disminución en la consistencia de las deposiciones. Para ser considerada como aguda, su aparición debe tener menos de tres semanas. La causa más importante y frecuente de EDA es la infección entero-cólica con

respuesta variable en los enfermos; algunos manifiestan cuadros graves, otros síntomas moderados y otros son asintomáticos [19].

#### 5.2.1.1. SÍNTOMAS Y SIGNOS DE LA EDA

Podemos poner a consideración la siguiente tabla que presentamos a continuación:

**TABLA 6. POSIBLES SÍNTOMAS Y LA EVALUACIÓN HACIA LOS DIFERENTES CUADROS DE TIPOS DE DIARREA GENERADA POR EL EDA [19]**

<b>Evaluación del estado de hidratación de un paciente con diarrea</b>			
<b>Signos</b>	<b>Plan A</b>	<b>Plan B</b>	<b>Plan C</b>
<b>Definición</b>	Leve o inaparente	Moderada o clínica	Grave
<b>Pérdida de agua corporal</b>	Menos de 50 ml/kg peso o menos de 5% del peso	50 a 100 ml/kg peso ó 6 a 9% del peso	100 ml/kg peso o más (10% o más del peso)
<b>Condición general</b>	Bien, alerta	Irritable	Letárgico o inconsciente
<b>Globo ocular</b>	Normales	Algo hundidos	Muy hundidos y secos
<b>Lágrimas</b>	Presentes	Ausentes	Ausentes
<b>Mucosa oral</b>	Húmeda	Seca	Muy seca
<b>Sed</b>	Paciente bebe normalmente	Paciente bebe con avidez, sediento	Paciente bebe mal o no es capaz de hacerlo

Pliegue cutáneo	Vuelve a lo normal rápidamente	Se retrae lentamente < 2 s	Se retrae muy lentamente > 2 s
Tratamiento	Plan A	Plan B	Plan C
			
Rehidratación oral	Mucosa húmeda, lágrimas	Ojos hundidos, mucosa seca	Terapia intravenosa

#### 5.2.1.2. ELABORACIÓN DEL ARCHIVO ARFF PARA LA EDA

**TABLA 7. PEUDOCÓDIGO DEL ARCHIVO .ARFF DESARROLLADO PARA E.D.A**

<pre> @relation EDA  @attribute nausea NUMERIC  @attribute vomito NUMERIC  @attribute fiebre NUMERIC  @attribute mialgias NUMERIC  @attribute tenesmo_rectal NUMERIC  @attribute ardor_perianal NUMERIC  @attribute diarrea NUMERIC  @attribute EDA {0, 1}  @data  0,0,0,0,0,0,0,0 1,0,0,0,0,0,0,0 1,1,0,0,0,0,0,1 1,0,1,0,0,0,0,1 1,0,0,1,0,0,0,1 1,0,0,0,1,0,0,1----- hasta completar y se comienza la fase de entrenamiento. </pre>
Elaborado por:

## 6. IRA (INFECCIÓN RESPIRATORIA AGUDA)

Las infecciones respiratorias agudas (IRA) son padecimientos infecciosos de las vías respiratorias con evolución menor a 15 días y en ocasiones se complican con neumonía. Las IRA constituyen un importante problema de salud pública, pues resultan con la morbilidad más alta en el mundo [22].

## 6.1. SÍNTOMAS Y SIGNOS DE LA IRA

Para reconocer las IRAS usted debe identificar en el enfermo los siguientes síntomas:

- Tos
- Nariz tapada (trancada) y con mocos.
- Dificultad para respirar, lo que obliga a la persona a respirar de manera rápida y agitada. En algunos momentos, el enfermo trata de respirar por la boca.
- Dificultad para tragar.
- Dolor de garganta.
- Dolor de oído.
- Fiebre.
- Ronquido, quejido o silbido en el pecho cuando inhala (toma aire).

### 6.1.1. ELABORACIÓN DEL ARCHIVO ARFF PARA LA IRA

**TABLA 8. PEUDOCÓDIGO DEL ARCHIVO .ARFF DESARROLLADO PARA I.R.A**

@relation IRA
@attribute somnolencia NUMERIC
@attribute nausea NUMERIC
@attribute vomito NUMERIC
@attribute sensacion_cansancio NUMERIC
@attribute disminucion_cantidad_orina NUMERIC
@attribute ausencia_gasto_urinario NUMERIC
@attribute miccion_excesiva NUMERIC
@attribute hinchazon_tobillos_pies_piernas NUMERIC
@attribute hinchazon_generalizado_retencion_liquidos NUMERIC



1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1 ----- hasta completar y se comienza
Elaborado por:

### 6.1.2. HEPATITIS VÍRICA

Enfermedad del hígado por lesión viral o vírica (HV), es decir de origen infeccioso y no tóxico. Aunque existen diferentes tipos (A,B,C,...) que difieren en general según el origen de la infección (jeringuillas, comida/bebida o utensilios, ...), el cuadro clínico siempre comienza con síntomas de cansancio y dolores de músculos y articulaciones, y a veces se asocia la fiebre a estas molestias [24].

#### 6.1.2.1. TRATAMIENTO DE LA HEPATITIS VÍRICA

El diagnóstico de la hepatitis Vírica es fácil clínicamente.

Las hepatitis víricas suelen descubrirse mediante análisis de sangre. Para el estudio definitivo de las hepatitis crónicas puede ser necesario realizar

**Ecografías**, T.A.C. (escáner) o R.M.N. (resonancia), toma de tejido hepático mediante una aguja (punción-biopsia) para su estudio microscópico.

#### Medidas Generales

Evitar el paracetamol en la fase aguda (puede ser tóxico para el hígado).

Baños de agua fresca, o algún antihistamínico suave para el picor.

### 6.1.2.2. ELABORACIÓN DEL ARCHIVO ARFF PARA LA HEPATITIS

**TABLA 9. PEUDOCÓDIGO DEL ARCHIVO .ARFF DESARROLLADO PARA HEPATITIS**

```

relation hepatitisVirica

@attribute sensacion_cansancio NUMERIC
@attribute inapetencia NUMERIC
@attribute ictericia NUMERIC
@attribute orina_oscura NUMERIC
@attribute heces_decoloradas NUMERIC
@attribute falta_animo NUMERIC
@attribute cambios_humor NUMERIC
@attribute esteatorrea NUMERIC
@attribute picor_generalizado NUMERIC
@attribute intolerancia_alimentos_grasos NUMERIC
@attribute malestias_abdominales_vagas NUMERIC
@attribute hinchazon_costillas_derecha NUMERIC
@attribute hepatitisVirica {0, 1}

@data

0,0,0,0,0,0,0,0,0,0,0,0,0
1,0,0,0,0,0,0,0,0,0,0,0,0
1,1,0,0,0,0,0,0,0,0,0,0,1
1,0,1,0,0,0,0,0,0,0,0,0,1
1,0,0,1,0,0,0,0,0,0,0,0,1
1,0,0,0,1,0,0,0,0,0,0,0,1
1,0,0,0,0,1,0,0,0,0,0,0,1

```

1,0,0,0,0,0,1,0,0,0,0,0,1

1,0,0,0,0,0,0,1,0,0,0,0,1

1,0,0,0,0,0,0,0,1,0,0,0,1

1,0,0,0,0,0,0,0,0,1,0,0,1----- hasta completar y se comienza la fase de entrenamiento.

Elaborado por:

### **ANEXO 3. Preguntas para dar inicio al proyecto análisis de las diez enfermedades más comunes en la sangre**

#### **DESARROLLO DE UN SISTEMA MÉDICO PARA EL ANALISIS DE LAS DIEZ ENFERMEDADES MÁS COMUNES EN LA SANGRE**

**1. ¿Cuál es el problema? (Enuncie la problemática de manera clara, asumiendo las situaciones negativas que se dan en el entorno afectado, y no las carencias de solución).**

La falta de un sistema experto para diagnóstico de las 10 enfermedades más comunes en pacientes, administración de historias clínicas y de medicamentos en un entorno web empresarial genera errores en el almacenamiento y correcto acceso a la información, un mal control de los horarios de visita para medicar a los pacientes, así como impide la detección de las diez enfermedades más comunes en pacientes etapas tempranas.

**2. ¿Por qué es importante investigar sobre el tema? (Enuncie la importancia o relevancia que tiene el dar solución al problema planteado, esto puede concluirse en satisfacer una necesidad, dar un aporte a la sociedad, o la generación y/o creación de un nuevo conocimiento).**

Los sistemas distribuidos hoy por hoy y la web 2.0 evolucionan marcando el futuro de la informática, el área de la salud de nuestro país y el gran volumen información ahí generada, requieren la utilización de arquitecturas empresariales que proveen soluciones

sofisticadas, elegantes implementado procesos que se ejecutan de manera más eficiente, información almacenada y accesible de forma oportuna.

Los sistemas expertos son un área de mucho interés en la actualidad, ya que es amplia las aplicaciones de los mismos en los diversos campos de nuestra sociedad, en la medicina pueden adelantarse en el análisis de datos, y llegar a una conclusión en base a los mismos ayudando de una gran manera a los profesionales de la medicina.

**3. ¿Qué se conoce al respecto hasta el momento, dentro y fuera del país? (Hacer referencias a hechos comprobables que demuestren lo que se ha hecho o dejado de hacer con respecto a la problemática planteada, hacer uso de referencias bibliográficas y webgráficas)**

Actualmente conocemos que los diferentes organismos de nuestro país, migran hacia el desarrollo de plataformas empresariales interconectadas, que permitan la disponibilidad de la información requerida de una forma oportuna, tenemos como ejemplo el Sri, la asamblea nacional, sistema de conteo de votos etc., los sistemas expertos dan un valor adicional a esto permitiendo al sistema evaluar ciertos indicadores y llegar a conclusiones de notificación al administrador.

**4. ¿Por qué lo va hacer? (Defina las razones que le motivan a desarrollar esta tesis, ya sea por necesidades insatisfechas, requerimientos o demanda del mercado o si es el caso una oferta para el mercado).**

Lo hare porque esto me ayudará a la formación personal en el desarrollo de plataformas empresariales y sistemas expertos además a la inexistencia actual de un software con las características expuestas, sumando a esto también la ayuda al personal de la médica.

**5. ¿Cómo lo va a realizar? (Defina las actividades macro y/o proceso por el cual se espera resolver el problema de la investigación definido en la pregunta 1).**

- Investigación de campo.
- Recolección de datos
- Análisis de datos y resultados.(determinar la diez enfermedades a diagnosticar)
- Determinación de requerimientos funcionales y no funcionales
- Diseño de la arquitectura de software adecuada para el proyecto
- Aplicación de una metodología de desarrollo de software
- Planificación de depuración de errores
- Implementación.

**6. ¿Cuáles son los resultados esperados? (Enuncie los productos y/o entregables que se espera obtener en el transcurso del desarrollo de la tesis, enunciarlos en términos de las 2C cantidad, y calidad)**

Una plataforma empresarial Para la administración de pacientes, enfermería y un experto para diagnóstico de enfermedades comunes:

Componente de software para administración de pacientes

Componente de software para administración de enfermería.

Componente de software experto para diagnóstico de enfermedades.

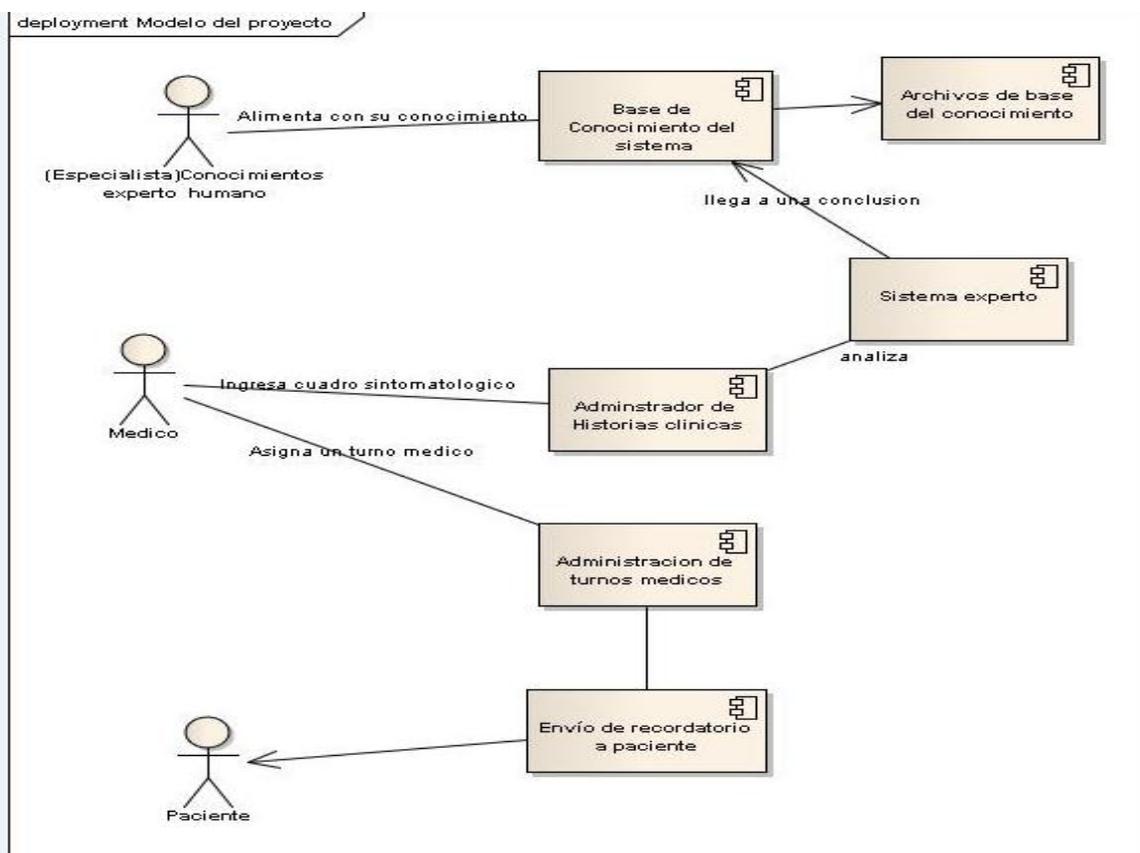
7. ¿Cómo va a transferir y difundir los resultados? (Explique cómo espera transferir y/o difundir los resultados de la tesis, estos pueden ser elementos de promoción, difusión, y publicación, así como planes de capacitación que estarán incluidos en la tesis).  
 Montar un sitio web informativo donde se pueda acceder de forma fácil a toda la información del proyecto.

8. ¿Qué efectos e impactos podría tener las nuevas tecnologías o los nuevos conocimientos en el grupo objetivo? (En los aspectos tecnológico, económicos, sociales, ambientales, institucionales, Sin asumir a todos estos como obligatorios).

Incentivar al desarrollo de este tipo de soluciones en nuestro medio.

Tomando como base el presente trabajo se puede desarrollar nuevos sistemas.

Incentivar la utilización de software libre en los proyectos informáticos.



### ANEXO 3. Reglas de Inferencia

Las reglas de inferencia para el sistema fueron traducidas a una nomenclatura estándar para ingreso de datos de la plataforma weka, estos son los archivos arff como ejemplo tenemos para la determinación aproximada de la fiebre reumática tenemos los siguientes síntomas:

- @attribute sensacion\_cansancio NUMERIC
- @attribute hinchazon\_generalizado\_retencion\_liquidos NUMERIC
- @attribute inapetencia NUMERIC
- @attribute inflamacion\_articular NUMERIC
- @attribute fiebre NUMERIC
- @attribute palidez NUMERIC
- @attribute malestar\_general NUMERIC
- @attribute dolor\_abdominal NUMERIC
- @attribute dolor\_pecho NUMERIC
- @attribute erupciones\_cutaneas NUMERIC
- @attribute bultos\_codos\_rodillas NUMERIC
- @attribute respiracion\_jadeante NUMERIC
- @attribute pulso\_rapido NUMERIC
- @attribute movimientos\_incontrolables\_brazos\_piernas NUMERIC

Establecemos la presencia o ausencia de estos síntomas mediante la sentencia

- @attribute fiebreReumatica {0, 1}

Donde 1 entablase la presencia y cero la ausencia del síntoma.

En una secuencia de valores como los que se presenta a continuación (línea 106 archivo fiebrereumatica.arff):

1,1,1,1,1,1,1,1,1,1,1,0,1,0,1

La última fila indica como resultado presencia (1)

