

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA EXPERTO PARA EL  
CONTROL DE LA TUBERCULOSIS**



**OSCAR HERNÁN MONDRAGÓN MARTÍNEZ  
JOSE MILCIADES ORDOÑEZ ARGOTE**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE SISTEMAS  
POPAYÁN  
2004**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA EXPERTO PARA EL  
CONTROL DE LA TUBERCULOSIS**



**OSCAR HERNÁN MONDRAGÓN MARTÍNEZ  
JOSE MILCIADES ORDOÑEZ ARGOTE**

**Proyecto de grado presentado como requisito para optar al título de  
Ingeniero en Electrónica y Telecomunicaciones**

**Director  
JUAN CARLOS VIDAL ROJAS  
Ingeniero en Electrónica y Telecomunicaciones**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE SISTEMAS  
POPAYÁN  
2004**

---

Jurado

---

Jurado

Ciudad y fecha (día, mes, año) \_\_\_\_\_

## TABLA DE CONTENIDO

<b>INTRODUCCIÓN</b>	I
<b>1 OBJETIVOS</b>	II
<b>1.1 Objetivo general</b>	II
<b>1.2 Objetivos específicos</b>	II
<b>2 ESTRUCTURA DEL DOCUMENTO</b>	II
<b>CAPÍTULO 1. MARCO TEÓRICO</b>	
	1
<b>1.1 SISTEMAS EXPERTOS</b>	1
<b>1.1.1 Introducción</b>	1
<b>1.1.2 Características de los sistemas expertos</b>	1
<b>1.1.3 Aplicaciones de los sistemas expertos</b>	2
<b>1.1.4 Arquitectura de sistemas expertos</b>	3
<b>1.1.5 Diseño de sistemas expertos</b>	5
1.1.5.1 Planeación	7
1.1.5.2 Definición de conocimiento	7
1.1.5.3 Diseño del conocimiento	7
1.1.5.4 Código y verificación	8
1.1.5.5 Evaluación del sistema	8
<b>1.1.6 Representación del conocimiento</b>	9
1.1.6.1 Reglas de producción	9
1.1.6.2 Redes semánticas	9
1.1.6.3 Marcos	11
<b>1.1.7. Sistemas expertos basados en reglas de producción</b>	12
1.1.7.1 Lógica deductiva y silogismos	12
1.1.7.2 Reglas de producción	13
1.1.7.2.1 Encadenamiento hacia delante y hacia atrás	14

1.1.7.2.2 Modus ponens y modus tollens	15
1.1.7.2.3 Propiedades de las reglas de producción	16
1.1.7.2.4 El manejo de las reglas	16
1.1.7.2.4.1 Memoria de trabajo	17
1.1.7.2.4.2 Mecanismo de inferencia	17
1.1.7.2.5 Resolución de conflictos	17
1.1.7.2.6 Control de la coherencia	18
1.1.7.3 Valoración de las reglas de producción como representación del conocimiento	19

## **CAPÍTULO 2 LOS SISTEMAS EXPERTOS COMO UN APOORTE A LA SOLUCIÓN DE LA PROBLEMÁTICA DEL TRATAMIENTO DE LA TUBERCULOSIS**

20

<b>2.1 LA TUBERCULOSIS: UN PROBLEMA DE SALUD PÚBLICA MUNDIAL.</b>	20
<b>2.1.1 Estrategias y operaciones claves para el control de la tuberculosis</b>	21
<b>2.1.2 Problemática de la tuberculosis en Colombia</b>	22
<b>2.2 POR QUE USAR SISTEMAS EXPERTOS</b>	24

## **CAPÍTULO 3 EXPLORACIÓN TECNOLÓGICA**

27

<b>3.1 EXPLORACIÓN TECNOLÓGICA</b>	27
<b>3.1.1 CLIPS</b>	27
3.1.1.1 Características	27
3.1.1.2 Ventajas	28
3.1.1.3 Desventajas	28
<b>3.1.2 JESS</b>	29
3.1.2.1 Características	29
3.1.2.2 Ventajas	29
3.1.2.3 Desventajas	30

<b>3.1.3 JEOPS</b>	31
3.1.3.1 Características	31
3.1.3.2 Ventajas	32
3.1.3.3 Desventajas	32
<b>3.1.4 Mandarax</b>	33
3.1.4.1 Características	33
3.1.4.2 Ventajas	33
3.1.4.3 Desventajas	34
<b>3.1.5 ILOG JRules</b>	35
3.1.5.1 Características	35
3.1.5.2 Ventajas	35
3.1.5.3 Desventajas	36
<b>3.2 CUADRO COMPARATIVO DE LAS TECNOLOGÍAS</b>	36
<b>3.3 SISTEMAS EXPERTOS DE APOYO EN LA TOMA DE DECISIONES CLÍNICAS.</b>	37
<b>3.3.1 MYCIN</b>	38
<b>3.3.2 PRODIGY</b>	40
<b>3.3.3 PERFEX</b>	41
<b>3.3.4 Murmur Clinic</b>	44
<b>CAPÍTULO 4 SIEXCO-TB COMO HERRAMIENTA DE SOPORTE PARA EL TRATAMIENTO DE LA TUBERCULOSIS</b>	46
<b>4.1 INTRODUCCIÓN</b>	47
<b>4.2 ANÁLISIS DEL SISTEMA</b>	49
<b>4.2.1 Caso del Dominio</b>	49
<b>4.2.2 Casos de uso esenciales</b>	51
4.2.2.1 Caso de uso: Ingresar la sistema	51
4.2.2.2 Caso de uso: Borrar paciente	52
4.2.2.3 Caso de uso: Salir	52
4.2.2.4 Caso de uso: Buscar Paciente	53

4.2.2.5 Caso de uso: ingresar nuevo paciente	53
4.2.2.6 Caso de uso: Cerrar sesión	53
4.2.2.7 Caso de uso: Configurar usuario	54
4.2.2.8 Caso de uso: Configurar régimen afiliación	54
4.2.2.9 Caso de uso: Configurar región	55
4.2.2.10 Caso de uso: Generar tratamiento paciente registrado	56
4.2.2.11 Caso de uso: Generar tratamiento paciente no registrado	56
4.2.2.12 Caso de uso: Ver formula medica	57
4.2.2.13 Caso de uso: Ver registro paciente	58
4.2.2.14 Caso de uso: Ver informe trimestral	58
4.2.2.15 Caso de uso: Ver análisis de cohorte	59
4.2.2.16 Caso de uso: Importar información paciente	59
4.2.2.17 Caso de uso: Importar información de tratamiento	59
4.2.2.18 Caso de uso: Exportar información paciente	60
4.2.2.19 Caso de uso: Exportar información del tratamiento	61
4.2.2.20 Caso de uso: Ver ayudas	62
4.2.2.21 Caso de uso: Realizar análisis estadístico	62
4.2.2.22 Caso de uso: Editar base de conocimiento	63
4.2.2.23 Caso de uso: Editar grupo de indicadores	65
4.2.2.24 Caso de uso: Editar tratamiento	66
4.2.2.25 Caso de uso: configurar servidor BD	67
<b>4.3 DISEÑO DEL SISTEMA</b>	68
<b>4.3.1 Arquitectura</b>	68
4.3.1.1 GUI	68
4.3.1.1 Modulo experto	69
4.3.1.3 Modulo de adquisición del conocimiento	69
4.3.1.4 Modulo explicativo	69
4.3.1.5 Modulo de control	70
4.3.1.6 Servidor de base de datos	70
4.3.1.7 Epi_Info	70
4.3.1.8 Otras herramientas Sinco-TB	71
<b>4.3.2 La Jerarquía de meta datos</b>	71
<b>4.3.3 Definición del conocimiento</b>	75

4.3.3.1	Identificación y selección del origen del conocimiento	75
4.3.3.2	Adquisición, análisis y extracción del conocimiento	76
4.3.3.3	Verificación del conocimiento y Evaluación del sistema	77
<b>4.3.4</b>	<b>Representación del conocimiento</b>	77
4.3.4.1	Reglas de definición del tratamiento	78
4.3.4.2	Reglas de modificación del tratamiento	78
4.3.4.3	Reglas de acciones a realizar	79
<b>4.4</b>	<b>DETALLES DE IMPLEMENTACIÓN</b>	80
<b>4.4.1</b>	<b>Java</b>	80
4.4.1.1	Características de Java	80
4.4.1.2	Uso de Java en SIEXCO – TB	82
4.4.1.2.1	JavaBeans	82
4.4.1.2.2	JDOM	85
4.4.1.2.3	Java Foundation Classes (JFC/Swing)	86
4.4.1.2.4	JavaMail	89
4.4.1.2.5	JDBC	90
4.4.1.2.6	JavaHelp	90
4.4.1.2.7	Mandarax	91
<b>4.4.2.</b>	<b>XML (eXtensible Markup Language)</b>	92
4.4.2.1	APIs de Java para XML	93
4.4.2.2	Uso de XML en SIEXCO-TB	94
<b>4.5</b>	<b>HERRAMIENTAS DE DESARROLLO</b>	97
<b>4.5.1</b>	<b>Oryx</b>	97
<b>4.5.2</b>	<b>Mandarax</b>	98
4.5.2.1	Clause Sets	100
4.5.2.2	Asignación de prioridades en la base de conocimientos	101
4.5.2.3	El proceso de inferencia	101
4.5.2.4	Trabajando con resulset	102
4.5.2.5	Conexión a bases de datos	103
4.5.2.6	Integración del modelo de objetos java	104
4.5.2.7	Base de conocimiento en XML	107
4.5.2.8	XKB versus RuleML	108



<b>4.5.3 MySql</b>	109
4.5.3.1 Diseño de la base de datos relacional de SIEXCO-TB	109
<b>4.5.4 Epi_Info</b>	112
4.5.4.1 Siexco_TB y Epi_Info	112
4.5.4.2 Análisis de datos con Epi_Info	115
<b>CONCLUSIONES</b>	116
<b>RECOMENDACIONES</b>	119
<b>BIBLIOGRAFÍA</b>	120

## LISTA DE FIGURAS

Figura 1 Arquitectura de Sistemas Expertos.....	4
Figura 2 Modelo Lineal del Ciclo de Vida para Desarrollar Sistemas Expertos.....	7
Figura 3 Ejemplo de una Red Semántica.....	10
Figura 4 Caso del dominio .....	50
Figura 5 Diagrama de Casos de Uso.....	52
Figura 6 Arquitectura de SIEXCO - TB.....	69
Figura 7 Jerarquía de Meta datos SIEXCO - TB.....	72
Figura 8 Grupo de Indicadores Básico usado por SIEXCO - TB.....	74
Figura 9 Meta modelo del tratamiento .....	84
Figura 10 GUI Mostrada para el Informe Trimestral de Casos y Actividades.....	89
Figura 11 GUIs usadas para desplegar la tarjeta individual del paciente .....	90
Figura 12 GUI para enviar vía correo electrónico los archivos generados por SIEXCO- TB....	91
Figura 13 Ayudas de SIEXCO - TB.....	93
Figura 14 Jerarquía de Lenguajes XML.....	94
Figura 15. Diagrama Base de Datos.....	112
Figura 16 Tabla de signos y síntomas .....	115
Figura 17 Tabla de signos de toxicidad .....	116
Figura 18 Tabla para análisis epidémico .....	117

## **LISTA DE ANEXOS**

ANEXO A. DIAGRAMAS DE DISEÑO DEL SISTEMA

ANEXO B. MANUAL DE USUARIO

## **INTRODUCCIÓN**

La Tuberculosis representa un grave problema de salud pública en Colombia, lo cual hace necesario evitar su incremento. Se requiere romper la cadena de transmisión de enfermo a sano mediante la búsqueda de los enfermos, y el suministro de un adecuado tratamiento supervisado.

Las administradoras de salud tienen la responsabilidad de garantizar una atención individual integral a su población afiliada y beneficiaria. Por lo tanto deben procurar actividades, intervenciones y procedimientos de educación, información y promoción de la salud. Así como también, prevención, diagnóstico, tratamiento y rehabilitación de los enfermos de tuberculosis. Por lo tanto, se hace necesario definir y estandarizar las actividades, intervenciones y procedimientos técnico-científicos y administrativos que garanticen la atención integral del usuario y contribuyan a controlar la tuberculosis en nuestro país. [1].

Conscientes de la problemática, las universidades del país deben convertirse en una apoyo importante en la búsqueda de procedimientos y soluciones ajustadas al contexto nacional. Lo anterior ha motivado en el presente trabajo el desarrollo de SIEXCO-TB, que está en capacidad de sugerir un tratamiento de acuerdo a un esquema determinado. La información de soporte es la norma nacional para el tratamiento de la tuberculosis, la experiencia de médicos especialistas y la información particular del paciente. Adicionalmente el sistema provee información estadística útil para establecer indicadores de efectos colaterales originados por el consumo de medicamentos, los cuales contribuirán a prevenir trastornos futuros en los pacientes.

De igual forma, atendiendo a la necesidad de hacer más eficientes los procesos administrativos relacionados con el manejo de la enfermedad por parte de las instituciones de salud colombianas, el sistema da soporte a las diferentes actividades de gestión estipuladas en la guía de atención de la tuberculosis Pulmonar y Extrapulmonar.

Este proyecto pretende consolidar el trabajo en equipo multidisciplinario entre la Facultad de Ciencias de la salud y la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, así como afianzar las políticas de proyección social establecidas en la misión de la universidad.

## **1. OBJETIVOS**

### **1.1. Objetivo general**

Desarrollar un sistema experto que facilite la toma de decisiones terapéuticas, de los profesionales de la salud, en el tratamiento de pacientes afectados por la tuberculosis de acuerdo a la guía de atención de la tuberculosis pulmonar y extrapulmonar expedida por el ministerio de Protección Social.

### **1.2. Objetivos específicos**

- Diseñar e Implementar un sistema experto que use como información de soporte la norma nacional para el control de la tuberculosis, donde se especifican protocolos, tratamientos y esquemas a seguir en el proceso de atención del paciente.
- Llevar un registro de la información particular de cada paciente y de control de seguimiento.
- Proporcionar, al profesional de la salud, información relacionada con datos clínicos del paciente, además de los medicamentos que deben ser formulados de acuerdo al tratamiento, así como sus contraindicaciones, dosificación y efectos colaterales.
- Permitir que el profesional de la salud pueda seguir diferentes esquemas en el tratamiento, de acuerdo a aspectos como: Casos nuevos, recaídas, reingreso de abandono, fracaso de tratamiento y casos crónicos.
- Generar información estadística complementaria a la norma, de acuerdo a los tratamientos suministrados a los pacientes, mediante el suministro de indicadores de efectos colaterales que contribuyan a prevenir trastornos futuros en los pacientes.

## **2. ESTRUCTURA DEL DOCUMENTO**

Este documento presenta el trabajo hecho durante la ejecución del proyecto que dio como resultado a SIEXCO-TB. Se muestran los conceptos básicos de sistema expertos, la exploración tecnológica realizada y los detalles de diseño e implementación de la herramienta. La estructura del documento es la siguiente:

En el capítulo 1 se abordará la teoría de sistemas expertos. Además, comprenderá un análisis de las metodologías y herramientas de desarrollo para sistemas expertos como resultado de las fases de exploración y evaluación de las tecnologías de soporte para el proyecto.

En el capítulo 2, se presentan los sistemas expertos como un aporte a la solución de la problemática del tratamiento de la Tuberculosis. En este capítulo hará una reflexión sobre la tuberculosis como un grave problema de salud pública en Colombia y cómo un sistema experto puede ser una herramienta útil para su tratamiento.

En el capítulo 3 se presentan los resultados de la exploración tecnológica realizada dentro del proyecto SIEXCO -TB, caracterizando cada una de las herramientas estudiadas, enunciando sus ventajas y desventajas, y mostrando el cuadro comparativo que permitió elegir una de ellas para la construcción de SIEXCO-TB. Además se hace énfasis en los sistemas expertos de apoyo a la toma de decisiones clínicas, analizando sistemas existentes e identificando las características que sirvieron de base para la construcción de SIEXCO-TB.

En el capítulo 4 se describe el sistema desarrollado, explicando su diseño modular, arquitectura, modelo de representación del conocimiento y demás características del diseño. De la misma forma, se exponen algunos detalles de implementación y las tecnologías empleadas en el proceso de desarrollo.

Finalmente se presentan las conclusiones y perspectivas del presente trabajo.

## CAPÍTULO 1

### MARCO TEÓRICO

#### 1.1 SISTEMAS EXPERTOS

##### 1.1.1 Introducción

Los sistemas expertos son una rama de la inteligencia artificial que hace un amplio uso del conocimiento especializado para resolver problemas como un especialista humano. Éste último tiene experiencia y conocimientos especiales que le permite resolver problemas que la mayoría no podría resolver, o los resuelve con mucha mayor eficiencia. Cuando los sistemas expertos se desarrollaron por primera vez en los años setenta, contenían exclusivamente conocimiento experto. Sin embargo, hoy en día se aplica el término *sistema experto* a cualquier sistema que utiliza tecnología de sistemas expertos. Esta tecnología puede incluir a los lenguajes y programas especiales de sistemas expertos, además del hardware diseñado para ayudar en su desarrollo y ejecución.

El conocimiento de los sistemas expertos puede obtenerse por experiencia o consulta de los conocimientos que suelen estar disponibles en libros, revistas y con personas capacitadas. Los términos *sistema experto*, *Sistema basado en conocimiento*, o *sistema experto basado en conocimiento*, se usan como sinónimo. La mayoría utiliza el término sistema experto tan solo porque es más corto, a pesar de que quizá no tengan experiencia en su sistema, sólo conocimiento general. Sin embargo vale la pena aclarar que la expresión *sistema basado en conocimiento* es un mejor término para la aplicación de tecnologías basadas en el conocimiento, que pueden usarse para la creación de sistemas expertos o basados en el conocimiento, en estos últimos el conocimiento no suele estar al nivel de un especialista humano [2].

##### 1.1.2 Características De Los Sistemas Expertos

Los sistemas expertos actuales cuentan con las siguientes características [2]:

- Las soluciones brindadas por el sistema deben tener un nivel de calidad igual o superior al brindado por un especialista en el campo.

- El tiempo de respuesta de un sistema experto debe ser comparable o menor que el utilizado por un experto humano en la solución de un problema.
- De la misma forma que un especialista puede explicar su razonamiento para llegar a una respuesta, el sistema experto debe ser capaz de explicar la estrategia de solución encontrada y el porqué de las decisiones tomadas. Esta característica es importante principalmente por dos razones. La primera es que proporciona una revisión del razonamiento comprensible para los humanos y la segunda se encuentra en la fase de desarrollo porque permite confirmar que el conocimiento ha sido adquirido y está siendo utilizado correctamente por el sistema.
- Un especialista humano puede adquirir experiencia y nuevo conocimiento en su especialidad. De la misma forma, un sistema experto debe tener un mecanismo eficiente para añadir, modificar y eliminar conocimiento. Una de las razones por la cual los sistemas basados en reglas son tan populares es la capacidad de almacenamiento modular y eficiente de las reglas.
- Los sistemas expertos se utilizan en situaciones donde no hay una solución algorítmica eficiente, es decir el sistema trata con entradas inesperadas que no siguen un patrón determinado.
- Los sistemas expertos separan explícitamente la base de conocimientos del mecanismo de inferencia. Esto significa que puede reutilizarse el núcleo básico del sistema experto (Mecanismo de inferencia) para construirse un nuevo sistema experto.
- Las aplicaciones desarrolladas con sistemas expertos no se orientan a resolver problemas de tipo general, las soluciones brindadas por los sistemas expertos se encuentran delimitadas por un área de conocimiento específica de la misma manera que un especialista humano es experto en su área de interés.

### **1.1.3 Aplicaciones de los Sistemas Expertos.**

A pesar de ser una tecnología aparentemente reciente los sistemas expertos se han aplicado en casi todos los campos del conocimiento, a continuación se muestran algunas clases de sistemas expertos [2]:



Clase	Descripción
Configuración	Ensamblar correctamente los componentes apropiados de un sistema.
Diagnostico	Inferir problemas basándose en la evidencia observada.
Instrucción	Enseñanza inteligente para que un estudiante pueda preguntar <i>por que, como y que pasaría si</i> , como lo haría un maestro humano.
Interpretación	Explicar los datos observados.
Supervisión	Comparar los datos observados con los esperados para juzgar el desempeño.
Planeación	Idear acciones para obtener el resultado esperado.
Pronóstico	Predecir el resultado de una situación dada.
Remedio	Prescribir el tratamiento para un problema.
Control	Regular un proceso. Talvez requiera de interpretación, diagnostico, supervisión, plantación, pronóstico y remedio.

#### 1.1.4 Arquitectura de Sistemas Expertos.

Según el profesor Edward Feigenbaum de la Universidad de Stanford Un sistema experto puede definirse como “un programa de computación inteligente que usa el conocimiento y los procedimientos de inferencia para resolver problemas que son lo suficientemente difíciles como para requerir significativa experiencia humana para su solución”. El conocimiento del experto en ese campo se organiza en una *base de conocimientos*, y en función de los datos disponibles de la aplicación (*base de datos o hechos*) se imita la forma de actuar del experto explorando en la base de conocimientos hasta encontrar la solución (*motor de inferencia*). Los resultados finales y la forma en que se obtienen se expresan a través de la *interfaz de usuario* [3, 4, 5]. La figura 1 muestra la arquitectura de un sistema experto.

- **Experto:** Es la persona que posee el conocimiento y la experiencia que se ingresará al sistema. El conocimiento también puede extraerse de otro tipo de fuentes como documentos escritos.
- **Ingeniero de conocimiento:** Trabaja en conjunto con el experto para obtener su conocimiento, estructurarlo e implementarlo en la base de conocimientos.

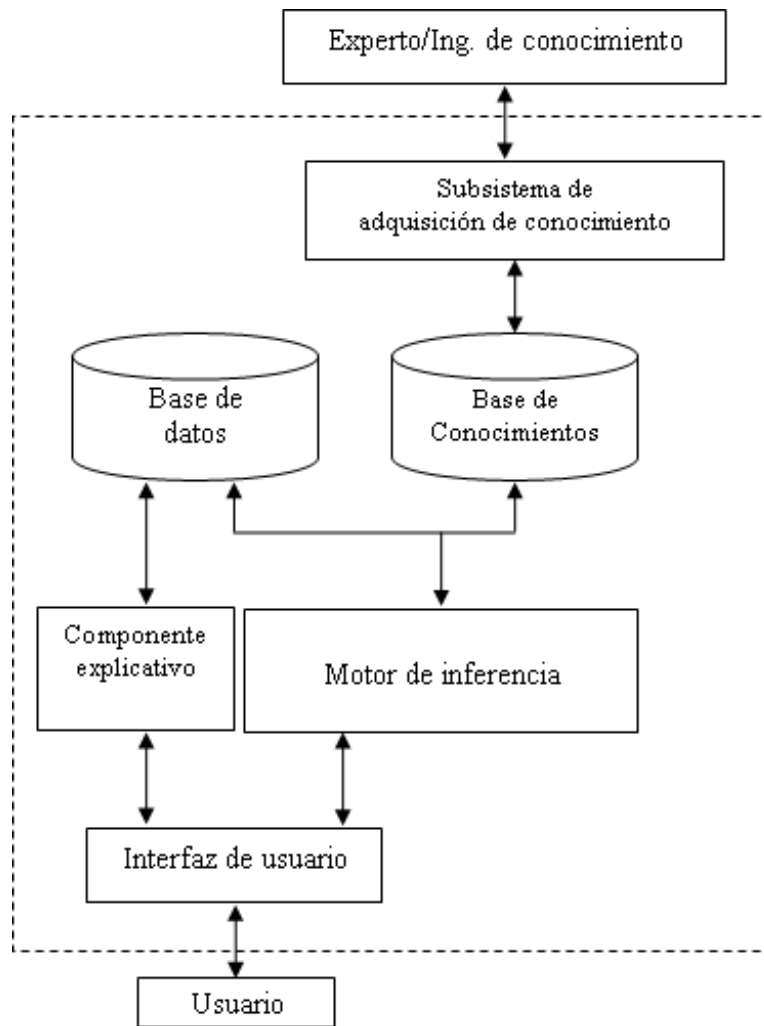


Figura 1 Arquitectura de Sistemas Expertos

- **Subsistema de adquisición de conocimiento:** Este componente permite que se puedan añadir, eliminar o modificar elementos de conocimiento en el sistema experto. Un buen componente de adquisición facilitara considerablemente la labor del Ingeniero del Conocimiento en la etapa de desarrollo y permitirá mantener actualizada la base de conocimiento durante su utilización.
- **Base de conocimiento:** Es uno de los módulos más importantes de los sistemas expertos porque contienen la representación del conocimiento obtenido del especialista humano y en algunas ocasiones de fuentes escritas. El conocimiento es estructurado e implementado en la base de conocimientos por el ingeniero de conocimiento. Una de las formas más extendida de representación del conocimiento es mediante reglas de producción, de esta

forma la representación del conocimiento se divide en pequeñas fracciones de conocimiento o reglas SI . . . ENTONCES.

- **Base de datos:** La base de datos se utiliza para almacenar los datos necesarios para la resolución de un problema, por lo tanto contiene conocimiento sobre el caso concreto en que se trabaja.
- **Componente explicativo:** La mayoría de los sistemas expertos contienen un módulo de explicación, diseñado para aclarar al usuario la línea de razonamiento seguida en el proceso de inferencia. Si el usuario pregunta al sistema cómo ha alcanzado una conclusión, éste le presentará la secuencia completa de reglas usada. Esta posibilidad de explicación es especialmente valiosa cuando se tiene la necesidad de tomar decisiones importantes amparándose en el consejo del sistema experto.
- **Motor de inferencia:** El motor de inferencias es un programa que controla el proceso de razonamiento que seguirá el sistema experto, según un método fijo de solución de problemas que esta configurado imitando el procedimiento humano de los expertos para solucionar problemas.
- **Interfaz de usuario:** Es el medio de comunicación entre el usuario y el sistema. Permite que el usuario pueda describir el problema al sistema experto, despliega la información del componente explicativo, muestra los resultados obtenidos y emite mensajes al usuario.
- **Usuario:** Es la persona que utilizará el sistema experto, por lo tanto gran parte de las características del escenario en que se usará el sistema serán suministradas por el.

### 1.1.5 Diseño De Sistemas Expertos

Usar el paradigma de los sistemas expertos para desarrollar una aplicación depende de varios factores. A continuación se formulan algunas preguntas que se deben tener en cuenta al momento de escoger la tecnología de sistemas expertos para implementar la solución [2].

- *¿Puede solucionarse eficazmente el problema con programación convencional?* Los sistemas expertos son más adecuados en los problemas en que no hay una solución algorítmica eficiente. A esos casos se les llama problemas de estructura nociva y solo el razonamiento puede ofrecer esperanzas de una solución adecuada.

- *¿Está bien delimitado el dominio?* Es importante tener límites bien definidos en lo que se espera sepa el sistema experto y en las aptitudes que debe tener. Si bien es cierto que entre más conocimiento tenga el sistema experto se podrá enfrentar a un mayor número de problemas, también es cierto que su desarrollo se volverá más complejo.
- *¿Hay la necesidad y el deseo de tener un sistema experto?* No tiene caso desarrollar un sistema experto si nadie lo utilizará. Si hay muchos especialistas humanos es difícil justificar un sistema experto con base en la escasez de experiencia humana.
- *¿Hay al menos un especialista que esté dispuesto a cooperar?* Debe haber un especialista humano en lo posible entusiasmado que esté dispuesto a brindar su conocimiento y experiencia para que sea evaluado, estructurado e ingresado en el sistema experto.

Por lo general la tecnología de los sistemas expertos tiene la misión de proporcionar experiencia en condiciones de alto desempeño y posiblemente arriesgadas donde la vida humana puede estar en juego, esto conlleva a que el sistema debe tener alta calidad o estará propenso a errores. La ingeniería de software proporciona metodologías para construir software de calidad.

Uno de los métodos clave de la ingeniería de software es el ciclo de vida. Proceso que se sigue para construir, entregar y hacer evolucionar el software desde la concepción de una idea hasta la entrega y el retiro del sistema.

Uno de los modelos de ciclo de vida utilizados con éxito en el desarrollo de varios sistemas expertos es el modelo lineal, mostrado en la siguiente figura [2].

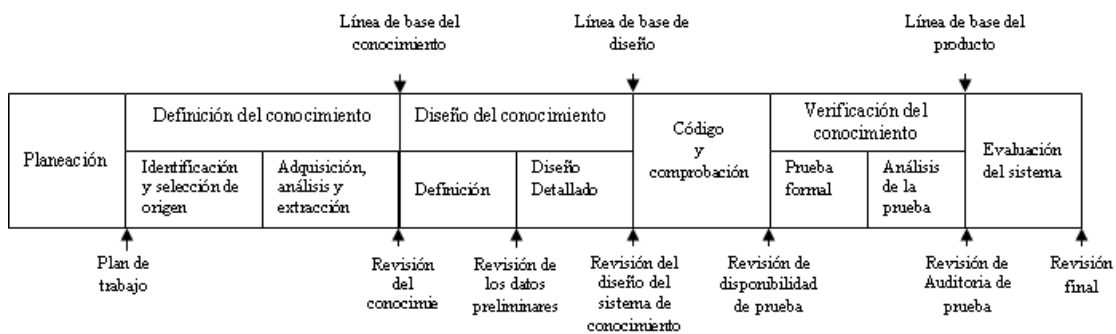


Figura 2 Modelo Lineal del Ciclo de Vida para Desarrollar Sistemas Expertos

Este ciclo de vida está formado por varias etapas, que van de la planeación a la evaluación de sistemas, y describe el desarrollo del sistema hasta el punto en que se evaluarán sus capacidades funcionales. Después de esto, el ciclo de vida repite la misma secuencia de planeación a evaluación hasta que el sistema se distribuye para uso rutinario; en consecuencia, se usa para el mantenimiento y la evolución posteriores del sistema.

#### **1.1.5.1 Planeación**

El objetivo de la etapa de planeación es producir un plan de trabajo formal para desarrollar el sistema experto. El resultado de esta etapa es un conjunto de documentos que se utilizarán para guiar y evaluar el desarrollo.

Una de las tareas más importantes para el ciclo de vida del sistema en esta etapa es la valoración de factibilidad. Determina si el proyecto vale la pena y si un sistema experto es el paradigma apropiado, además incluye el análisis de costo, utilidad y otros factores del sistema experto.

#### **1.1.5.2 Definición De Conocimiento**

En esta etapa se debe definir el conocimiento requerido por el sistema experto. En esta etapa se distinguen dos tareas principales. A continuación se explica en que consiste cada una.

- 1. Identificación y selección del origen del conocimiento:** En esta etapa se debe identificar todas las fuentes de conocimiento. Después se debe priorizar las fuentes de acuerdo a su importancia y disponibilidad. Usualmente es más fácil disponer de libros y otros documentos que de especialistas humanos.
- 2. Adquisición, análisis y extracción del conocimiento:** El objetivo principal esta tarea es producir y verificar el conocimiento que necesita el sistema.

#### **1.1.5.3 Diseño Del Conocimiento**

En esta etapa se debe producir el diseño detallado del conocimiento para el sistema experto. Las dos tareas principales que integran esta etapa son:

1. **Definición del conocimiento:** En esta etapa se debe especificar como se representará el conocimiento como: reglas, marcas o lógica, dependiendo de la herramienta de sistemas expertos que le dará soporte. También se debe definir una estructura de control detallada, una estructura interna de hechos, la interfaz preliminar de usuario y un plan de pruebas inicial.
2. **Diseño detallado:** El objetivo final de la etapa de diseño detallado es el documento de diseño que se toma como referencia para proceder con la codificación. En esta etapa se distinguen las siguientes tareas:
  - **Estructura de diseño:** Especificar como se organizará lógicamente el conocimiento en la base de conocimientos.
  - **Estrategia de implantación:** Especificar la forma en que se establece el sistema.
  - **Interfaz detallada de usuario:** Especificar la interfaz del usuario detallada después de recibir retroalimentación por el uso del diseño preliminar.
  - **Documentar el diseño**
  - **Plan detallado de pruebas:** Especificar claramente cómo se probará y especificará el código.

#### 1.1.5.4 Código y verificación

Esta etapa contempla las siguientes tareas:

- Codificación
- Pruebas
- Código fuente comentado y documentado
- Guía de instalación

#### 1.1.5.5 Evaluación del sistema

En esta etapa se debe verificar que el sistema sea correcto, este completo, y sea congruente. Esta etapa se divide en dos tareas principales:

1. **Pruebas formales:** En esta tarea se deben realizar los procedimientos de prueba formal y documentar los resultados obtenidos.

- 2. Análisis de pruebas:** En el análisis de pruebas se hace énfasis en los problemas relacionados con respuestas incorrectas, respuestas incompletas y repuestas incongruentes. Se identifica el origen de los problemas. Se analizan los resultados de la prueba y se documentan las recomendaciones y conclusiones de la prueba.

### **1.1.6 Representación Del Conocimiento**

Existen varias técnicas de representación del conocimiento las más comunes son reglas de producción, redes semánticas y marcos. A continuación se explica en que consiste cada una [2, 3, 4, 5, 6].

#### **1.1.6.1 Reglas de producción**

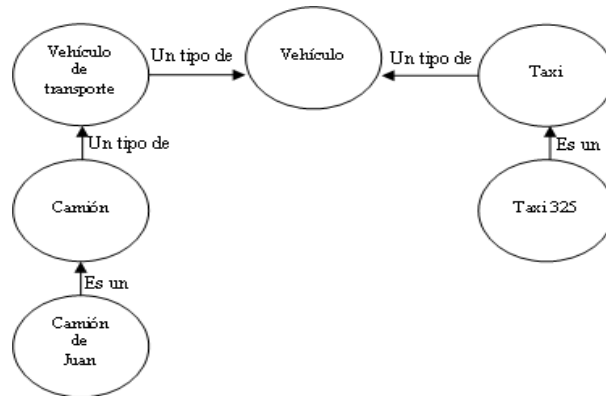
Es la forma más extendida y más sencilla para la representación del conocimiento. De esta forma la representación del conocimiento se divide en pequeñas fracciones de conocimiento o reglas SI . . . ENTONCES. Por tratarse de un de las técnicas con mayores beneficios se realizará un mayor análisis en el numeral 1.1.8 Sistemas expertos basados en reglas de producción.

#### **1.1.6.2 Redes semánticas**

Las redes semánticas son una forma clásica de representación del conocimiento de la inteligencia artificial. A estas redes también se le llama redes de proposiciones por que se basan en proposiciones, Una proposición es una frase verdadera o falsa y se le llama atómica porque su veracidad ya no puede dividirse más.

La estructura de una red semántica se puede visualizar gráficamente en términos de nodos y los arcos que los conectan. Los nodos suelen definirse como objetos y los arcos como vínculos o bordes. Los nodos por lo general se utilizan para representar objetos físicos, conceptos o situaciones y los arcos se utilizan para expresar la relación que existe entre los nodos, un ejemplo de frase que se podría representaren en una red semántica es *el perro es un mamífero* en donde existirían dos nodos representados por *perro* y *mamífero* y el vinculo entre ellos seria *es un*.

Existen dos tipos de vínculos utilizados con frecuencia en una red semántica: ES-UN y UN TIPO DE a continuación se muestra una red semántica que muestra este tipo de relación. El vínculo UN TIPO DE se utiliza para relacionar una clase individual con una clase padre, mientras que ES UN relaciona un caso o un individuo con una clase genérica.



**Figura 3 Ejemplo de una Red Semántica**

Uno de los métodos de inferencia utilizado en redes semánticas es la coincidencia de patrones de manera parecida a la coincidencia de patrones en las reglas de producción, Sin embargo en las redes semánticas la búsqueda empieza desde la primera instrucción que se introduce, la superior, y sigue hasta la última. Uno de los mayores inconvenientes de este método de búsqueda se da cuando las respuestas son negativas. En estos casos tendría que comparar todos o la mayoría de los vínculos para concluir que ninguno coincide y por lo tanto deducir una falsedad.

Las redes semánticas tienen algunas limitaciones. A continuación se mencionan algunas de ellas.

- No hay estándares definidos para asignar el nombre a los vínculos, esto hace difícil verificar si la red se diseñó en forma congruente y entender para que se diseñó la red.
- Cuando una consulta tiene como resultado una respuesta negativa el sistema tendría que buscar en todos o la mayoría de los vínculos de la red. Aunque no todas las redes necesitan un alto grado de conectividad existe la posibilidad de que se genere una explosión combinatoria.
- Las redes semánticas se utilizan principalmente para representar relaciones binarias porque una línea tiene solo dos extremos. Una representación bidimensional del conocimiento dificulta su representación lógica.



- Las redes semánticas son heurísticamente inadecuadas por que no hay forma de insertar en la red información que ayude a explotarla de forma eficiente.

### 1.1.6.3 Marcos

Para hablar de marcos se debe conocer primero el significado de *esquema* debido a que los marcos hacen parte o son un tipo de esquema. En la Inteligencia artificial el término esquema se utiliza para describir una estructura de conocimiento más compleja que una red semántica. Habitualmente los esquemas tienen un estructura interna en sus nodos, mientras que las redes semánticas no, en ellas el rotulo es todo el conocimiento acerca del nodo. Un esquema es como una estructura de datos donde los nodos contienen registros. Cada registro puede incluir datos, registros o apuntadores a otros nodos.

Informalmente, un marco es una estructura de datos compleja que representa un objeto o una clase de objetos. Cada marco posee un número de casillas (*slots*) que representan las propiedades o atributos del objeto. Al igual que las redes semánticas, Un marco se puede visualizar como una red de *nodos* y relaciones entre nodos (*arcos*). Los slots se pueden llenar con otros marcos, pueden tener valores por defecto, o estar restringidos a rangos específicos de valores. A nivel práctico, se pueden considerar los marcos como una red semántica con un número de posibilidades mucho mayor, entre las que destacan especialmente, la capacidad de activación de procesos y de herencia en la que un nodo hijo puede heredar todos los *slots* de su padre a menos que se especifique lo contrario. Al utilizarse marcos en las ranuras de relleno y en la herencia, pueden construirse sistemas de representación del conocimiento muy versátiles.

Como es evidente, la representación de conocimiento basada en marcos se debe mucho a las redes semánticas. Sin embargo, los esquemas de representación basados en marcos insisten en una organización jerárquica de éstos. La estructura de nodos de los marcos es también mucho más rica que la de las redes semánticas, conteniendo sistemas de activación de procesos (*triggering*). Esto ocurre cuando en lugar de llenar un slot con un valor determinado, se indica un procedimiento que será el encargado de devolver un determinado valor.

A continuación se muestra un marco que describe un león

Nombre: León  
Especialización de: Un tipo de mamífero  
Color: Castaño  
Patas: 4  
Tipo: Carnívoro

El paradigma de los marcos tiene un aspecto intuitivo, porque la representación del conocimiento suele ser más fácil de comprender que cualquier sistema lógico o de producción con muchas reglas, sin embargo han aparecido problemas importantes en los sistemas de marcos. A continuación se mencionan algunos.

- Por lo general ningún sistema de marcos proporciona una manera de definir slots inalterables. Cualquier slots puede cambiarse de modo que pueda modificarse o cancelarse las propiedades que un marco hereda de cualquier lugar de la jerarquía. Nada es realmente cierto en un sistema sin restricciones como este.
- Del mismo modo que las redes semánticas no hay estándares para definir los sistemas basados en marcos.

### 1.1.7 Sistemas Expertos Basados en Reglas de Producción

#### 1.1.7.1 Lógica Deductiva y Silogismos

Uno de los métodos más comunes para formular inferencias es la lógica deductiva, utilizada para determinar la validez de un argumento. Un argumento lógico es un grupo de afirmaciones en el que la última de ellas está justificada en las anteriores de la cadena de razonamiento. El silogismo es un tipo de argumento lógico. [2]

Ejemplo:

*Premisa:* Cualquier paciente que tenga tos por más de 15 días es sintomático respiratorio.

*Premisa:* Pedro tiene tos por más de 15 días

---

*Conclusión:* Pedro es sintomático respiratorio

A las premisas, las cuales justifican la conclusión, se les llama **antecedentes**, mientras que la conclusión es la **consecuencia**. La principal característica de la lógica deductiva es que una conclusión es verdadera cuando las premisas también lo son.

Los silogismos son útiles por que pueden expresarse en términos de reglas SI... ENTONCES. Por ejemplo, podemos describir el silogismo anterior así:

**SI** Cualquier paciente que tenga tos por más de 15 días es sintomático respiratorio Y Pedro tiene tos por más de 15 días

**ENTONCES** Pedro es sintomático respiratorio.

Forma	Esquema	Significado
A	Toda A es B	Afirmativo Universal
E	No A es B	Negativo Universal
I	Alguna A es B	Afirmativo Particular
O	Alguna A no es B	Negativo Particular

**Tabla 1 . Afirmaciones Categóricas**

En conclusión, un silogismo es cualquier argumento deductivo válido que cuenta con dos premisas y una conclusión. El silogismo clásico es un tipo especial llamado **silogismo categórico**; las premisas y conclusiones se definen como afirmaciones categóricas como se muestra en la Tabla 1.

### 1.1.7.2 Reglas de Producción

Los silogismos solo soportan una pequeña parte de las afirmaciones lógicas posibles. Particularmente los silogismos categóricos solo soportan afirmaciones de la forma A, E, I, O.

La lógica ofrece otros mecanismos para describir argumentos. Por ejemplo, considerando el siguiente argumento de proposición:

Si hay combustible en el tanque , el automóvil arrancará

Hay combustible en el tanque

---

POR LO TANTO: El automóvil arrancará

Este argumento se puede representar formalmente mediante letras, así:

p = Hay combustible en el tanque

q = El automóvil arrancará

$$\begin{array}{l} p \longrightarrow q \\ p \\ \hline \therefore q \end{array}$$

Este esquema general, es el usado para la representación de argumentos de este tipo. p y q son variables lógicas que pueden representar cualquier información. Estas variables lógicas permiten afirmaciones más complejas que las cuatro formas silogísticas A, E, I y O.

El ejemplo anterior también puede expresarse en forma silogística así:

Todos los automóviles con combustible en el tanque arrancarán

Este automóvil tiene combustible en el tanque

---

$\therefore$  Este automóvil arrancará

### 1.1.7.2.1 Encadenamiento hacia adelante y hacia atrás

A un grupo de varias inferencias que conecta un problema con su solución se le llama cadena. Una cadena se puede recorrer desde el problema hasta su solución o desde la solución hasta el problema. El primer caso se denomina encadenamiento hacia adelante y el segundo encadenamiento hacia atrás.

El encadenamiento hacia adelante se puede describir como el razonamiento que va de los hechos a las conclusiones que se desprende de ellos. Una buena aplicación para el encadenamiento hacia adelante ocurre si el árbol es ancho y no muy profundo; esto se debe a

que el encadenamiento hacia adelante facilita primero una búsqueda hacia lo ancho, es decir este encadenamiento es bueno si la búsqueda de conclusiones avanza nivel por nivel. Por otra parte en el encadenamiento hacia adelante la explicación se dificulta un poco, por que los sub objetivos no se conocen de manera explícita si no hasta que se descubren.

El encadenamiento hacia atrás se puede describir desde el punto de vista de un objetivo que puede lograrse al satisfacer los sub-objetivos. El encadenamiento hacia atrás se aplica eficientemente en árboles estrechos y profundos debido a que en este tipo de encadenamiento se facilita primero una búsqueda a fondo. Además con el encadenamiento hacia atrás se facilita la explicación ya que el sistema puede explicar con facilidad y exactitud que objetivo trata de cumplirse [2].

#### **1.1.7.2.2 Modus Ponens y Modus Tollens**

El Modus Ponens es quizás la regla de inferencia más comúnmente utilizada. Se utiliza para obtener conclusiones simples. En ella, se examina la premisa de la regla, y si es cierta, la conclusión pasa a formar parte del conocimiento. Como ilustración, supóngase que se tiene la regla, *Si A es cierto, entonces B es cierto* y que se sabe además que *A es cierto*. La regla Modus Ponens concluye que *B es cierto*. Esta regla de inferencia, que parece trivial, debido a su familiaridad, es la base de un gran número de sistemas expertos.

La regla de inferencia Modus Tollens se utiliza también para obtener conclusiones simples. En este caso se examina la conclusión y si es falsa, se concluye que la premisa también es falsa. Por ejemplo, supóngase de nuevo que se tiene la regla, *Si A es cierto, entonces B es cierto* pero se sabe que *B es falso*. Entonces, utilizando la regla Modus Ponens no se puede obtener ninguna conclusión pero la regla Modus Tollens concluye que *A es falso*.

El rendimiento del motor de inferencia depende del conjunto de reglas en su base de conocimiento. Hay situaciones en las que el motor de inferencia puede concluir utilizando un conjunto de reglas, pero no puede, utilizando otro (aunque éstos sean lógicamente equivalentes).

<b>Modus Ponens</b>	$  \begin{array}{l}  p \longrightarrow q \\  p \\  \hline  \therefore q  \end{array}  $
<b>Modus Tollens</b>	$  \begin{array}{l}  p \longrightarrow q \\  \sim q \\  \hline  \therefore \sim p  \end{array}  $

**Tabla 2 Modus Ponens y Modus Tollens**

### 1.1.7.2.3 Propiedades de las Reglas de Producción

Las reglas de producción tienen en general, las siguientes propiedades:

*Modularidad:* Cada regla define una pequeña e independiente parte de conocimiento

*Incrementalidad:* Nuevas reglas pueden ser añadidas a la base de conocimiento relativamente independiente de las demás

*Modificabilidad:* Como consecuencia de la modularidad, las reglas viejas pueden ser modificadas

*Transparencia:* Habilidad de explicar sus decisiones y soluciones

### 1.1.7.2.4 El Manejo de las Reglas

Un sistema de producción tiene principalmente: un conjunto de reglas (base de conocimiento), un interprete de reglas o mecanismo de inferencia (que decide que regla aplicar) y una memoria de trabajo (que guarda los datos, metas, y resultados intermedios).

#### **1.1.7.2.4.1 Memoria de Trabajo**

La memoria de trabajo guarda inferencias temporalmente. Es la estructura de datos que es observada y manipulada por las reglas.

Los datos de la memoria de trabajo son los que permiten cumplir las condiciones de las reglas y dispararlas. Las reglas disparadas modifican, añaden, quitan elementos de la memoria de trabajo o producen efectos secundarios.

#### **1.1.7.2.4.2 Mecanismo de Inferencia**

El mecanismo de inferencia controla que reglas se disparan.

Generalmente el ciclo se empieza con unos datos iniciales y se para cuando no hay reglas aplicables.

El intérprete o máquina de inferencia realiza el ciclo reconoce-actúa:

1. Aparea las condiciones de las reglas con los elementos de la memoria de trabajo
2. Si existe más de una regla que puede disparar, escoge una (resolución de conflictos)
3. Aplica la regla.

#### **1.1.7.2.5 Resolución de Conflictos**

En razonamiento hacia delante es posible que exista más de una regla que se puede disparar. En razonamiento hacia atrás puede existir más de una regla que cumple con las metas.

Lo que se necesita es una estrategia de resolución de conflictos que decida cual de las posibles reglas disparar.

Existen ciertas estrategias para solucionar conflictos, entre las cuales se tiene:

- No permitir disparar una regla más de una vez con los mismos datos
- Preferir reglas que utilizan datos más recientes (la idea es de seguir una línea de razonamiento)

- Preferir reglas que son más específicas. Por ejemplo, aquellas reglas que tienen una mayor cantidad de condiciones y por lo tanto son más difíciles de disparar.
- Establecer prioridades en las reglas u ordenarlas.
- Establecer prioridades para los hechos

Algunos sistemas permiten tener reglas que razonan en cuanto a que reglas disparar (meta-reglas). Las meta-reglas tienen el rol de dirigir el razonamiento más que realizar el razonamiento.

#### **1.1.7.2.6 Control de la Coherencia**

En situaciones complejas, incluso verdaderos expertos pueden dar información inconsistente. Por ello, es muy importante controlar la coherencia del conocimiento tanto durante la construcción de la base de conocimiento como durante los procesos de adquisición de datos y razonamiento [11].

Si la base de conocimiento contiene información inconsistente (por ejemplo, reglas y/o hechos), es muy probable que el sistema experto se comporte de forma poco satisfactoria y obtenga conclusiones absurdas.

El objetivo del control de la coherencia consiste en:

1. Ayudar al usuario a no dar hechos inconsistentes, por ejemplo, dándole al usuario las restricciones que debe satisfacer la información demandada.
2. Evitar que entre en la base de conocimiento cualquier tipo de conocimiento inconsistente o contradictorio.

El control de la coherencia debe hacerse controlando la coherencia de las reglas y la de los hechos.



### **1.1.7.3 Valoración de las reglas de Producción como Representación del Conocimiento**

Se presentan aquí algunos criterios para evaluar las reglas de producción como representación del conocimiento.

#### **Desempeño**

Depende de la cantidad y profundidad de su conocimiento. La modularidad de las reglas es fundamental para un buen desempeño, entre otras cosas para añadir nuevas reglas, detectar inconsistencias, etc.

#### **Explicación**

La forma de explicación debe de ser muy amigable para los expertos, esta es una de las razones por las cuales usar procesamiento simbólico y reglas.

Se debe facilitar:

- Mostrar en cualquier momento la regla considerada
- Almacenar las reglas utilizadas para propósitos de explicación
- Encontrar reglas específicas para contestar algún tipo de pregunta

#### **Adquisición**

Se debe contar con un modulo de adquisición de conocimiento muy flexible que permita fácilmente introducir y modificar las reglas. Las explicaciones permiten encontrar fallas en la base de conocimiento y facilitan su modificación.

La adquisición debe estar libre de contradicciones, redundancias y problemas con contradicciones indirectas, efectos secundarios (por ejemplo, actualización de información relacionada) .

## CAPÍTULO 2

### LOS SISTEMAS EXPERTOS COMO UN APOORTE A LA SOLUCIÓN DE LA PROBLEMÁTICA DEL TRATAMIENTO DE LA TUBERCULOSIS

#### 2.1 LA TUBERCULOSIS: UN PROBLEMA DE SALUD PÚBLICA MUNDIAL.

La Tuberculosis es una enfermedad infectocontagiosa bacteriana producida por una micobacteria del complejo Mycobacterium: M. Tuberculosis, M. bovis o M. africanum. (Bacilo de Koch), con diversas manifestaciones clínicas y con amplia distribución mundial [1].

En todo el mundo, más personas mueren de tuberculosis que de otra enfermedad infecciosa curable. Todos los días, más de 20.000 personas contraen la tuberculosis activa, y 5.000 mueren a causa de ello [7]. Un tercio de la población mundial está infectada por el bacilo de la tuberculosis. La tuberculosis es causada por el bacilo de Koch, una bacteria transportada por el aire de persona a persona, como la del resfriado común. Si no se trata, una persona con tuberculosis activa puede infectar en promedio 10 a 15 personas por año. La tuberculosis ataca a las personas en todo el mundo, independientemente de su raza o poder económico.

A pesar de los enormes esfuerzos realizados por los gobiernos del mundo, la tuberculosis (TB) persiste como un problema mundial de salud pública de grave magnitud, que requiere atención urgente. La cuadragésima cuarta Asamblea Mundial de la Salud (1991) estableció las metas mundiales para el control de la TB para el año 2000. Estas eran curar el 85% de los casos de TB infecciosa y detectar el 70% de estos casos. En vista del lento progreso en muchos países, la fecha fue revisada y proyectada al año 2005. [8].

Aliviar la enfermedad, el sufrimiento y la muerte de los individuos infectados por la TB, es la principal inquietud humanitaria y requiere un enfoque centrado en el paciente para el control de la TB. La dimensión de salud pública incluye el diagnóstico y tratamiento adecuado de los pacientes con TB para reducir la transmisión de la enfermedad dentro de las comunidades. Esto requiere la elaboración de programas de control de la TB bien organizados.

La cuadragésima cuarta Asamblea Mundial de la Salud (1991) reconoció la importancia creciente de la TB como un problema de salud pública y el potencial para el control eficaz en

función de los costos, empleando las herramientas actualmente disponibles [8]. Esto condujo a una revaloración de los esfuerzos progresivos de control de la TB. La persistencia de la TB se ha debido principalmente al descuido del control de la enfermedad por los gobiernos, los programas de control de la TB mal administrados, la pobreza, el crecimiento de la población y la migración. Para ayudar a abordar la situación, se elaboró entonces un marco nuevo para el control eficaz de la TB [8] y se introdujo una estrategia mundial llamada DOTS. Los cinco elementos de la estrategia de DOTS, considerada como esencial para el control mundial de la TB son: el compromiso político, la detección de casos usando la microscopía de esputo entre las personas que buscan la atención por tiempos prolongada, la quimioterapia de corta duración estandarizada en condiciones adecuadas del manejo del caso incluido el tratamiento directamente observado, suministro regular de medicamentos y un sistema de registro y notificación estandarizado, que permita la evaluación individual de los pacientes, así como el rendimiento general del programa [8].

Varios retos impiden la ejecución sostenible y la expansión de las actividades de control de la TB. Muchos de estos retos provienen de una débil voluntad política para mejorar el sistema de salud y la respuesta social necesaria para el control de la TB. Los servicios generales de salud pública necesitan mejorar su capacidad de mantener y ampliar la ejecución de DOTS sin comprometer la calidad de la detección de casos y el tratamiento. La participación comunitaria en la atención de TB y la necesidad de enfatizar el enfoque de la promoción en el paciente son necesarias para mejorar tanto el acceso como la utilización de los servicios de salud. La colaboración entre los sectores públicos, privados y voluntarios es esencial para asegurar accesibilidad al diagnóstico y tratamiento de TB con garantía de calidad.

### **2.1.1 Estrategias y Operaciones claves para el control de la Tuberculosis**

Las metas del control de la TB son reducir la mortalidad, la morbilidad y la transmisión de la enfermedad, así como la prevención de la fármaco resistencia, logrando que esta, no plantee una amenaza para la salud pública. También procura reducir el sufrimiento humano, la carga económica y social que la familia y las comunidades tienen que soportar en consecuencia. Para lograr esto, es necesario garantizar el acceso al diagnóstico, tratamiento y la curación para cada paciente con TB, y proteger a las poblaciones vulnerables a la TB y sus formas fármaco resistente. [8].

Dentro de la estrategia mundial para el control de la tuberculosis, se han adoptado algunos elementos clave entre los que se pueden destacar:

- Lograr un firme compromiso político sostenido de aumentar los recursos humanos y financieros. El programa de DOTS debe ser una actividad integral de salud con cobertura para toda la nación que incluya las actividades de TB en todo el sistema de salud, a todos los niveles, incluidos los establecimientos de salud periféricos y la comunidad.
- Las guías prácticas estandarizadas para los que prestan atención de las enfermedades respiratorias, pueden mejorar la detección de casos entre los pacientes con síntomas respiratorios, e incrementar el diagnóstico de TB y fortalecer las actividades de control de la TB por los prestadores de asistencia sanitaria.
- Asegurar la administración de un tratamiento adecuado a los pacientes, así como el suministro ininterrumpido de los medicamentos. El objetivo fundamental de cualquier programa antituberculoso que espera lograr el control eficaz lo constituye un suministro ininterrumpido y sostenido de los medicamentos antituberculosos, con garantía de calidad.
- Desarrollar sistemas de registro y notificación que permitan la evaluación del resultado de cada paciente y evaluación del desempeño general de los programas. Tal sistema debe convertirse en una herramienta útil no solo para vigilar el progreso y resultados de tratamiento en cada paciente, sino también para evaluar el rendimiento general de los programas. Estos sistemas deben usar mecanismos estandarizados, tales como informes trimestrales y análisis de cohorte que permitan evaluar los aspectos esenciales del programa de control.
- Los programas de investigaciones operacionales deben ser un componente integral de la ejecución de DOTS. El diseño y la realización local de las investigaciones operacionales pertinentes pueden ayudar a identificar los problemas y las razones del débil rendimiento y encontrar las soluciones prácticas.

### **2.1.2 Problemática de la Tuberculosis en Colombia**

En la Región de las Américas, aunque muchos países están utilizando la estrategia de DOTS, la mayoría de ellos lo hacen a escala limitada [7]. La expansión de DOTS es crucial para controlar la tuberculosis en la Región, y sobre todo para llegar a las metas mundiales de tuberculosis para 2005. También quieren aumentar la conciencia entre los líderes políticos, los responsables de la

toma de decisiones y de opinión acerca de las implicaciones de la tuberculosis. Otras prioridades son forjar las alianzas estratégicas con socios que puedan apoyar los programas de control, especialmente en los países más pobres, donde la tuberculosis es responsable del mayor número de víctimas y defunciones, y a movilizar a los pacientes de tuberculosis para que exijan mayor acceso a diagnóstico, tratamiento y rehabilitación física y social; de esa manera, pueden hacerse socios productivos de la sociedad.

En Colombia, la situación de la tuberculosis no es alentadora. Según información del Ministerio de Salud (hoy ministerio de Protección Social), la tasa de incidencia venía en descenso desde 1970, cuando se reportaron 58.6 casos por 100.000 habitantes. En 1997 se describió una incidencia de 21.5 y un discreto incremento en el año 2000 a 27.4 casos por 100.000 habitantes. El Ministerio de Salud consideró que el descenso de la tasa de incidencia se debió a la disminución de la búsqueda de casos y al subregistro de los enfermos diagnosticados.

En el Municipio de Popayán, la tasa de incidencia venía en descenso desde 1990 hasta 1996, cuando se reportó 9.9 casos por 100.000 habitantes, fenómeno atribuido a las mismas razones del orden nacional. En el año 2000, y gracias a una campaña de mayor búsqueda de pacientes, la tasa de incidencia se incrementó en 330%, alcanzando 32 casos por 100.000 habitantes. Los diferentes factores para el desarrollo de la enfermedad se combinan en nuestra región de manera preocupante: altas tasas de desempleo, pobreza extrema, desnutrición, hacinamiento, población desplazada en crecimiento, servicios médicos insuficientes y últimamente la aparición de cepas bacterianas resistentes a los antibióticos.

La Organización Mundial de la Salud y la Organización Panamericana de la Salud han concluido que el divorcio entre las Universidades y los Programas de Control de Tuberculosis gubernamentales tienen un impacto negativo importante sobre los indicadores de gestión en la lucha contra la enfermedad [9]. El porcentaje de captación de sintomáticos respiratorios en Colombia es de 36.7%, siendo la meta propuesta de 80%, con una alta proporción de abandonos terapéuticos (12.2 %) y mortalidad (7.5%). Los errores terapéuticos cometidos por los profesionales de la salud en el manejo de la enfermedad son ahora responsables del gran incremento de casos de tuberculosis resistente a múltiples drogas.

El Ministerio de Salud expidió la norma nacional para el control de la tuberculosis mediante la Resolución 0412 del 25 de febrero de 2000, en la cual se establecen las actividades, intervenciones y procedimientos de obligatorio cumplimiento por todos los profesionales de la salud, las administradoras y las instituciones prestadoras de servicios en salud del territorio

nacional, y con la cual se pretende unificar las acciones diagnósticas y los esquemas terapéuticos, evitando así la catástrofe de la aparición de tuberculosis resistente a medicamentos y la alta morbimortalidad asociado a esto [10].

## **2.2 POR QUE USAR SISTEMAS EXPERTOS**

En vista de la problemática planteada, se vislumbra la necesidad de que los diferentes sectores sociales tomen parte en su solución. Las universidades hacen parte de uno de los sectores que más efectivamente podrían contribuir a realizar propuestas tecnológicas que aporten soluciones ajustadas al cumplimiento de la metas, estrategias y operaciones claves establecidas por la OMS.

Dentro de las actividades sugeridas por la OMS, están aquellas que se centran en brindar una atención idónea al paciente, asegurando la administración de tratamientos adecuados y el estricto suministro de medicamentos. Con la Guía de Atención de la Tuberculosis Pulmonar y Extrapulmonar se intenta cumplir con esta recomendación; sin embargo, no es suficiente con tener una guía oficial, si no se garantiza su difusión y correcta aplicación en el territorio nacional.

La Guía de Atención de la Tuberculosis Pulmonar y Extrapulmonar es el conjunto de actividades, intervenciones y procedimientos técnico-científicos y administrativos, a través de los cuales se busca garantizar la atención integral del usuario, determinando los pasos mínimos a seguir y el orden secuencial y lógico de éstos, que aseguren la oportunidad de la atención, la eficiencia y calidad de los servicios y su impacto, así como la satisfacción del usuario. [1].

Esta guía, es el resultado de años de experiencia mundial en el tratamiento de la tuberculosis; maneja diferentes esquemas de tratamiento según las características del paciente y la presencia de situaciones especiales. Sin embargo, existen casos que no se hacen explícitos en la norma y cuya solución depende del criterio médico especializado. En algunos de estos casos se ve comprometida la calidad de los tratamientos administrados cuando el profesional de la salud no posee la experiencia suficiente para tomar decisiones adecuadas en aspectos tales como medicamentos que se debe prescribir, dosis de los medicamentos, duración de las fases del tratamiento, entre otros.

Mediante el uso de tecnologías informáticas, se pueden desarrollar aplicaciones que garanticen la adecuada administración de tratamientos. Los sistemas expertos tienen amplia aplicación en

la solución de problemas de este tipo. Un sistema experto, es capaz de proponer soluciones a problemas de toma de decisiones teniendo como base información obtenida desde fuentes como libros y revistas, además de la información adquirida de expertos en el dominio de la aplicación, resultado de años de práctica en el campo. Teniendo en cuenta esta característica podemos decir que un sistema experto es una solución apropiada a la necesidad planteada, ya que este puede combinar la información extraída de la guía con la información de médicos especialistas para sugerir tratamientos convenientes según las características del paciente.

Además de las características mencionadas arriba, es necesario que el sistema que se desarrolle sea lo suficientemente flexible como para soportar actualizaciones continuas al conocimiento del dominio, debido tanto a los adelantos en investigación en el tratamiento de la enfermedad, así como a las nuevas experiencias adquiridas por los expertos encargados de alimentar al sistema. De igual manera, es necesario que los resultados arrojados por el sistema, sean muy fiables. Estas características son soportadas con amplitud por los sistemas expertos, que tienen la habilidad de permitir una fácil actualización de su base del conocimiento, y además proveen mecanismos de explicación del razonamiento empleado para llegar a las conclusiones finales con lo que el usuario puede estar seguro de que los resultados obtenidos tienen un alto grado de certeza.

Otros criterios nos permiten inclinarnos por los sistemas expertos como el paradigma adecuado en el que debe estar sustentado el sistema:

- **Delimitación del Dominio:** Para la construcción de un sistema experto eficiente es necesario que el dominio de aplicación este bien delimitado, con el objetivo de hacer al sistema simple y no muy cargado de conocimiento de diferentes dominios. La combinación de dominios hace compleja la coordinación de los mismos, así como el rendimiento y mantenimiento del sistema. En nuestro caso, el dominio está restringido al tratamiento de una enfermedad lo que hace positiva la evaluación de este criterio.
- **Necesidad y el deseo de tener un sistema experto:** En el medio existe la necesidad de tener un sistema experto para el control de la tuberculosis, debido a la escasez de especialistas humanos que cubran las necesidades en todo el territorio nacional, además existe el interés de organismos tales como la Dirección Departamental de Salud del Cauca.
- **Presencia de al menos un especialista dispuesto a cooperar:** Este es quizás el criterio más importante a tener en cuenta, ya que sin una fuente importante de experiencia no se

podría llegar a la obtención de una solución de calidad. Afortunadamente, en el desarrollo de este proyecto se contó con la ayuda de médicos y profesionales en enfermería que hicieron valiosos aportes para construir la base de conocimiento del sistema.

- **Parte del conocimiento es heurístico ó empírico.** Es importante que un porcentaje significativo del conocimiento provenga de la experiencia de un especialista. Para nuestro sistema, aunque existía una fuente base para extraer el conocimiento la cual era la guía para la atención de la tuberculosis gran parte del conocimiento fue aportado por médicos y profesionales de enfermería con gran experiencia en el campo.

Conscientes que en Colombia no se han desarrollado aplicaciones cuyo objetivo sea dar solución al problema de control de la tuberculosis se construyó una herramienta software (SIEXCO-TB), la cual consiste en un sistema experto que tiene como información de soporte la guía de atención de la tuberculosis pulmonar y extrapulmonar, la experiencia de especialistas en el área y la información particular del paciente, con lo cual estará en capacidad de sugerir un tratamiento de acuerdo a un esquema determinado por dicha norma.

Adicionalmente el sistema provee información estadística acerca de indicadores de efectos colaterales originados por el consumo de medicamentos, los cuales contribuirán a prevenir trastornos futuros en los pacientes.

Para el desarrollo del proyecto se contó con la participación de dos expertos, perteneciente al Grupo de Investigación en Tuberculosis, quienes son el Dr. Alfonso Tenorio y la Dra. Regina Plaza. También contribuyó en la ejecución del proyecto la enfermera Helena Quintero perteneciente a dicho grupo.



## CAPITULO 3

### EXPLORACIÓN TECNOLÓGICA

#### 3.1 EXPLORACIÓN TECNOLÓGICA

En la fase de exploración tecnológica para el desarrollo de SIEXCO-TB se encontraron diferentes herramientas para la representación de conocimiento en sistemas expertos, entre las ellas las más difundidas y utilizadas para la construcción de aplicaciones son CLIPS [12], Jess [13], Jeops [14], Mandarax [15] y JRules [16].

A continuación se presentan los resultados de este estudio, caracterizando cada una de estas herramientas, enunciando sus ventajas y desventajas, y mostrando el cuadro comparativo que nos permitió elegir una de ellas para la construcción de SIEXCO-TB.

##### 3.1.1 CLIPS (C Language Integrated System)

CLIPS es una herramienta para el desarrollo de Sistemas Expertos que ofrece un entorno completo para la construcción de Sistemas Expertos basados en reglas y objetos. CLIPS está siendo utilizado por numerosos usuarios entre los que se encuentran la NASA (que son sus creadores), numerosas universidades y empresas. CLIPS viene de (C Language Integrated Production System) y como su nombre lo indica, uno de los objetivos que buscaban sus creadores era la fácil integración con programas en C.

###### 3.1.1.1 Características

**Motor de Inferencia:** Utiliza el algoritmo RETE.

**Entornos de desarrollo:** Tiene su propio lenguaje y entorno de programación. Sin embargo también puede ser utilizado con lenguajes de programación convencionales.

**Licencia:** CLIPS Es de libre distribución.

**Orientación a objetos:** SI.

**Documentación:** Cuenta con manuales y tutoriales muy completos que facilitan la construcción de sistemas con esta herramienta.

### 3.1.1.2 Ventajas

**Representación de conocimiento heurístico:** Las reglas se usan principalmente para la representación de conocimiento heurístico (basado en la experiencia), funciones específicas y funciones genéricas, para la representación de conocimiento procedimental.

**Uso con otro Lenguajes:** Puede ser usado en conjunto con lenguajes de programación convencionales como C, Ada y C++.

**Lenguaje de programación Propio:** Clips define su propio lenguaje de programación (CLIPS Object-Oriented Language).

**Estructura de datos y Clases:** CLIPS hace una diferencia entre tipos de estructuras de datos y clases de forma similar a la diferencia entre struct y class en c++.

**Licencia:** CLIPS es de libre Distribución.

### 3.1.1.3 Desventajas

**Falta de Soporte:** CLIPS raramente ha sido usado para fines comerciales principalmente debido a la falta de soporte.

**No orientado a Objetos:** No es un sistema completamente orientado a objetos.

**Reglas en XML:** CLIPS no soporta documentos tipo XML para la representación del conocimiento.

**Integración con aplicaciones:** Una aplicación realizada en COOL (Lenguaje de programación propio de CLIPS), no es fácil de integrar con otra aplicación realizada en un lenguaje de programación convencional que soporte CLIPS.

### 3.1.2 JESS (Java Expert System Shell)

Inicialmente fue una conversión de un subconjunto de CLIPS para java, manteniendo su misma filosofía. Posteriormente, se agregaron otras características propias para resolver algunos de los problemas de clips.

Jess utiliza objetos JAVA. Sin embargo existe una restricción para los objetos java que pueden ser utilizados. Como no existe ninguna forma explícita de avisar a un sistema cuando un objeto es modificado, los objetos deben ser capaces de avisar cuando una de sus propiedades es alterada. Así, Jess interactúa con JAVA a través de JavaBeans.

#### 3.1.2.1 Características

**Motor de Inferencia:** El motor de Inferencia de Jess usa el algoritmo RETE y puede utilizar encadenamiento hacia atrás y hacia adelante.

**Entornos de Desarrollo:** Posee 3 interfaces de programación. Mediante línea de comandos, consola o applets. Jess provee una interfaz de línea de comandos a través de la clase “jess.Main”. Si se usa “jess.Console” se tiene un interfaz en modo gráfico (consola).

**Licencia.** Para ser utilizado en desarrollos comerciales el valor de la licencia oscila entre los 40.000 y 75.000 dólares. Se permite el uso libre en ámbitos educativos, sin fines comerciales.

**Orientación a Objetos:** Se presenta la separación entre dos mundos: El de objetos Java y el de objetos Jess. La sintaxis utilizada para definir las reglas no se parece en nada a la utilizada en el lenguaje java, haciendo que el programador java tenga que pasar un tiempo mayor de entendimiento del sistema.

**Documentación:** Existe documentación completa, además debido a su popularidad, existen muchos ejemplos de aplicaciones desarrolladas con esta herramienta.

### 3.1.2.2 Ventajas

**Trabajo con CLIPS.** Los objetos Java pueden ser accedidos desde las reglas CLIPS

**Estandarización.** Los desarrolladores de Jess hacen parte del comité que actualmente sigue trabajando en la especificación del API para la máquina de reglas de java.

**Popularidad.** Al tomar como base para su desarrollo a CLIPS, se ha convertido en una herramienta muy popular y su funcionamiento ha sido comprobado mediante el desarrollo de diferentes aplicaciones.

**Lógica difusa.** Incluye lógica difusa en la versión 5.0 (FuzzyJess)

**Extensibilidad.** Se pueden añadir nuevas funciones a Jess programadas en Java (Userfunction). También se puede añadir paquetes de funciones (Userpackage)

### 3.1.2.3 Desventajas

**Licencia.** El principal problema de Jess radica en que para poder utilizarlo en desarrollos comerciales su licencia no es libre.

**Inferencia sobre objetos.** El motor de inferencia de JESS no puede interactuar directamente con los objetos Java así que es necesario construir una interfaz entre los objetos JESS y los objetos JAVA. Esta característica hace el desarrollo de aplicaciones algo complejo.

**Complejidad.** El desarrollo de un sistema sobre Jess es complejo e incrementa el tiempo de ejecución de un proyecto (en dos, tres meses) cuando no se tiene experiencia previa en su estructura y formas de representación. Uno de los principales problemas a los que se ve enfrentado el desarrollador que esta acostumbrado a trabajar con orientación a objetos es la pérdida de control sobre el proceso de ejecución del programa. Esto lleva a que se requiera cierta pericia en el modelado de las reglas.

## **Dificultad de Integración de sistemas desarrollados en Jess con otros sistemas**

La necesidad de que los objetos deban ser capaces de avisar sobre las modificaciones ocurridas en sus propiedades (Java Beans), limita la utilización del sistema que se ve imposibilitado por ejemplo a utilizar algunas clases definidas por terceros (Aquellas que no notifican un cambio en sus propiedades).

**Reglas en XML:** Si soporta documentos tipo XML para la representación del conocimiento.

### **3.1.3 JEOPS**

La idea del sistema surgió durante la elección de proyectos para un trabajo de graduación, Inteligencia Artificial Simbólica, que consistía en crear un sistema en el cual primase la facilidad de uso y el paradigma de orientación a objetos. Esto dio como resultado el desarrollo de la primera versión de JEOPS. Esta versión presentaba serios problemas de desempeño (el algoritmo de unificación era bastante ineficiente), soportaba un subconjunto muy limitado de la funcionalidad de java (No permitía llamadas encadenadas a métodos, problemas con las variables de tipo String entre otras). Todo esto impedía utilizarlo en aplicaciones más complejas.

Terminado el trabajo de grado el autor se interesó por el proyecto y continuó su desarrollo por su propia cuenta. Luego se continuó perfeccionando con el desarrollo de nuevos trabajos de grado, trabajos de maestría y aportes hechos por diferentes personas debido a que el software es de libre distribución. Actualmente JEOPS ha sido usado en diferentes proyectos en áreas como: Fonética, Administración de redes, Juegos y recuperación de información. El sistema ya es lo suficientemente maduro como para ser utilizado en el desarrollo de aplicaciones más robustas.

#### **3.1.3.1 Características**

**Motor de inferencia:** Utiliza el algoritmo RETE.

**Entornos de desarrollo:** No tiene un entorno de desarrollo propio.

**Licencia:** es de libre distribución (Licencia LGPL).

**Orientación a objetos:** Es completamente orientado a objetos

**Documentación:** Existe documentación pero no es muy extensa.

### 3.1.3.2 Ventajas

**Lenguaje de desarrollo:** Esta desarrollado en java y proporciona toda su funcionalidad. JEOPS mantiene todas las ventajas de la programación orientada a objetos.

**Licencia:** JEOPS es un software de libre distribución.

**Sencillez:** Uno de los principales objetivos de JEOPS en el momento de su desarrollo fue crear una herramienta fácil de utilizar en la cual el usuario tenga a disposición todos los servicios ofrecidos por el lenguaje JAVA.

**Sintaxis de las Reglas:** La sintaxis para la definición de reglas es similar al lenguaje de programación java, esto disminuye el tiempo necesario para el aprendizaje del manejo de la herramienta.

**Madurez:** JEOPS ha sido usado en diferentes proyectos en áreas como: Fonética, Administración de redes, Juegos y recuperación de información. El sistema ya es lo suficientemente maduro como para ser utilizado en el desarrollo de aplicaciones más robustas.

### 3.1.3.3 Desventajas

**Reglas en XML:** No facilita la utilización de archivos de texto XML para la implementación de las reglas.

**Entorno de desarrollo:** No tiene un entorno de desarrollo que facilite al usuario la implementación de las aplicaciones.

**Popularidad:** El uso de JEOPS ha disminuido respecto a la utilización de herramientas como Jess y Mandarax

### 3.1.4 Mandarax

Es una librería JAVA de fuente abierta para deducción de reglas. Es puramente orientado a objetos. Utiliza razonamiento hacia atrás al estilo de prolog. Mandarax soporta contenedores flexibles de estructuras de datos llamados clause sets para la representación del conocimiento. Los clause sets pueden ser utilizados para integrar, en la base de conocimiento, los resultados de consultas de otras bases de conocimiento.

Mandarax integra fácilmente una variedad de fuentes de datos, por ejemplo, los registros de una base de datos pueden ser integrados como conjuntos de hechos. Otras fuentes de datos como EJBs, datos retornados por web services, etc, también pueden ser integrados.

Para la persistencia de las reglas, Mandarax usa el módulo XKB, responsable de una serialización de objetos de propósito general, utilizando XML. Una interfase XML alternativa es el Framework ZKB.

#### 3.1.4.1 Características

**Motor de inferencia:** a diferencia de otras herramientas como JESS este no utiliza el algoritmo RETE. Trabaja con encadenamiento hacia atrás.

**Entornos de desarrollo:** Posee una interfaz gráfica para facilitar el desarrollo de aplicaciones y provee soporte para trabajar con XML.

**Orientación a objetos:** Orientado a objetos.

**Documentación:** Existe documentación, aunque no muy amplia

**Licencia:** Software de libre distribución (Licencia GPL)

#### 3.1.4.2 Ventajas

**Reglas:** Provee el soporte necesario para representar las reglas en lenguaje natural a través de los *blueprints*.

**Lenguaje de desarrollo:** Está desarrollado en lenguaje java y por lo tanto provee todas sus funcionalidades y ventajas.

**Orientado a Objetos:** Mandarax es orientado a objetos totalmente, no es una traducción de un interprete Prolog c a java. Su diseño es flexible y abierto.

**Fuentes de datos:** Es fácilmente integrable con todos los tipos de fuentes de datos como web services.

**Persistencia de las Reglas:** La persistencia de la base de reglas se puede obtener con la utilización del módulo XKB. Este módulo almacena las reglas y el conocimiento en un formato similar a RuleML el cual trabaja con el estándar XML. De esta forma se tiene todas las ventajas que proporciona el trabajar con XML.

Las últimas versiones de Mandarax presentan un amplio desarrollo en una interfase XML alternativa: el Framework ZKB. ZKB usa XML para representar bases del conocimiento, hechos, reglas y términos; delegando la serialización de referencias de objetos a un objeto separado llamado Object Persistence Service (OPS). El propósito de OPS es generar nombres de objetos, la interfase es similar al contexto JNDI.

**Reglas en RuleML:** Soporta la importación y exportación de reglas en formato RuleML, el cual es un estándar mundialmente utilizado en la representación de reglas.

### 3.1.4.3 Desventajas

**Documentación:** No se encuentra mucha documentación con relación al manejo de esta herramienta.

**Complejidad:** La sintaxis que se debe utilizar para la codificación es un poco compleja, por lo tanto no es muy fácil de aprender.



### 3.1.5 ILOG JRules

JRules es una herramienta para la construcción de sistemas expertos basados en reglas en Java. Incluye las herramientas necesarias para modelar, escribir, probar y mantener reglas. Provee un repositorio para organizar y almacenar reglas, y un motor de inferencia para ejecutarlas.

ILOG JRules permite a los desarrolladores de aplicaciones combinar programación basada en reglas y programación orientada a objetos. Permite un rápido desarrollo y mantenimiento de aplicaciones sin sacrificar propiedades como la escalabilidad e integración.

#### 3.1.5.1 Características

**Motor de Inferencia:** JRules usa un algoritmo Rete Optimizado. El algoritmo Rete es usado gracias a su capacidad de manejar un gran número de reglas y a su eficiencia.

**Entorno de Desarrollo:** Provee una interfaz gráfica para la creación y mantenimiento de bases de conocimiento.

**Licencia:** No es de libre distribución. La licencia más económica es la ofrecida para fines académicos, la cual cuesta alrededor de US \$1000.

**Orientación a Objetos:** Es totalmente orientado a objetos.

**Documentación:** Ampliamente documentado.

#### 3.1.5.2 Ventajas

**Representación de Reglas con XML:** Se pueden definir las reglas usando XML.

**Entorno de desarrollo:** Provee un entorno de desarrollo propio.

**Integración con otras aplicaciones:** Fácil integración con aplicaciones en Java.

**Reglas:** Provee el soporte necesario para representar las reglas en lenguaje natural y luego ser ejecutadas por el motor de inferencia.

**Trabajo con J2EE:** Provee las clases necesarias para hacer el manejo de las reglas en entornos J2EE.

### 3.1.5.3 Desventajas

**Licencia.** No es de libre distribución

## 3.2. CUADRO COMPARATIVO DE LAS TECNOLOGÍAS

Tecnología	Algoritmo del Motor Inferencia	Entorno Desarrollo Propio	Libre Distribución	Orientado Objetos	Calidad de Documentación	Lenguaje natural para Reglas	Reglas en XML	Variedad de Fuentes de Datos
<i>CLIPS</i>	RETE	SI	SI	SI	Alta	NO	NO	Baja
<i>Jess</i>	RETE	SI	NO	SI	Alta	NO	SI	Media
<i>Jeops</i>	RETE	NO	SI	SI	Baja	NO	NO	Baja
<i>Mandarax</i>	Otro	SI	SI	SI	Media	SI	SI	Amplia
<i>JRules</i>	RETE Mejorado	SI	NO	SI	Alta	SI	SI	Amplia

**Tabla 3 Comparación Tecnologías**

En la tabla 3 se muestra un cuadro comparativo de las 5 tecnologías estudiadas para la construcción de sistemas expertos basados en reglas. Se puede observar que existen tres de ellas que se adaptan a las necesidades del proyecto por las características y ventajas sobre las demás.

Estas tres tecnologías son *Jess*, *Mandarax* y *JRules*.

Las tres tienen un entorno para la creación y mantenimiento de reglas propio a diferencia de *Jeops* que no lo tiene. Esta es una característica importante a la hora de construir una base de conocimiento flexible, fácilmente actualizable por el ingeniero de conocimiento.

Los tres son completamente orientados a objetos, aunque en *Jess* la sintaxis utilizada para definir las reglas es diferente a la utilizada en el lenguaje java lo que dificulta el aprendizaje de los programadores.

*Jess* y *JRules* tienen una amplia documentación. *Mandarax* tiene documentación disponible aunque no tan amplia como *Jess* y *JRules*.

*Mandarax* y *JRules* soportan la representación de reglas en lenguaje natural, esto facilita la comprensión de la base del conocimiento del usuario del sistema encargado de mantenerla.

*Mandarax*, *Jess* y *JRules* proporcionan los mecanismos necesarios para la representación de las reglas usando XML. Esto nos aporta todas las ventajas que trae consigo la representación de la información del sistema experto en un lenguaje estándar y mundialmente utilizado.

Tanto *Mandarax* como *JRules* integran una gran variedad de fuentes de datos, tales como archivos XML, bases de datos, EJBs, lo que nos da variedad de opciones para integrar conocimiento desde diferentes contextos de aplicación.

Hasta aquí se ha visto como *Mandarax*, *Jess* y *JRules* se destacan como las tres tecnologías de mayor viabilidad para el desarrollo de SIEXCO –TB, sin embargo existe una diferencia fundamental entre *Mandarax* y las otras dos, es que *Mandarax* es una herramienta de libre distribución, mientras que *Jess* y *JRules* ofrecen licencias comerciales. Esto sumado a las demás ventajas antes mostradas, hacen de *Mandarax* una herramienta con prestaciones comparables a las de *Jess* y *JRules* que dan las bases suficientes para decir que la mejor opción tecnológica para el manejo del modulo experto de SIEXCO – TB es *Mandarax*.

### **3.3. SISTEMAS EXPERTOS DE APOYO EN LA TOMA DE DECISIONES CLÍNICAS.**

El análisis de decisión está siendo gradualmente reconocido como una técnica poderosa para seleccionar las estrategias óptimas en problemas de decisión clínica. Muchos esfuerzos en el área de los sistemas de decisión basados en conocimiento pretenden automatizar el proceso de análisis de decisión usando técnicas de inteligencia artificial.

Los sistemas expertos de apoyo en la toma de decisiones clínicas representan una aproximación importante a la solución de problemas en el área de la medicina, tales como el diagnóstico, tratamiento y control de enfermedades. Se presentan aquí algunas aplicaciones en esta área, mostrando algunas de sus características y resaltando sus aportes para la construcción de SIEXCO-TB.

### 3.3.1 MYCIN

#### Descripción funcional

Su función es la de aconsejar a los médicos en la investigación y determinación de diagnósticos en el campo de las enfermedades infecciosas de la sangre [17,22,23]. Mycin permite:

- A partir de la información sobre los síntomas del paciente y los resultados de los tests, trata de identificar el organismo responsable de la infección.
- Durante la consulta, Mycin puede requerir más información, formulando preguntas. El programa puede también sugerir que se hagan nuevos tests.
- Recomienda los medicamentos que el paciente debe consumir de acuerdo a la infección diagnosticada.
- Explica el razonamiento que le lleva a su diagnóstico.

#### Entradas del sistema

- Datos generales del paciente (nombre, edad, síntomas, etc.).
- Datos particulares del paciente que el sistema pide por medio de preguntas.
- Datos relacionados con lo exámenes de laboratorio.

#### Representación del conocimiento

- La base de conocimientos de MYCIN está formada por reglas de producción, en las que la premisa está compuesta por una conjunción de proposiciones correspondientes a síntomas o a hipótesis auxiliares ya establecidas, y las conclusiones hacen referencia a las enfermedades o a las hipótesis intermedias que se establecen en el diagnóstico.
- Emplea un mecanismo de inferencia con encadenamiento hacia atrás

### **Criterios de diseño.**

A continuación se mencionan algunos de los criterios de diseño más importantes tenidos en cuenta en el desarrollo de MYCIN.

- Utiliza la técnica de factores de certeza. Esto es, MYCIN incorpora unos coeficientes de certeza aplicables a las reglas y a las conclusiones que expresan mediante números racionales del intervalo  $[-1,1]$ , la confianza que merece al experto el conocimiento expresado por las reglas o por su conclusión. Estos coeficientes de certeza se modifican para las distintas hipótesis de diagnóstico que se establecen durante el proceso, acumulando evidencia en algunas y rechazándola en otras según las respuestas del usuario. En los casos en los que varias reglas pueden ser seleccionadas a la vista de un subobjetivo, los factores de confianza, y el orden en que aparecen escritas las reglas, permiten obtener la secuencia en la que se exploran las distintas alternativas.
- Mycin fue uno de los primeros sistemas expertos que separó el conocimiento del dominio y el mecanismo de inferencia.

### **Comparación con SIEXCO-TB**

SIEXCO-TB cuenta con las siguientes características comunes a Mycin.

- Para representar el conocimiento SIEXCO-TB utiliza reglas de producción.
- Emplea un mecanismo de inferencia con encadenamiento hacia atrás.
- SIEXCO-TB cuenta con un módulo encargado de explicar el razonamiento que le permitió llegar a la conclusión.
- Proporciona los medicamentos que el paciente debe consumir de acuerdo a sus características particulares.
- Separa explícitamente la base de conocimiento del mecanismo de inferencia.

### 3.3.2. PRODIGY

#### Descripción funcional

*Prodigy* [18,21] está destinado a ofrecer una base de conocimientos médicos de alta calidad en un entorno informatizado, y concretamente en los diferentes lugares de trabajo y con los diferentes programas que los médicos generales utilizan en el proceso de atención. Desde sus inicios han intervenido un gran número de usuarios finales, médicos generales cuyas opiniones y sugerencias son continuamente incorporadas al proceso de desarrollo del programa.

Básicamente, consiste en un sistema que ofrece una serie de recomendaciones actualizadas y basadas en la evidencia sobre el manejo de un determinado proceso o enfermedad. A continuación se mencionan algunas funcionalidades relevantes del sistema.

- Realiza el diagnóstico.
- Brinda información acerca de las opciones recomendadas de acuerdo al diagnóstico.
- Genera la prescripción de los medicamentos.
- Brinda consejos no farmacológicos.
- Este sistema y la información que contiene se acompaña de aplicaciones informáticas que le permiten integrarse en los diferentes programas de historia y gestión clínica.

#### Entradas del sistema

- Información relacionada con los datos del paciente, como síntomas, signos, datos personales y resultados de exámenes de laboratorio.

#### Representación del conocimiento

- Reglas de producción.
- Hace uso de operadores.

- Mecanismo de inferencia con encadenamiento hacia atrás.

### **Criterios de diseño**

- PRODIGY utiliza planes de simulación los cuales se generan mediante objetos, estos planes de simulación se generan inicialmente vacíos y posteriormente se modifican hasta que se obtiene un plan correcto.

### **Comparación con SIEXCO-TB**

A continuación se mencionan las características similares de SIEXCO-TB con PRODIGY.

- SIEXCO-TB al igual que PRODIGY permite la integración con otro tipo de sistema utilizado ampliamente en medicina como es Epi\_Info.
- SIEXCO-TB utiliza una metodología similar a la de los planes de simulación utilizada en PRODIGY. SIEXCO-TB utiliza esta metodología para manejar el perfil del paciente y el tratamiento a suministrar, es decir se genera inicialmente una plantilla vacía la cual se llena con los datos del paciente (hechos). Luego con los datos (hechos) más relevantes se genera un tratamiento inicial, este es modificado de acuerdo a los datos (hechos) más particulares hasta generar un tratamiento adecuado para el paciente.
- SIEXCO-TB proporciona información relacionada con la prescripción de medicamentos y algunas recomendaciones para pacientes afectados por tuberculosis.
- SIEXCO-TB utiliza encadenamiento hacia atrás.
- El conocimiento se encuentra representado por medio de reglas de producción.

### **3.3.3. PERFEX**

#### **Descripción Funcional**

PERFEX es un sistema basado en conocimiento clínico útil para interpretar distribuciones 3D de perfusión del miocardio. Es un sistema experto desarrollado para asistir en el proceso de

toma de decisiones, proporcionando conocimiento basado en expertos para procesar e interpretar información específica de pacientes, basada tanto en imágenes como en otras fuentes, suministrando además un entorno gráfico dinámico e interactivo. [19].

Este sistema provee los mecanismos necesarios para cumplir con las siguientes funciones:

- Extraer automáticamente características desde imágenes 3D.
- Convertir las características extraídas a formas simbólicas para realizar el procesamiento basado en conocimiento.
- Integrar otros tipos de información de pacientes no basada en imágenes.
- Manejar incertidumbre clasificando la exactitud de la respuesta en varios niveles.
- Determinar la condición general del paciente.
- Presentar la información de diagnóstico relevante en forma amigable y familiar a un médico.

**Entradas del Sistema:**

- La información de entrada consiste de información derivada de imágenes (por ejemplo imágenes correspondientes a tensión) , así como otra información específica del paciente (edad, peso).

**Representación del Conocimiento:**

- La base de conocimiento de PERFEX fue diseñada para trabajar con el conocimiento obtenido desde las imágenes, así como con la información extraída desde los datos del paciente.
- El conocimiento está organizado en términos de islas de conocimiento, cada una de las cuales contiene conocimiento acerca de aspectos particulares del proceso.
- El proceso interpretativo empieza considerando los datos específicos del paciente (diferentes de las imágenes). Esta información considerada al principio incluye edad, sexo,



síntomas y resultados de electrocardiografía, y es usada para obtener un diagnóstico inicial (La información de sexo y edad es usada para acceder a otras reglas durante la consulta). Luego es considerada la información de las imágenes.

- PERFEX tiene una base de conocimiento con más de 300 reglas de producción. Las reglas contienen conocimiento de expertos y estrategias de razonamiento representando el proceso de diagnóstico.
- Acerca del mecanismo de inferencia, durante una consulta típica, la inferencia es conducida en un modo de encadenamiento hacia delante: De los datos iniciales a las hipótesis (Aunque es posible hacerlo en forma contraria).
- PERFEX incluye razonamiento con incertidumbre, para ello incluye reglas difusas, razonamiento probabilístico y redes de aprendizaje

### **Criterios de Diseño**

- El conocimiento de PERFEX se hace modular al clasificarlo en conjuntos de reglas a las que se le llama Islas.
- Las reglas de PERFEX, a diferencia de las reglas de producción utilizadas normalmente en Sistemas Expertos (las cuales están conformadas por las premisas y la conclusión) contienen premisas, conclusión y cláusulas de acción.
- PERFEX fue diseñado para ser utilizado en ambientes clínicos reales, por esta razón, se hace énfasis en un diseño de interfaces gráficas de usuario bastante interactivo y fácil de usar por los usuarios finales.

### **Comparación con SIEXCO-TB**

- Este sistema utiliza reglas de producción obtenidas mediante el análisis de casos clínicos reales con la colaboración de médicos expertos. SIEXCO –TB al igual que PERFEX, utiliza reglas de producción para representar el conocimiento, el cual esta organizado en islas, cada una de las cuales contiene conocimiento acerca de aspectos particulares en todo el proceso interpretativo.

- En el caso de SIEXCO-TB, esas “islas” se refieren a grupos de reglas muy específicas ligadas a un tratamiento en particular. Sin embargo existe una parte del conocimiento conformado por reglas generales, en donde se analizan parámetros como sexo, edad, tiempo de tratamiento, resultado de las baciloscopias, para luego acceder a un grupo de reglas más específico.
- A diferencia de PERFEX que usa encadenamiento hacia adelante, SIEXCO-TB trabaja con encadenamiento hacia atrás.

### **3.3.4. Murmur Clinic**

#### **Descripción Funcional:**

Murmur Clinic es un sistema experto de ayuda en la toma de decisiones en el proceso de auscultación cardíaca. La auscultación es una técnica usada en examinación cardíaca para detectar irregularidades mediante el análisis de los sonidos del corazón. [20].

Murmur Clinic fue diseñado para simular a un médico experto. El proceso de razonamiento es modelado teniendo en cuenta las siguientes tareas:

- Adquirir información de entrada en cada caso;
- Evaluar la importancia relativa de los diferentes signos y síntomas; y
- Generar y Evaluar hipótesis, y entonces producir decisiones por eliminación de candidatos menos calificados.

#### **Entradas del Sistema:**

- Murmur Clinic tiene como información de entrada, la obtenida del proceso de auscultación, así como otros síntomas e información específica del paciente.

### **Representación del Conocimiento:**

- Mumar Clinic trabaja con tres tipos de conocimiento: El *conocimiento asociativo* de relaciones sintomáticas, el *conocimiento discriminatorio* acerca del grado de importancia de entidades sintomáticas y el *conocimiento crítico* de hechos médicos. El conocimiento asociativo de murmur clinic es principalmente extraído de diferentes textos médicos y publicaciones; los conocimientos discriminatorio y crítico involucran la información obtenida mediante la interrogación a expertos humanos.
- Murmur Clinic hace énfasis en la modularidad del conocimiento con el fin de poder actualizar y mantener las tres categorías del conocimiento independientemente. Los conocimientos asociativo y discriminatorio son representados por estructuras de conceptos que contienen atributos los cuales modifican sus valores de acuerdo a las manifestaciones encontradas en el proceso de auscultación.
- El conocimiento crítico, conformado por hechos médicos, es representado mediante reglas de producción de la forma “SI ... ENTONCES”. Estas reglas son clasificadas en subcategorías, cada una de las cuales tiene subtareas bien definidas. Las reglas, además, son clasificadas con “terminantes” y “no terminantes”. La ejecución de reglas terminantes inhibe la activación de las otras reglas en el mismo grupo, mientras que la ejecución de reglas no terminantes tiene efectos simplemente adicionales a los de las demás reglas.

### **Criterios de Diseño:**

- Murmur Clinic incorpora un componente explicativo que le permite justificar y explicar su razonamiento al usuario. Este componente, además de explicar el por que y como de los resultados obtenidos, explica por que una conclusión es preferida sobre otra.
- La modularidad del conocimiento, dividido en partes, con diferentes formas de representación hace que este se acerque más al conocimiento del médico experto y el proceso de diagnóstico sea más exacto.

### **Comparación con SIEXCO-TB:**

- Murmur Clinic, divide el conocimiento en categorías, cada una de las cuales tiene una forma de representación. Dentro de estas categorías esta el conocimiento crítico, el cual esta

representado por reglas de producción, las cuales a su vez se dividen en reglas concluyentes y no concluyentes. SIEXCO-TB divide sus reglas de manera similar: Como en Murmur Clinic existe un tipo de reglas cuya ejecución inhibe la activación de las demás del mismo grupo, mientras que existe otras cuya ejecución tiene efectos simplemente agregados a los producidos por las demás reglas.

## CAPITULO 4

### SIEXCO-TB COMO HERRAMIENTA DE SOPORTE PARA EL TRATAMIENTO DE LA TUBERCULOSIS

#### 4.1 INTRODUCCIÓN

SIEXCO – TB fue concebida como una herramienta de soporte para todo el conjunto de actividades y procedimientos tanto médicos como administrativos que se llevan a cabo en nuestro país para la atención oportuna y eficiente de la Tuberculosis.

Con esto en mente, el sistema está habilitado para proponer un tratamiento adecuado de acuerdo a las características del paciente, a la guía de atención de la Tuberculosis pulmonar y extrapulmonar del ministerio de protección social y al conocimiento de tipo heurístico proporcionado por médicos expertos. De esta manera SIEXCO –TB facilita la toma de decisiones terapéuticas de los profesionales de la salud, en el tratamiento de pacientes afectados por la enfermedad, garantizando un mayor cumplimiento de las metas propuestas por el Ministerio de protección social.

Aunque el corazón del sistema es el módulo experto, SIEXCO – TB provee otras funcionalidades de interés fundamental en el manejo administrativo de la enfermedad. Así, suministra los mecanismos necesarios para generar los siguientes informes:

- *Análisis de Cohorte:* Es un registro que sirve para obtener indicadores epidemiológicos y operativos del programa. En este informe se realiza trimestralmente e incluye información tal como la negativización de las baciloscopias al segundo mes de la cohorte a evaluar y los resultados del tratamiento en la cohorte de casos con BK positivo.
- *Informe Trimestral de Casos y Actividades:* Al igual que el análisis de cohorte se muestra trimestralmente y sirve para obtener indicadores epidemiológicos y operativos del programa. Muestra información tal como:

Los enfermos nuevos diagnosticados en el trimestre, clasificados en: Casos pulmonares y extrapulmonares, nuevos y recaídas, BK positivos y BK negativos.

Las actividades desarrolladas durante el trimestre: Sintomáticos respiratorios examinados, baciloscopias realizadas, personas positivas al cultivo, etc.

- *Registro de Pacientes con Tuberculosis:* Es un registro dinámico y funcional que permite seguir a los pacientes en el tiempo, facilitando el análisis de información tal como el control mediante baciloscopias, la condición al egreso, entre otros.

Adicionalmente al manejo médico y administrativo, SIEXCO – TB suministra información estadística útil para ser analizada por Epi Info, una herramienta de dominio público que sirve para analizar datos con estadísticas de uso en epidemiología y representarlos con gráficos y mapas.

SIEXCO – TB es una de las herramientas del proyecto SINCO – TB (Sistema Inteligente para el control de la Tuberculosis). Interactúa con las demás herramientas suministrándole información de los pacientes:

Al SIM, un simulador genérico de difusión de la tuberculosis utilizado para soportar las hipótesis de difusión de la enfermedad, le proporciona información útil para analizar la difusión de la tuberculosis.

Al SIG, un sistema de información geográfico, cuyo objetivo es determinar el aumento o disminución de la tuberculosis, identificando los grupos de población que sufren la enfermedad y donde se localizan, se le entrega información relacionada con la ubicación geográfica del paciente.

SIEXCO-TB está diseñado para ser utilizado en ambientes clínicos reales por diferentes tipos de usuario. Soporta 4 roles bien definidos: Médico Experto, Enfermera, Médico general, Administrador del Sistema.

El sistema está pensado para tener como usuarios directos del sistema a los profesionales de la salud (médicos, enfermeras, auxiliares de enfermería) de la red pública y privada de atención en salud y los funcionarios de los programas de control de tuberculosis de todo el país.

Indirectamente podrían ser usuarios del sistema las facultades de medicina de las universidades públicas y privadas, quienes podrán usar el Sistema Experto en actividades de capacitación. Los

enfermos de tuberculosis inscritos en los Programas de Control de Tuberculosis de la red pública o privada de atención de salud.

En este capítulo se describe el sistema desarrollado, explicando su diseño modular, arquitectura, modelo de representación del conocimiento y demás características del diseño. De la misma forma, se exponen algunos detalles de implementación y las tecnologías empleadas en el proceso de desarrollo.

## 4.2 ANÁLISIS DEL SISTEMA

### 4.2.1 Caso del Dominio

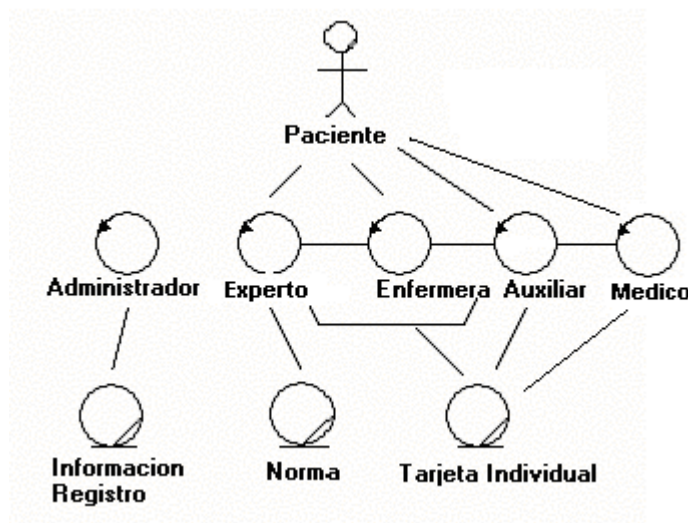


Figura 4 Caso del dominio

#### Actores del dominio.

**Paciente:** Es la persona enferma de Tuberculosis que esta recibiendo el tratamiento en el centro médico.

#### Trabajadores del dominio

**Administrador:** Es la persona encarga de gestionar el sistema. Entre sus funciones se cuentan la de modificar la información de configuración del sistema y crear, actualizar y eliminar

cuentas de usuario, experto y administrador. El administrador del sistema puede ser un Medico, Enfermero y/o Auxiliar de enfermería.

**Experto:** Es la persona que interactúa directamente con el sistema experto. Es el encargado de alimentar la base del conocimiento del sistema.

**Medico:** Es quien realiza consultas al sistema experto y quien finalmente toma las decisiones en lo que tiene que ver con el tratamiento a seguir por el paciente. En el tratamiento convencional, el médico realiza controles al paciente al 2, 4 y 6 mes o al final del tratamiento.

**Enfermera profesional:** Realiza un control al paciente cada dos meses, al 1, 3 y 5 mes del tratamiento. En caso de encontrar problemas de intolerancia medicamentosa, toxicidad, enfermedad intercurrente u otra situación que justifique el control médico, el paciente debe ser remitido a consulta.

**Auxiliar de Enfermería:** Interactúa diariamente con el paciente. Realiza un control cada vez que el enfermo asiste a recibir la dosis correspondiente de medicamentos al organismo de salud.

## Entidades

**Información de Registro:** Es la información de los usuarios del sistema ya sea Administrador, experto, médico, enfermera y auxiliar de enfermería. Esta conformada por sus datos personales y el login y el password proporcionados para ingresar al sistema.

**Norma nacional para el control de la tuberculosis:** Es la guía oficial expedida por el ministerio de protección social para el control de la tuberculosis. Es la fuente principal de donde se extrae la base de conocimiento del sistema.

**Tarjeta individual:** Es el registro dinámico y funcional que permite identificar al enfermo y sus contactos, su estado bacteriológico, evolución e intolerancia a los medicamentos entre otras. Permite conocer al organismo responsable del Control, los siguientes aspectos: Número de enfermos en tratamiento, cumplimiento en la asistencia, pérdida de control y otros aspectos técnico-administrativos como los resultados del tratamiento de la cohorte.



### 4.2.2 Casos de uso esenciales

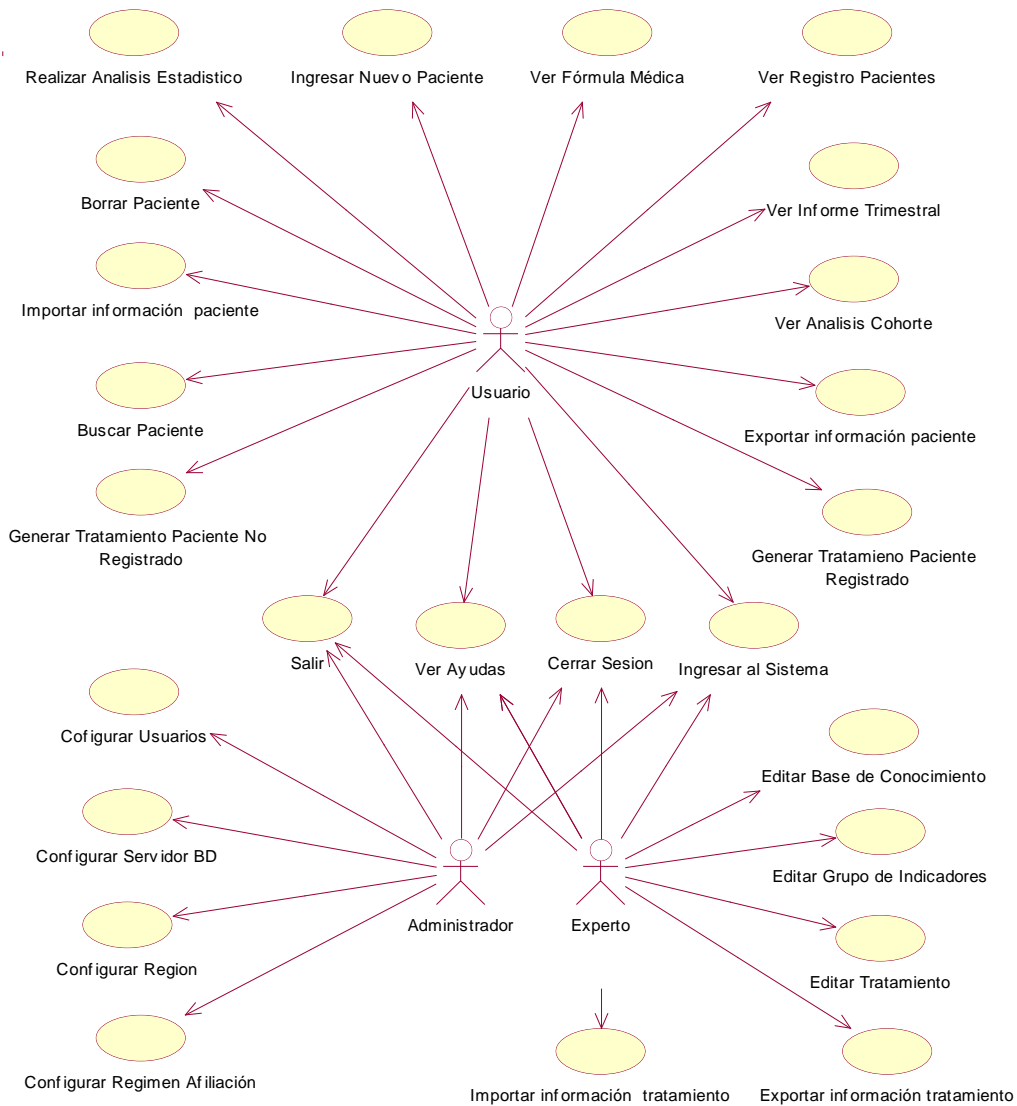


Figura 5 Diagrama de Casos de Uso

#### 4.2.2.1 Caso de uso: Ingresar al sistema

**Iniciador:** Usuario, Administrador, Experto.

**Resumen:** Este caso de uso se inicia cuando se desea ingresar al sistema. En este caso el sistema desplegará una interfaz gráfica por medio de la cual permite el acceso al sistema, en esta interfaz el usuario deberá ingresar su login y password y presionar el botón “Aceptar”. Al hacer esto el sistema verificará si el usuario es válido e identificará el tipo de usuario (Experto,

Administrador, Usuario), dependiendo del tipo de usuario el sistema le permitirá acceder a determinados servicios. Si el usuario no es válido el sistema no le permitirá el acceso y le desplegará un mensaje indicando este hecho.

#### 4.2.2.2 Caso de uso: Borrar Paciente.

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** Este caso de uso se inicia cuando el usuario selecciona la opción del menú principal “Borrar paciente”, al hacer esto el sistema desplegará una interfaz gráfica por medio de la cual brindará dos alternativas para eliminar un paciente. Para utilizar la primera alternativa el usuario deberá seleccionar la opción “Número de Historia”, luego ingresar el número de historia del paciente y presionar el botón borrar. Si el paciente se encuentra en la base de datos el sistema desplegará un mensaje de confirmación indicando si realmente desea borrar la información del paciente, si la confirmación es positiva el paciente será borrado automáticamente, de lo contrario el sistema no eliminará ningún tipo de información. La segunda alternativa consiste en seleccionar la opción “Año de inicio tratamiento”, luego ingresar el año en que el paciente inició el tratamiento y presionar el botón “Buscar”. Si el paciente se encuentra registrado el sistema desplegará una lista con el nombre de todos los pacientes registrados en ese año, en esta lista el usuario podrá seleccionar el nombre del paciente que desea eliminar y presionar el botón borrar, al hacer esto el sistema desplegará un mensaje de confirmación indicando si realmente desea borrar la información del paciente, si la confirmación es positiva el paciente será borrado automáticamente, de lo contrario el sistema no eliminará ningún tipo de información.

#### 4.2.2.3 Caso de uso: Salir

**Iniciador:** Usuario, Administrador, Experto.

**Resumen:** Este caso de uso se inicia al seleccionar la opción “Salir” del menú principal del sistema. Al hacer esto el sistema desplegará un mensaje de confirmación indicando si realmente desea salir de la aplicación. Si la confirmación es positiva se cerrará el programa automáticamente, de lo contrario el usuario podrá seguir utilizando la aplicación.

#### 4.2.2.4 Caso de uso: Buscar Paciente

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** Este caso de uso se inicia cuando el usuario selecciona la opción del menú principal “Paciente Registrado”, al hacer esto el sistema desplegará una interfaz gráfica por medio de la cual brindará dos alternativas para buscar un paciente. Para utilizar la primera alternativa el usuario deberá seleccionar la opción “Número de Historia”, luego ingresar el número de historia del paciente y presionar el botón “Buscar”. Si el paciente se encuentra en la base de datos el sistema desplegará un formulario con toda la información del paciente correspondiente a la tarjeta individual, de lo contrario el sistema desplegará un mensaje indicando que el paciente no se encuentra registrado. La segunda alternativa consiste en seleccionar la opción “Año de inicio tratamiento”, luego ingresar el año en que el paciente inició el tratamiento y presionar el botón “Buscar”. Si el paciente se encuentra registrado el sistema desplegará una lista con el nombre de todos los pacientes registrados en ese año, en esta lista el usuario podrá seleccionar el nombre del paciente que desea buscar y presionar el botón “Buscar”, al hacer esto el sistema desplegará un formulario con toda la información del paciente correspondiente a la tarjeta individual. Mediante dicho formulario se tiene acceso a la interfaz de tratamiento del paciente de acuerdo a sus datos particulares.

#### 4.2.2.5 Caso de uso: Ingresar Nuevo Paciente

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** Este caso de uso se inicia cuando el usuario selecciona la opción “Nuevo Paciente” del menú principal del sistema. Al hacer esto el sistema desplegará una interfaz gráfica que consiste principalmente en un formulario. En este formulario el usuario podrá ingresar toda la información del paciente correspondiente a la tarjeta individual. Adicionalmente el usuario podrá enviar esta información por correo electrónico, imprimir el formulario y guardar la información en el sistema.

#### 4.2.2.6 Caso de uso: Cerrar Sesión

**Iniciador:** Usuario, Administrador, Experto.

**Resumen:** Este caso de uso se inicia cuando el usuario selecciona la opción “Cerrar Sesión” del menú principal del sistema. Al hacer esto el sistema ocultará la aplicación y desplegará la

interfaz grafica que permite el acceso al sistema, en esta interfaz el usuario deberá ingresar su login y password y presionar el botan aceptar. Después de esto el sistema verificara si el usuario es valido e identificara el tipo de usuario (Experto, Administrador, Usuario), dependiendo del tipo de usuario, el sistema le permitirá acceder a determinados servicios. Si el usuario no es valido el sistema no el permitirá el acceso y le desplegara un mensaje indicando este hecho.

#### 4.2.2.7 Caso de uso: Configurar Usuario

**Iniciador:** Administrador.

**Resumen:** Este caso de uso se inicia cuando el administrador del sistema selecciona la opción “Usuarios” del menú principal del sistema. Al hacer esto el sistema desplegara una interfaz grafica que consistirá principalmente de una tabla que contiene los datos de todos los usuarios registrados, por medio de esta interfaz grafica el sistema le permitirá al administrador ingresar un nuevo usuario, modificar o eliminar los datos de uno ya existente. Para ingresar un nuevo usuario el administrado deberá dar clic sobre el botón “Adicionar” al hacer esto el sistema le desplegara un formulario en el cual podrá ingresar los datos personales del nuevo usuario así como el tipo de usuario su login y password. Para modificar los datos de un usuario ya existente primero debe seleccionar, en la tabla, el usuario a modificar, después de esto debe dar clic sobre el botón “Modificar”, al hacer esto el sistema le desplegara un formulario con todos los datos del usuario en donde el administrador puede modificar cualquiera de ellos y guardar los cambios realizados. Para eliminar un usuario el administrador debe seleccionar, en la tabla, el usuario a eliminar, después de esto debe dar clic sobre el botón “Eliminar”. Antes de eliminar los datos del usuario, el sistema desplegara un mensaje de confirmación con el fin de verificar si realmente se desea eliminar los datos del usuario seleccionado. Si la confirmación es positiva el sistema eliminara los datos del usuario, de lo contrario el sistema no borrará ningún tipo de información.

#### 4.2.2.8 Caso de uso: Configurar Régimen Afiliación

**Iniciador:** Administrador.

**Resumen:** Este caso de uso se inicia cuando el usuario selecciona la opción “Ips” o “Administradora” del menú principal del sistema. Si selecciona la opción Ips el sistema le desplegara un interfaz grafica que contendrá una lista con todas las Ips registradas en el sistema, en esta interfaz grafica el sistema le permitirá ingresar o remover una Ips. Para ingresar una Ips

el administrador deberá ingresar el nombre y oprimir el botón “Adicionar”, después de esto el sistema adicionara la Ips a la lista automáticamente. Para eliminar una Ips el Administrador primero deberá seleccionarla de la lista y luego oprimir el botón “Eliminar”, después de esto el sistema desplegara un mensaje de confirmación para verificar si el administrador realmente desea eliminar la Ips seleccionada. Si la confirmación es positiva el nombre de la ips será borrado automáticamente de la lista, de lo contrario el sistema no eliminara ningún tipo de información. Cuando el Administrador selecciona la opción “Administradora” ocurre el mismo proceso mencionado anteriormente, la única diferencia es que ahora no se trata de Ips si no de Administradora de salud.

#### 4.2.2.9 Caso de uso: Configurar Región.

**Iniciador:** Administrador.

**Resumen:** Este caso de uso se inicia cuando el usuario selecciona la opción “Municipio” o “Departamento” del menú principal del sistema. Si selecciona la opción municipio el sistema le desplegara un interfaz grafica en la que se encontrara dos listas. La primera con todos los departamentos registrados y la segunda con los municipios pertenecientes al departamento seleccionado en la lista de departamentos, en esta interfaz grafica el sistema le permitirá al administrador ingresar o remover un municipio. Para ingresar un nuevo municipio el administrador deberá seleccionar el departamento donde se encuentra el municipio, después de esto el sistema cargara automáticamente el la lista de municipios, todos los municipios registrados en ese departamento, posteriormente el administrador deberá ingresar el nombre del municipio y oprimir el botón “Adicionar”, después de esto el sistema adicionara el municipio a la lista de municipios automáticamente. Para eliminar un municipio el administrador primero deberá seleccionar el departamento donde se encuentra el municipio para que los nombres de los municipios registrados en ese departamento se carguen automáticamente en la lista de municipios, después de esto deberá seleccionar el nombre del municipio y oprimir el botón “Eliminar”, antes de eliminar el nombre del municipio el sistema desplegara un mensaje de confirmación para verificar si el administrador realmente desea eliminar el municipio seleccionado. Si la confirmación es positiva el sistema eliminara de la lista el nombre del municipio, de lo contrario el sistema no borrara ningún tipo de información.

Por otra parte si el Administrador selecciona la opción “Departamento” el sistema le desplegara un interfaz grafica con la lista de todos los departamentos registrados. En esta interfaz gráfica el Administrador podrá adicionar o remover un departamento. Para adicionar un departamento el

Administrador deberá ingresar el nombre del departamento y oprimir el botón “Adicionar”, después de esto el sistema adicionara automáticamente el nombre del departamento a la lista. Para remover un departamento el administrador deberá seleccionar su nombre de la lista de departamentos y oprimir el botón “Remover”. Antes de eliminar el nombre del departamento el sistema desplegara un mensaje de confirmación con el fin de verificar si realmente desea eliminar el nombre del departamento seleccionado. Si la confirmación es positiva el sistema eliminara el nombre del departamento, de lo contrario el sistema no borrara ningún tipo de información.

#### **4.2.2.10 Caso de uso: Generar Tratamiento Paciente Registrado.**

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería).

**Resumen:** Este caso de uso se inicia cuando el usuario da clic sobre el botón “Generar Tratamiento” que se encuentra en la interfaz grafica que muestra los datos del paciente. Al hacer esto el sistema desplegara una interfaz grafica por medio de la cual se puede realizar la búsqueda del paciente por año de ingreso al programa ó número de historia. Al encontrar el paciente el sistema despliega su tarjeta individual a través de a cual se puede acceder al tratamiento del paciente presionando un botón. Al presionar el botón se solicitan otros datos adicionales necesarios al momento de generar un tratamiento adecuado para el control de la tuberculosis. En esta interfaz el usuario puede seleccionar datos relacionados con: el Esquema del paciente, situaciones especiales que presente el paciente, Medicamentos a los que el paciente puede ser resistente o sensible, ingresar el peso y la edad del paciente. Después de ingresar toda la información del paciente el usuario deberá oprimir el botón “Aceptar”, al hacer esto el sistema le generara y desplegara automáticamente el tratamiento a seguir por el paciente afectado con tuberculosis. Este se mostrara en un formato similar al utilizado en la guía de atención de la tuberculosis pulmonar y extrapulmonar. Finalmente el sistema permitirá al usuario cargar la información del tratamiento en el formulario donde se encuentran los datos del paciente (Interfaz grafica donde el usuario inicio este caso de uso).

#### **4.2.2.11 Caso de uso: Generar Tratamiento Paciente No Registrado.**

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería).

**Resumen:** Este caso de uso se inicia cuando el usuario selecciona la opción “Paciente no Registrado” que se encuentra en el menú principal. Al hacer esto el sistema desplegara una

interfaz grafica donde el usuario puede ingresar los datos del paciente que son necesarios al momento de generar un tratamiento adecuado para el control de la Tuberculosis. En esta interfaz el usuario puede seleccionar datos relacionados con: el Esquema del paciente, situaciones especiales que presente el paciente, Medicamentos a los que el paciente puede ser resistente o sensible, ingresar el peso y la edad del paciente. Después de ingresar toda la información del paciente el usuario deberá oprimir el botón “Aceptar”, al hacer esto el sistema le generara y desplegara automáticamente el tratamiento a seguir por el paciente afectado con tuberculosis. Este se mostrara en un formato similar al utilizado en la guía de atención de la tuberculosis pulmonar y extrapulmonar. Después de generado el tratamiento el sistema le permitirá al usuario generar automáticamente la formula medica para el paciente de acuerdo a los medicamentos y dosis indicados en el tratamiento, así como, visualizar el razonamiento que utilizo para llegar a la solución final (Tratamiento). Finalmente el sistema permitirá al usuario salir de la aplicación.

#### 4.2.2.12 Caso de uso: Ver Fórmula Médica

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** Este caso de uso se inicia cuando un usuario del sistema desea visualizar una fórmula médica, previamente almacenada, de un paciente determinado. El usuario debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente se le desplegará el menú del usuario donde podrá escoger la opción de visualizar fórmula médica. Al seleccionar esta opción, el sistema solicitará el número de historia del paciente, dato con el cual busca las fórmulas médicas asociadas al paciente que han sido almacenadas en la base de datos del sistema y despliega una lista con la fecha de las encontradas.

El usuario deberá seleccionar la fórmula médica que desea visualizar y confirmar la selección mediante un botón aceptar. Al confirmar la selección el sistema desplegará la fórmula médica, mostrando las opciones de imprimir fórmula ó cerrar la ventana. Si el usuario escoge la opción imprimir, se le mostrará una ventana de diálogo en donde podrá escoger las diferentes propiedades de impresión antes de confirmar la impresión de la fórmula.

#### 4.2.2.13 Caso de uso: Ver Registro Pacientes

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** Este caso de uso se inicia cuando un usuario del sistema desea visualizar un listado de pacientes registrados en el sistema. El usuario debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente se le desplegará el menú del usuario donde podrá escoger la opción de visualizar registro de pacientes. Al seleccionar esta opción se despliega una interfaz donde el usuario podrá ingresar información acerca del departamento, municipio y año de ingreso al programa de los pacientes que desea visualizar. Al confirmar los datos de ingreso para el informe se despliega un listado de pacientes con los datos más relevantes del paciente. En esta interfaz, se muestran las opciones de imprimir el informe y ver más información de cada paciente.

Si el usuario escoge la opción imprimir, se le mostrará una ventana de diálogo en donde podrá escoger las diferentes propiedades de impresión antes de confirmar la impresión del informe.

Si el usuario selecciona un paciente y escoge la opción de ver más información, se le desplegará una interfaz con información detallada del paciente. La información desplegada en esta interfaz no es la información completa del paciente, razón por la cual se incluye un botón por medio del cual el usuario podrá tener acceso a la tarjeta individual correspondiente.

#### 4.2.2.14 Caso de uso: Ver Informe Trimestral

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** Este caso de uso se inicia cuando un usuario del sistema desea visualizar un informe trimestral. El usuario debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente se le desplegará el menú del usuario donde podrá escoger la opción de visualizar informe trimestral. Al seleccionar esta opción se despliega una interfaz donde el usuario podrá ingresar información acerca del departamento, municipio, año de ingreso, trimestre e Ips para la que se desea generar el informe. Al confirmar los datos de ingreso para el informe puede suceder lo siguiente:

A. Que el informe haya sido generado previamente. En este caso se informa al usuario que el informe ya fue generado y que si quiere reemplazarlo ó no. Si el usuario decide reemplazarlo, se genera un nuevo informe el cual es desplegado en pantalla. En caso de que no desee



reemplazarlo el sistema preguntará si desea ver el informe almacenado ó abandonar este submenú. Si el usuario escoge visualizar el informe, se le mostrará el previamente almacenado, si no, regresará al menú principal.

B. Que el informe no haya sido generado previamente. En este caso se genera un nuevo informe el cual es desplegado en pantalla.

En la interfaz que muestra el informe trimestral, se muestra el informe completo, además de las opciones de guardarlo, enviarlo vía correo electrónico, imprimirlo y volver al menú principal.

Si el usuario elige la opción *guardar* se despliega una interfaz en la cual deberá llenar información correspondiente al nombre del responsable del informe y a la fecha de generación.

Si el usuario elige la opción *enviar* se despliega una interfaz en la cual deberá llenar información correspondiente al nombre del responsable del informe, a la fecha de generación, a su correo electrónico, al correo de destino y al asunto y cuerpo del mensaje. Al presionar un botón de confirmación de esta información, se envía el correo electrónico con un archivo adjunto correspondiente al informe.

Si el usuario escoge la opción *imprimir*, se le mostrará una ventana de diálogo en donde podrá escoger las diferentes propiedades de impresión antes de confirmar la impresión del informe.

#### **4.2.2.15 Caso de uso: Ver Análisis de Cohorte**

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** El proceso para el análisis de cohorte es idéntico al llevado a cabo en el caso de uso Ver informe trimestral.

#### **4.2.2.16 Caso de uso: Importar Información Pacientes**

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** Este caso de uso se inicia cuando un usuario del sistema desea importar información relacionada con pacientes. El usuario debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente se le

desplegará el menú del usuario donde podrá escoger la opción de importar Información de pacientes.

Al seleccionar esta opción, el usuario podrá importar una tarjeta individual, un informe trimestral ó un análisis de cohorte.

En cualquiera de las opciones de importación, el sistema despliega una interfaz para buscar el archivo a importar. Una vez el usuario encuentre el archivo a importar deberá seleccionarlo y presionar un botón de confirmación de la selección. Al realizar esto el sistema valida el tipo de archivo (Archivos .xml) y la información contenida en el para verificar si es un archivo correspondiente a la información que se pretende almacenar en el sistema (una tarjeta individual, un informe trimestral ó un análisis de cohorte). Si la información no es válida el sistema despliega un mensaje de error, de lo contrario un mensaje confirmando que la información fue almacenada.

#### **4.2.2.17 Caso de uso: Importar Información del Tratamiento**

**Iniciador:** Experto

**Resumen:** Este caso de uso se inicia cuando un usuario del sistema desea importar información relacionada con tratamientos y/o reglas de la base de conocimiento. El proceso de importación de esta información es similar al que se lleva a cabo en el caso de uso *importar información Pacientes*.

#### **4.2.2.18 Caso de uso: Exportar Información Pacientes**

**Iniciador:** Usuario (Médico, Enfermera, Auxiliar de Enfermería)

**Resumen:** Este caso de uso se inicia cuando un usuario del sistema desea exportar información relacionada con pacientes. El usuario debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente se le desplegará el menú del usuario donde podrá escoger la opción de exportar Información de pacientes.

Al seleccionar esta opción, el usuario podrá exportar una tarjeta individual, un informe trimestral ó un análisis de cohorte.

*Exportar una Tarjeta Individual.* Al seleccionar exportar tarjeta individual se despliega una interfaz en la cual el se debe ingresar el número de historia de la tarjeta a exportar ó el año de inicio del tratamiento.

Si se ingresa el número de historia, el sistema busca la tarjeta con correspondiente a dicha identificación y la exporta al directorio especificado por el usuario.

Si se ingresa el año de inicio del tratamiento se despliega la lista de todos los pacientes que ingresaron ese año. El usuario podrá seleccionar un paciente de la lista y el directorio donde desea guardar la tarjeta individual correspondiente.

*Exportar un informe trimestral ó un análisis de cohorte.* Para exportar uno de estos informes, el usuario deberá ingresar en una interfaz desplegada por el sistema, información sobre el departamento, municipio, año, trimestre e Ips del informe. Al confirmar esta información, el sistema busca el informe en la base de datos, si no lo encuentra, despliega un mensaje informando que el informe no ha sido generado. En caso de que lo encuentre, desplegará una interfaz en la cual el usuario escogerá el directorio donde desea guardar el informe. Finalmente el usuario deberá confirmar que desea exportar el archivo, acción mediante la cual se copiará el archivo al directorio seleccionado.

#### **4.2.2.19 Caso de uso: Exportar Información del Tratamiento**

**Iniciador:** Experto

**Resumen:** Este caso de uso se inicia cuando un usuario del sistema desea exportar información relacionada con tratamientos y/o reglas de la base de conocimiento.

*Exportar Reglas.* Al elegir exportar reglas, el sistema muestra todos los archivos .xkb que conforman la base de conocimiento. El experto deberá elegir uno de esos archivos y presionar un botón de confirmación. Al realizar esta acción, el sistema despliega una interfaz por medio de la cual el experto puede escoger el directorio a donde desea enviar el archivo seleccionado. Después de que el experto seleccione un directorio a donde enviar el archivo y lo confirme, se exportará el archivo.

*Exportar Tratamientos.* Al seleccionar esta opción, el sistema despliega una interfaz en la cual el usuario debe ingresar el nombre del tratamiento a exportar. El sistema busca el tratamiento y

despliega una interfaz por medio de la cual el experto puede escoger el directorio a donde desea enviar el archivo del tratamiento seleccionado. Después de que el experto seleccione un directorio a donde enviar el archivo y lo confirme, se exportará el archivo.

#### 4.2.2.20 Caso de uso: Ver Ayudas

**Iniciador:** Usuario, Administrador, Experto.

**Resumen:** Este caso de uso se inicia cuando un usuario consulta las ayudas que el sistema le ofrece. El usuario puede obtener información sobre los siguientes aspectos: Guía de la tuberculosis pulmonar y extrapulmonar, Formatos y registros para el control de la gestión de la tuberculosis y manual de usuario de SIEXCO-TB.

Al consultar cualquiera de estos tipos de ayuda el sistema despliega una interfaz que contiene: Una tabla de contenido, un índice, y un mecanismo de búsqueda para fácil navegación sobre el contenido de las ayudas. Además provee un conjunto de palabras claves para búsqueda rápida de contenidos. El usuario puede hacer uso de cualquiera de estas funcionalidades, además de imprimir cualquiera de los temas desplegados por dicha interfaz de ayuda.

#### 4.2.2.21 Caso de uso: Realizar Análisis Estadístico

**Iniciador:** Usuario, Experto.

**Resumen:** Este caso de uso se inicia cuando un usuario desea realizar análisis estadísticos. El usuario debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente se le desplegará el menú del usuario donde podrá escoger la opción de realizar análisis estadístico. Al escoger esta opción, se ejecutará la interfaz principal de Epi Info, la cual ofrece la opción de *Analizar datos*. Al seleccionar esta opción, el usuario tendrá acceso a la base de datos de estadísticas de SIEXCO – TB y a las diferentes opciones de análisis estadístico.

Aquí el usuario puede consultar algunas estadísticas relacionadas con efectos colaterales debido a los medicamentos suministrados a los pacientes y algunos indicadores operativos y epidemiológicos del programa de atención de tuberculosis. Después de realizar las estadísticas requeridas por el usuario, este podrá abandonar este submenú por medio de la opción salir del menú principal de Epi Info.

#### 4.2.2.22 Caso de Uso: Editar base de conocimiento

**Iniciador:** Experto

**Resumen:** Este caso de uso se inicia cuando el experto en Tuberculosis decide modificar la base de conocimientos del sistema experto. El experto debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente al sistema, se le presentara el menú principal del experto en donde podrá escoger, entre otras opciones, la opción “Editar base de conocimiento”. Al escoger esta opción el experto podrá realizar las siguientes acciones:

*Crear una nueva base de Conocimiento.*

Por medio de esta opción, el usuario queda habilitado para crear una base de conocimiento desde cero. Permittedosele, por lo tanto definir las nuevas reglas que la conforman.

*Buscar en la base de conocimiento.*

Al elegir esta opción, el sistema despliega una interfaz con las diferentes opciones de búsqueda en la base de conocimiento. Estas opciones.

El usuario puede buscar objetos del conocimiento (reglas, hechos, etc.) usando las siguientes condiciones:

1. Buscar ciertos segmentos de texto.
2. Buscar por tipo de conocimiento (regla, hecho, hechos desde base de datos, etc).
3. Buscar objetos del conocimiento que contengan cierto predicado ó función.
4. Las condiciones pueden ser agrupadas usando conectores “o” e“y”.

*Editar base de Conocimiento.*

Con esta opción, el experto puede editar la base del conocimiento del sistema Experto, para esto, se le ofrecen las siguientes opciones:

### Adicionar Hechos y Reglas

1. Para agregar hechos y reglas a la base de conocimiento, se debe presionar un botón de adicionar conocimiento. Esto desplegará una ventana donde se puede seleccionar el tipo de conocimiento a agregar.
2. Seleccione agregar regla para agregar una regla y agregar hecho para agregar un hecho. Entonces presionar el botón “siguiente”.
3. Si el depósito asociado contiene plantillas de hechos (plantillas de reglas, respectivamente), se puede seleccionar una de las plantillas en la página siguiente.

Después de seleccionar una plantilla y de presionar “siguiente”, la página exhibe un editor para modificar la plantilla seleccionada.

4. Si no hay plantillas disponibles, se salta el tercer paso y se elige un hecho o una regla por defecto.

Después de corregir el hecho o la regla y de presionar finalizar. El hecho o la regla serán agregados a la base de conocimiento.

### Editar las Reglas

Para editar una regla, el usuario debe seleccionar la regla deseada y dar doble clic sobre ella. De esta manera, se desplegará el editor apropiado para el ítem seleccionado.

### Eliminar una Regla

El usuario deberá seleccionar la regla y presionar un botón que confirme el deseo de borrarla de la base de conocimiento

### Ordenar la base de Conocimiento

El conocimiento, de SIEXCO-TB está ordenado (priorizado) y se puede mover en diferentes posiciones. Al seleccionar un ítem, este se puede mover al principio, al final, hacia arriba y hacia abajo.

#### 4.2.2.23 Caso de Uso: Editar Grupo de indicadores

**Iniciador:** Experto

**Resumen:** Este caso de uso se inicia cuando el experto en Tuberculosis decide modificar o ingresar un grupo de indicadores del estado del paciente. El experto debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente al sistema, se le presentara el menú principal del experto en donde podrá escoger, entre otras opciones, la opción “Indicadores”. Al escoger esta opción, El sistema le permitirá modificar un grupo de indicadores ya existente o ingresar un nuevo grupo de indicadores.

Para Ingresar un nuevo grupo de indicadores al sistema se deben seguir los siguientes pasos:

1. Entrar a la opción Nuevo Grupo Indicadores en el ítem indicadores. Se desplegará una interfaz con la que se puede crear un árbol correspondiente a la jerarquía del grupo de indicadores que se desea almacenar. Aquí se muestran las siguientes opciones:
  - Adicionar un nuevo elemento al árbol. Para esto, es necesario colocar el mouse sobre el nodo padre y presionar el botón, esta acción creará un elemento hijo al cual se le podrá cambiar el nombre dando doble clic sobre él. En la jerarquía predefinida, los parámetros son hojas del árbol, es decir no pueden tener hijos.
  - Eliminar un elemento del árbol. . Para esto, es necesario colocar el mouse sobre el elemento a borrar y presionar el botón, esta acción eliminará el elemento.
  - Clarear el árbol, es decir borrar todos los nodos del árbol, excepto el nodo padre.
  - Insertar una descripción para cada elemento. Para esto, seleccionar el elemento, escribir en un área de texto. Todos los elementos deberán tener una descripción, de lo contrario, el árbol no podrá ser almacenado.
2. Una vez construido el árbol se debe almacenar en la base de datos, presionado el botón guardar. Se desplegará una ventana en la cual se debe llenar el nombre con el que se desea guardar el grupo de indicadores construido.

Para modificar un grupo de indicadores al sistema, el usuario debe entrar a la opción Modificar Grupo Indicadores. Se desplegará una interfaz en la cual el usuario deberá digitar el nombre del grupo de indicadores a modificar y presionar buscar. Se desplegará una ventana para modificar el árbol de la misma forma que cuando se ingresa un nuevo grupo de Indicadores.

#### 4.2.2.24 Caso de Uso: Editar Tratamiento

**Iniciador:** Experto

**Resumen:** Este caso de uso se inicia cuando el experto en Tuberculosis decide ingresar o modificar un tratamiento. El experto debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente al sistema, se le presentara el menú principal del experto en donde podrá escoger, entre otras opciones, la opción “Tratamiento”. Al escoger esta opción, el sistema le permitirá ingresar un nuevo tratamiento o modificar uno ya existente, solicitándole información tal como: las fases del tratamiento, duración de las fases, no de dosis, medicamentos, y dosis asociadas a dichos medicamentos.

##### Ingresar nuevo Tratamiento

Para introducir un nuevo tratamiento al sistema, el usuario deberá seguir los siguientes pasos:

1. Llenar los campos de texto correspondientes a nombre del tratamiento, descripción del tratamiento, número de dosis (número de dosis total suministrados en la fase) y duración la fase.
2. Seleccionar los medicamentos de la fase. Así:
  - Seleccionar el medicamento
  - Ingresar información adicional del medicamento como cantidad (número de dosis diarias) y el presentación.
  - Presionar un botón de selección. Esto pasará al medicamento a la lista de selección.
  - Si se desea eliminar un de los medicamentos ya seleccionados. Escoger el medicamento de la lista de selección y presionar el botón de borrar medicamento.



3. En cualquier momento del proceso se puede visualizar el tratamiento introducido, presionando un botón de visualización del tratamiento.
4. Para almacenar el tratamiento presionar el botón aceptar. Acción después de la cual se registrará el tratamiento en el sistema.

#### Modificar Tratamiento

Con esta opción, el experto puede modificar un tratamiento del sistema. Al seleccionar modificar tratamiento, aparecerá una interfaz en la cual se deberá ingresar el nombre del tratamiento a buscar y presionar un botón de búsqueda.

Así, el sistema desplegará una interfaz con el tratamiento cargado. Para modificar el tratamiento, se sigue un proceso similar al seguido para introducir uno nuevo.

#### **4.2.2.25 Caso de Uso: Configurar servidor BD**

**Iniciador:** Administrador

**Resumen:** Este caso de uso se inicia cuando el administrador decide configurar los parámetros de acceso al servidor de la base de datos. El administrador debe haber ingresado al sistema correctamente mediante la introducción de sus respectivos login y password. Una vez haya ingresado correctamente al sistema, se le presentará el menú principal del administrador en donde podrá escoger, entre otras opciones, la opción “Configurar servidor de bases de datos”.

Al escoger esta opción el sistema desplegará una ventana con la información actual de acceso a la base de datos: Host, Login y password. El usuario podrá editar esta información y confirmar los cambios mediante un botón aceptar. Al aceptar los cambios, el sistema escribirá dicha información en un archivo de configuración de la aplicación.

### 4.3 DISEÑO DEL SISTEMA

#### 4.3.1 Arquitectura

A continuación se hace una breve descripción de cada uno de los bloques que conforma la arquitectura.

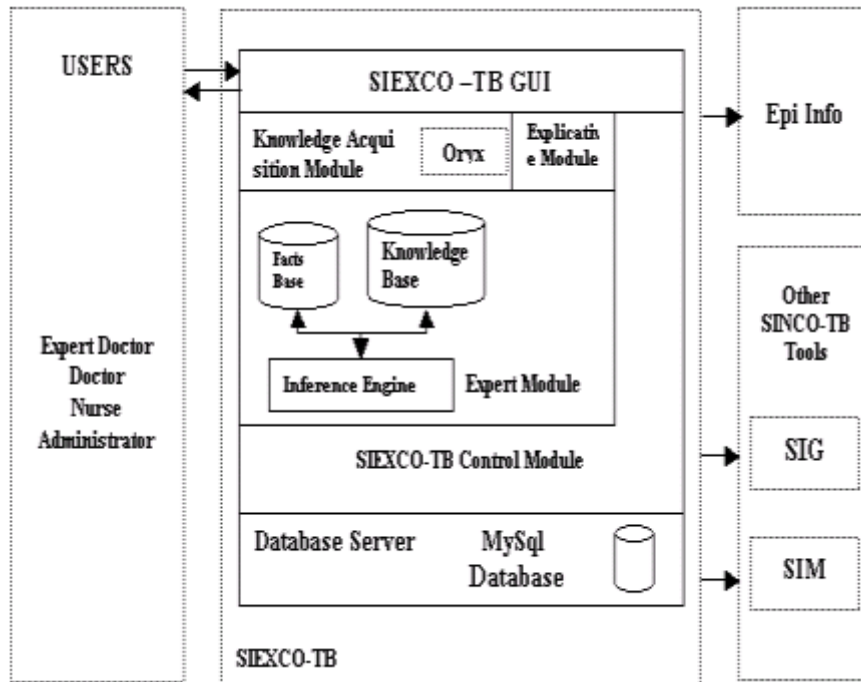


Figura 6 Arquitectura de SIEXCO - TB

##### 4.3.1.1 GUI

Contiene las diferentes interfaces gráficas por medio de las cuales el usuario puede interactuar con el sistema. Dichas interfaces permiten que el usuario suministre información del paciente útil al sistema experto para generar el tratamiento, tal como peso, edad, tiempo de tratamiento, otras enfermedades, etc. Además tiene acceso a la base de conocimiento para editarla cuando lo considere necesario.

A través de la GUI de SIEXCO – TB, el usuario visualiza los informes trimestrales, análisis de cohorte y registros de pacientes, previa introducción de los parámetros necesarios para su generación.

Por medio de la Interfaz principal de SIEXCO – TB, el usuario puede acceder a Epi Info para hacer el análisis de datos estadísticos.

#### **4.3.1.2 Módulo Experto**

Este módulo incluye La representación, manejo y procesamiento (querying ) de la base del conocimiento.

La base de conocimiento de SIEXCO – TB está constituida por reglas de producción, las cuales son manejadas por el mecanismo de inferencia. El mecanismo de inferencia controla que reglas disparan: Aparea las condiciones de las reglas con los hechos, realiza la resolución de conflictos cuando existe más de una regla que puede disparar y aplica las reglas.

#### **4.3.1.3 Módulo de Adquisición del Conocimiento**

Permite editar la base del conocimiento. Esta diseñado para ser usada por expertos del dominio. El objetivo de este módulo, es dar al experto una vista no técnica del Dominio. Las reglas, hechos y otros objetos del dominio son representados como oraciones de lenguaje común y secuencias de palabras. De esta manera, se le ofrece al médico experto una interfase gráfica amigable para el mantenimiento de la base del conocimiento.

Conceptualmente, se agrega el concepto de repositorio. Una base del conocimiento es asociada con un repositorio que contiene meta información acerca de predicados, funciones, fuentes de datos, etc.

#### **4.3.1.4 Módulo Explicativo**

Es el que permite justificar y explicar el análisis completo del problema y las soluciones propuestas. Básicamente, muestra las reglas utilizadas, el por que se utilizaron, de acuerdo a las características del paciente y las conclusiones obtenidas.

#### **4.3.1.5. Módulo de Control**

Provee los mecanismos necesarios para el intercambio de información entre los demás módulos del sistema. Es el encargado de manejar información relacionada con las consultas hechas a la base del conocimiento. Los resultados de estas consultas son procesados para luego ser desplegados por la interfaz de usuario. Por otra parte este módulo también es el encargado de facilitar la consulta y el almacenamiento de la información en la base de datos, la cual esta relacionada con los datos del paciente, usuarios, Expertos y Administradores del sistema.

En este módulo esta implementada toda la lógica necesaria para la generación de los informes de tipo administrativo así como la información estadística utilizada por Epi Info.

#### **4.3.1.6. Servidor de Base de Datos**

SIEXCO-TB utiliza una base de datos relacional para asegurar la persistencia de la información. Esta se encuentra en un servidor de bases de datos el cual puede ser local ó remoto, según lo configure el administrador del sistema. Esta base de datos contiene principalmente información relacionada con las tarjetas individuales de los pacientes, los tratamientos básicos, informes administrativos e información de configuración del sistema.

#### **4.3.1.7. Epi Info**

Epi Info es un programa de dominio público diseñado por el Centro para el Control de Enfermedades de Atlanta (CDC) de especial utilidad para la Salud Pública. Infinidad de profesionales en el mundo, han utilizado y desarrollado aplicaciones en Epi Info para resolver sus necesidades. Tiene un sistema fácil para analizar datos con las estadísticas de uso básico en epidemiología y representarlos con gráficos y mapas. SIEXCO-TB genera información compatible con los formatos manejados por Epi Info, acerca de los efectos colaterales producidos por los medicamentos según un registro de pacientes que lleva el sistema, relacionado directamente con el tratamiento proporcionado. Además, SIEXCO-TB proporciona información relacionada con indicadores epidemiológicos y operativos del programa.

#### 4.3.1.8. Otras Herramientas SINCO-TB

Dentro de SINCO-TB (Sistema Inteligente para el Control de la Tuberculosis), SIEXCO-TB es el encargado de proveer información de los pacientes a las demás herramientas.

Estas son:

El SIM: un simulador genérico de difusión de la tuberculosis utilizado para soportar las hipótesis de difusión de la enfermedad.

El SIG, un sistema de información geográfico, cuyo objetivo es determinar el aumento o disminución de la tuberculosis, identificando los grupos de población que sufren la enfermedad y donde se localizan.

#### 4.3.2. La Jerarquía de Meta datos.

SIEXCO - TB es una herramienta que ha sido concebida como una primera aproximación a partir de la cual se pretende construir toda una familia de herramientas para el control de enfermedades de alta prevalencia. Es por esto que el sistema debe ser tan flexible y extensible como sea posible.

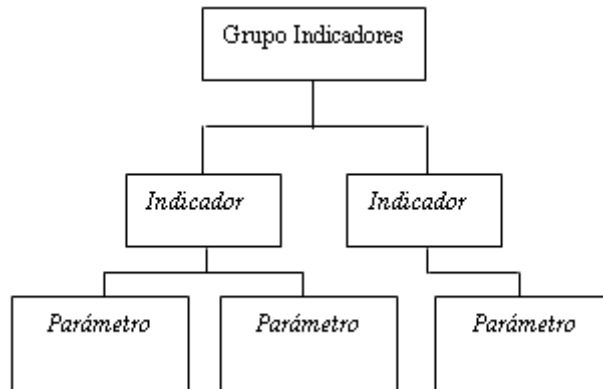


Figura 7 Jerarquía de Meta datos SIEXCO - TB

SIEXCO – TB trabaja con una meta modelo constituido por indicadores y parámetros, los cuales son necesarios para extraer del usuario la información necesaria para que el sistema experto genere sus conclusiones. Los indicadores y parámetros están organizados jerárquicamente. Así, los parámetros se agrupan formando indicadores, mientras que los indicadores forman grupos de indicadores, como se muestra en la figura 7.

Cada componente de la jerarquía contiene un nombre y una descripción, estos atributos son útiles para facilitar la comprensión y el mantenimiento de la estructura por los usuarios del sistema.

En cuanto a los Parámetros, estos son los componentes de la estructura que contienen las respuesta finales que dispararan los hechos que serán consultados en el proceso de inferencia. Estos parámetros toman dos estados, dependiendo si la respuesta es afirmativa ó negativa, según la información proporcionada por el usuario.

Durante una consulta al sistema experto, el sistema extrae de la base de datos una estructura de datos previamente definida como la que se muestra en la figura 8, la cual es una instancia de la jerarquía de meta datos arriba explicada.

En la estructura mostrada en la figura 8, existen 8 *indicadores*, agrupados en un indicador de grupo, los cuales son: Esquema, Clasificación, Peso, Edad, BK, TiempoTratamiento, Contraindicaciones y Resistencia.

Cada uno de estos *indicadores* contiene un conjunto de *parámetros*, los cuales son los que toman los estados finales, de acuerdo a las características particulares del paciente. Por ejemplo, el indicador *Esquema* agrupa 5 parámetros: CasoNuevo, Recaída, ReingresoAbandono, Fracaso, CasoCrónico. En este caso solo uno de los parámetros estará seleccionado, según el tipo de paciente sobre el que se realizará la consulta.

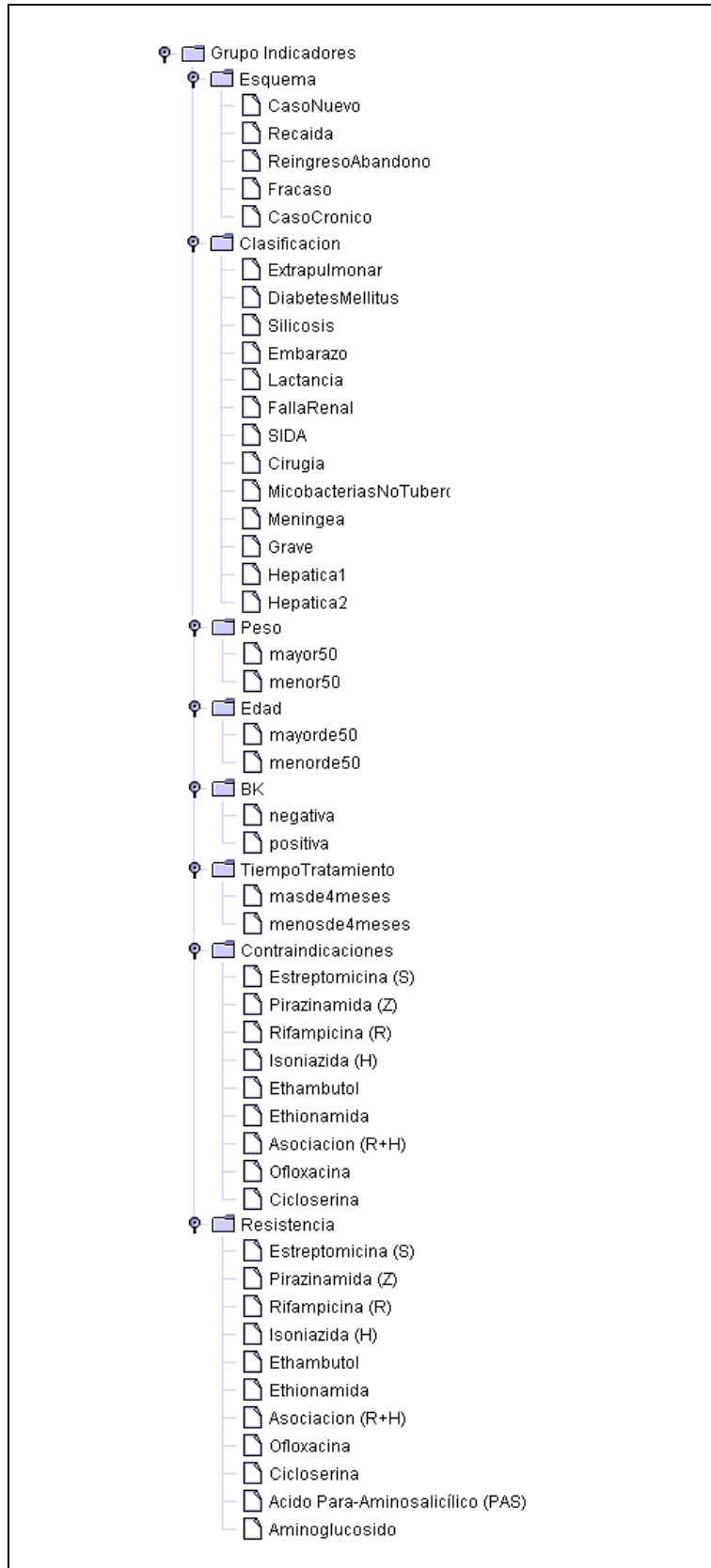


Figura 8 Grupo de Indicadores Básico usado por SIEXCO - TB

Ejemplo:

Veamos el caso de un paciente que reingresa después de haber abandonado el tratamiento (con baciloscopia de esputo positiva).

Este paciente tiene, además, las siguientes características:

Peso : 65 Kg.

Edad: 30 Años

Otras Enfermedades: Diabetes Mellitus, Falla renal.

Sensible a: Estreptomina.

Resistente a: Ethambutol, Isoniacida

De acuerdo a esta información, la cual es requerida por el sistema durante el proceso de consulta, quedarían seleccionados algunos parámetros de los indicadores así;

Esquema: ReingresoAbandono

Clasificación: Diabetes Mellitus y Falla Renal

Peso: Mayorde50

Edad: Menor50

BK: Ningún parámetro seleccionado

TiempoTratamiento: Ningún parámetro seleccionado

Contraindicaciones: Estreptomina

Resistencia: Ethambutol, Isoniacida.

En el ejemplo anterior, podemos observar que existen Indicadores cuyos parámetros son excluyentes, este es el caso del indicador Peso, donde solo uno de sus parámetros puede estar seleccionado (Mayorde50 ó Menorde50 ).

De la misma forma existen indicadores en los cuales más de un parámetro puede estar seleccionado, como en el indicador *Resistencia*, donde uno o más de sus parámetros (enfermedades para las que el paciente presenta resistencia) puede estar seleccionado.

Otra característica de la estructura es que puede haber indicadores para los cuales no haya ningún parámetro seleccionado, ya sea por falta de información ó por que el sistema no los necesita para entregar los resultados correspondientes a la consulta realizada.



Estas características hacen que se puedan construir instancias de la estructura de meta datos de manera muy flexible de acuerdo a la información que el experto crea necesaria para hallar las soluciones de tratamiento correspondientes.

### **4.3.3 Definición del Conocimiento**

Una de las etapas más importantes en el desarrollo de un sistema experto es la definición del conocimiento que será utilizado para llegar a las conclusiones requeridas durante los procesos de consulta. SIEXCO – TB abordó esta etapa a través de las subetapas que se exponen a continuación:

#### **4.3.3.1 Identificación y selección del origen del conocimiento**

El origen principal del conocimiento manejado por SIEXCO – TB es la guía oficial para el control de la tuberculosis en la cual se establecen las actividades, intervenciones y procedimientos de obligatorio cumplimiento por todos los profesionales de la salud y las instituciones prestadoras de servicios en salud del territorio nacional, y con la cual se pretende unificar las acciones diagnósticas y los esquemas terapéuticos. Esta guía es el resultado de años de experiencia en el control de la Tuberculosis y es de estricto cumplimiento, por lo tanto es la fuente más importante a partir de la cual se debía extraer el conocimiento de SIEXCO – TB. Esta fuente fue expedida por el ministerio de protección social mediante resolución 0412 del 25 de febrero de 2000 y es de acceso público, lo que facilita su consulta.

A pesar de que un porcentaje importante de los procedimientos a seguir para el tratamiento de la tuberculosis están estipulados en la guía, existe otra parte que no se hace explícita en ella, por tal razón, era de vital importancia incorporar al equipo de trabajo a un médico experto que proporcionara la información no contenida en ese documento. Desde los inicios del proyecto y hasta muy avanzado este, se contó con el apoyo del Dr. Alfonso Tenorio coordinador del Grupo de Estudio de Tuberculosis del Cauca desde 1997. Posteriormente, debido a su incorporación como asesor en Tuberculosis de la OPS en Washington, se continuó con el trabajo, ahora con la ayuda de la Dr. Regina Plaza de la dirección departamental de salud del Cauca hasta el final del proyecto. Afortunadamente, ambos médicos prestaron su colaboración de manera oportuna y útil al proyecto.

Es así como la combinación de estas dos fuentes hizo posible la construcción de la base de conocimiento de SIEXCO – TB.

#### **4.3.3.2 Adquisición, análisis y extracción del conocimiento**

Para esta etapa se cumplió con las siguientes tareas:

##### *Definición de la estrategia de adquisición del conocimiento*

La estrategia de adquisición del conocimiento de SIEXCO – TB consistió primero en la lectura y correcta interpretación de la información contenida en la guía de Tuberculosis, esto se logró con la ayuda del médico experto. Una vez resueltas las dudas surgidas inicialmente se identificaron algunos de los elementos del conocimiento, principalmente aquellos que tenían que ver con los esquemas básicos de tratamiento.

Posteriormente, con el permanente apoyo del médico experto se identificaron el resto de elementos de conocimiento, aquellos que describían casos muy específicos no contenidos en la guía. Este proceso se llevo a cabo durante casi todo el ciclo de vida del proyecto con el objetivo de depurar incrementalmente la base de conocimiento, hasta obtener la versión definitiva.

##### *Clasificación del Conocimiento*

Una vez obtenido el conocimiento se procedió a clasificarlo. Esto con el objetivo de hacer una base de conocimiento modular, eficiente y de fácil mantenimiento. Para esto se dividieron las reglas en reglas generales aplicables a todos los tratamientos y reglas particulares aplicables a cada tratamiento específico. A su vez, teniendo en cuenta las tareas a realizar por cada regla se agruparon en reglas de definición del tratamiento, reglas de modificación del tratamiento y reglas de acciones a realizar. Para más detalle de la clasificación de las reglas ver sección 4.3.4 de este documento.

#### **4.3.3.3 Verificación del conocimiento y Evaluación del Sistema**

Después de implementar el sistema se procedió a verificar el conocimiento. En esta etapa se realizaron las pruebas, analizando los resultados obtenidos y realizando los correctivos necesarios para garantizar el buen funcionamiento.

Mediante el proceso de análisis de las prueba se observaron los resultados buscando respuestas incorrectas, incompletas e incongruentes dadas por el sistema. Una vez encontrado un problema en los resultados obtenidos se buscaba su fuente y se procedía a corregirlo, validando la corrección a través de la ejecución de la prueba de nuevo.

Este proceso se realizó primero internamente, por los desarrolladores de la aplicación y luego con el apoyo del médico experto quien realizó pruebas a través de la definición de casos de ejemplo y la comprobación de los resultados obtenidos.

En la etapa final del ciclo de vida del proyecto se procedió a su evaluación. Es esta etapa intervinieron, tanto el médico experto asociado al proyecto, como personal de enfermeras de la dirección departamental de salud del Cauca. El médico experto, quien ya había participado en la etapa de verificación, cumplió ahora el rol de evaluador del sistema en lo que tiene que ver con el módulo experto. De esta forma a través de la misma metodología de definición de casos de pacientes válido la herramienta. Por otra parte el grupo de enfermeras realizó la validación del correcto funcionamiento de la parte administrativa del sistema, relacionada principalmente con la generación de los diferentes informes del programa de atención de la tuberculosis y la tarjeta individual de pacientes.

#### **4.3.4 Representación del Conocimiento**

En SIEXCO- TB el conocimiento esta representado mediante reglas de producción, las cuales están clasificadas en reglas generales y reglas particulares de cada tratamiento. Esto con el objetivo de hacer el sistema modular, sencillo y eficiente. De esta manera la adquisición de nuevo conocimiento, así como su mantenimiento se simplifica de manera notable.

Las reglas generales, las cuales forman un grupo de reglas independientes, son aquellas que se aplican sin importar el tratamiento básico asignado a la consulta, mientras que las reglas particulares son aquellas que son solo aplicables a un tratamiento específico y son consultadas solo para dicho tratamiento.

De otro lado, las reglas se dividen en:

- Reglas de Definición del Tratamiento.
- Reglas de Modificación del Tratamiento
- Reglas de Acciones a realizar.

#### **4.3.4.1 Reglas de Definición del Tratamiento**

Al principio de una consulta a la base de conocimiento, el sistema solicita del usuario información básica. Con esta información está en capacidad de generar un tratamiento básico, el cual sufrirá modificaciones de acuerdo a características más específicas del paciente.

Las reglas de definición del tratamiento son de la siguiente forma:

**SI** el paciente tiene menos de 15 años **Y** es un caso nuevo **ENTONCES** administrar el tratamiento acortado supervisado para TBC Infantil.

Cuando el sistema detecta que se ha disparado una regla de definición del tratamiento, consulta la base de datos (base de datos relacional) para extraer un tratamiento con sus fases, duración y medicamentos a suministrar, datos que se conservarán si el paciente no tiene otras características especiales que modifiquen el tratamiento básico.

#### **4.3.4.2 Reglas de Modificación del Tratamiento**

Existen características especiales del paciente que hacen necesario modificar el tratamiento básico obtenido desde la base de datos del sistema. Estas características, se relacionan, por ejemplo, con otras enfermedades que el paciente padece, ó reacciones negativas a algún medicamento (sensibilidad y/o resistencia).

Las reglas de modificación del tratamiento son de la siguiente forma:

SI hay sensibilidad a la Estreptomina **ENTONCES** cambiar Estreptomina en la fase 1 por Ethambutol a dosis de 3 Tabletas por 400 mg.

Como se ve , las reglas de modificación del tratamiento, son un poco más complejas que las de simple definición. Por ejemplo, la regla anterior incluye tres cambios en un medicamento de la fase 1: el nombre del medicamento, la presentación y la dosis.

Dentro de los posibles cambios que se pueden hacer con las reglas de modificación del tratamiento están:

Modificar un medicamento de la fase 1 ó 2 (dosis, presentación, nombre)

Modificar la duración de la fase 1 ó 2

Modificar el número de dosis de la fase 1 ó 2

Adicionar un medicamento a la fase 1 ó 2

Eliminar un medicamento de la fase 1 ó 2

#### 4.3.4.3 Reglas de Acciones a realizar

Para el tratamiento de la tuberculosis, existen algunos procedimientos, actividades y recomendaciones que no se pueden expresar en términos de los componentes de un esquema de tratamiento. Esta información, que no afecta directamente la información del tratamiento cargada desde la base de datos es manejada por SIEXCO-TB como *acciones a realizar*.

Las reglas de acciones a realizar son de la siguiente forma:

SI el tipo de tuberculosis es extrapulmonar **ENTONCES** el diagnóstico se comprueba con cultivo de secreción o biopsia de tejido afectado.

Este tipo de información no se utiliza para la creación de la fórmula médica final y más bien es usada para dar algunas indicaciones al paciente.

## 4.4 DETALLES DE IMPLEMENTACIÓN

A nivel de implementación, SIEXCO-TB además de estar soportado en el paradigma de sistemas expertos para su construcción, se hace fuerte gracias al uso de dos tecnología ampliamente difundidas gracias a la larga lista de características que ofrecen. Estas son JAVA y XML. Con estas tecnologías se logran dos objetivos fundamentales: Código portable y datos portables.

### 4.4.1 Java

JAVA es un lenguaje de gran aceptación mundial gracias a sus características: Es un lenguaje simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutral, y portable [21].

#### 4.4.1.1 Características de Java

##### **Simple**

Aprovechando la popularidad de C++, Java fue creado con una sintaxis muy similar, con el propósito de eliminar problemas de comprensión del lenguaje.

Java omite algunas características de C++ tales como sobrecarga de operadores, herencia múltiple y punteros. La gestión de memoria se realiza por medio de un recolector de basura (garbage collector), lo que elimina muchos de los problemas tradicionales de C++ que exigía técnicas de programación rigurosas.

##### **Orientado a Objetos**

Java fue diseñado como un lenguaje orientado a objetos desde el principio y por lo tanto tiene todas las características de este paradigma, lo que facilita el diseño de código modular y reusable.

## **Distribuido**

Java provee los mecanismos necesarios para construir aplicaciones en red, manejando protocolos TCP/IP (Tales como HTTP y FTP).

## **Robusto**

Java fue diseñado para crear programas confiables, para ello hace comprobación temprana de errores (en tiempo de compilación) y comprobación dinámica en tiempo de ejecución.

El manejo de memoria realizado por Java es muy simple: No se definen punteros explícitamente y provee un recolector de basura automático, permitiendo que el programador se despreocupe de este tema.

## **Seguro**

Java fue diseñado para ser usado en ambientes distribuidos, por lo tanto se hizo mucho énfasis en la seguridad. Provee técnicas de autenticación basadas en encriptación.

## **Arquitectura Neutral**

Para que las aplicaciones Java se puedan ejecutar sobre cualquier plataforma Hardware y Software, el compilador genera un formato de archivos de objetos de arquitectura neutral (bytecode), el código compilado es ejecutable en variedad de procesadores gracias a la existencia del *Java Runtime System*.

La arquitectura neutral dada por los “bytecodes” es lo que hace que las aplicaciones Java sean portables. El programa intérprete de este código en cada dispositivo es la llamada máquina virtual de Java (Java Virtual Machine, JVM).

## **Portable**

El ser de arquitectura neutral, representa sólo una parte de su portabilidad [21]. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.

## **Interpretado**

El intérprete de Java (La máquina virtual JVM) puede ejecutar los bycodes directamente sobre cualquier máquina que lo contenga.

## **MultiThread**

El multihilado es una forma de construir aplicaciones que corren múltiples hilos, ejecutándose concurrentemente y llevando a cabo tareas distintas. Para esto Java tiene un sofisticado conjunto de primitivas de sincronización.

### **4.4.1.2 Uso de Java en SIEXCO – TB**

En la implementación de SIEXCO –TB, se aprovecho ampliamente muchas de las características ofrecidas por Java al emplear una variedad de APIs de la edición estándar y otros adicionales que permitieron dotar de la funcionalidad requerida al sistema. En esta sección se describen dichos APIs, así como su aporte a SIEXCO –TB.

#### **4.4.1.2.1 JavaBeans**

Los JavaBeans son un modelo de componentes software de plataforma independiente y portable. Esto le permite a los desarrolladores construir componentes reusables una sola vez y correrlos en cualquier máquina, beneficiándose de la portabilidad de Java.

Los Beans son clases de Java que pueden ser manipuladas desde un entorno visual. Cualquier clase de Java que cumple con ciertas propiedades y convenciones puede ser un Bean. En principio, un bean puede considerarse frecuentemente como un elemento con una interfaz de usuario visible, aunque es posible que tal interfaz no exista y el bean ofrezca su funcionalidad sin ningún comportamiento visual [22].



Las principales propiedades de los Beans son [22,23]:

**Propiedades.** Son las características de un Bean que pueden ser modificadas durante el diseño. Las propiedades se exponen a las herramientas de desarrollo mediante una clase llamada BeanInfo.

**Personalización.** Mediante la herramienta de ensamblaje, se puede personalizar el aspecto y/o comportamiento de un bean, alterando los valores de sus propiedades.

**Eventos.** Por medio de los eventos los Beans se comunican entre sí. Estos pueden anunciarse como generadores o receptores de ciertos tipos de eventos. Las herramientas de desarrollo pueden examinar un Bean para determinar que eventos puede disparar y cuales puede manejar.

**Persistencia.** Los Beans pueden guardar su estado, y restaurarlo posteriormente. Para soportar esta característica utilizan la serialización de objetos Java.

**Introspección.** Las herramientas de ensamblaje pueden consultar al bean para saber qué propiedades, eventos y métodos ofrece.

SIEXCO – TB utiliza Beans para manejar la información relevante del sistema. Estos Beans permiten implementar la Jerarquía de Meta datos de la que se habla en la sección 4.3.2., mediante la definición de cada elemento (nodo) de la jerarquía como un JavaBean.

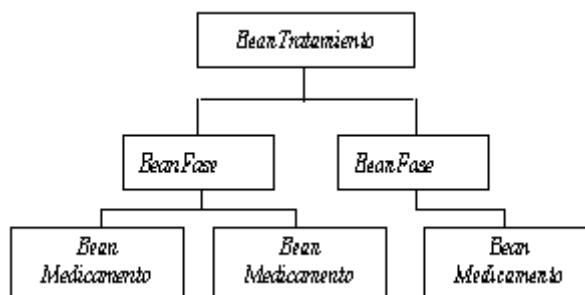


Figura 9 Meta modelo del tratamiento

Siguiendo este criterio de diseño se maneja información como aquella relacionada con pacientes y esquemas de tratamiento. Esta información en varios casos es a su vez serializada a documentos XML para su exportación ó almacenamiento en la base de datos relacional.

Las principales clases del sistema, implementadas como JavaBeans son:

#### *BeanGrupoPaciente*

Las instancias de esta clase representan grupos de indicadores, los cuales representan la información que el sistema solicita para generar el tratamiento que va a ser cargado en el BeanTratamiento. Cada BeanGrupoPaciente agrupa objetos BeanIndicador que contienen objetos BeanParametro. Así, cada parámetro contiene la información final acerca de características del paciente (edad, peso, resultado de las baciloscopias de control, etc). Cada objeto de la jerarquía contiene un nombre y una descripción, en caso de que el usuario desee conocer mas detalles acerca de la información solicitada por el sistema.

#### *BeanTratamiento*

Las instancias de esta clase representan un tratamiento determinado según un esquema específico. Este a su vez contiene objetos BeanFase representando las fases de cada tratamiento. Finalmente, cada BeanFase se conforma de objetos BeanMedicamento que almacenan información tal como el nombre del medicamento, las presentación comercial y la dosis asociada al tratamiento.

#### *BeanRepositorio*

Haciendo uso del concepto de repositorio, las instancias de esta clase, relacionan un BeanGrupoPaciente con una base del conocimiento específica. Además contiene una descripción del repositorio e información acerca de experto que lo creo.

#### *BeanFormulaMédica*

El resultado final que ofrece el sistema experto al médico que realiza la consulta de un tratamiento para un paciente determinado es la fórmula médica.

Dicha formula médica contiene información acerca de los medicamentos que se le deben suministrar al paciente, con sus dosis correspondientes, además de información relacionada con el organismo prestador del servicio de salud, el paciente y el médico tratante. Toda esta información está contenida en el *BeanFormulaMedica*, lo que facilita su posterior manipulación.

### *BeanConector*

SIEXCO – TB utiliza este Bean para realizar las funciones de actualización y lectura de la base de datos relacional.

### *BeanMensajes*

Utilizado para manejar los diferentes tipos de mensajes que el sistema muestra al usuario en todo el proceso de interacción

#### **4.4.1.2.2 JDOM**

JDOM es un API de fuente abierta para parsing , creación, manipulación y serialización de documentos XML. JDOM fue diseñado puramente para Java y XML, lo que lo hace simple y fácil de usar por desarrolladores JAVA. Para mayor información acerca de JDOM remitirse a la sección 4.4.2.2. de este documento.

SIEXCO-TB hace uso de JDOM para manejar cierta información del sistema mediante documentos XML.

Manejar información sobre tratamientos, tarjetas individuales, usuarios del sistema, entre otras, en la forma de documentos XML trae consigo algunas ventajas. Principalmente esto contribuye a que el diseño de la base de datos sea muy simple, ya que, por ejemplo, toda la información de una tarjeta individual se almacena como un solo campo de tipo TEXT, mientras que si se hace de la forma convencional se aumentaría de manera considerable el número de tablas y campos que deberían tener para soportar la cantidad de información contenida en cada una de estas tarjetas.

Con el método utilizado por SIEXCO – TB, la base de datos se ve reducida a pocas tablas con pocos campos. Así, la tabla que contiene la información de las tarjetas individuales (por ejemplo), solo contiene un campo donde está almacenada la tarjeta individual y algunos campos adicionales, necesarios para almacenar la clave primaria de la tabla y las claves foráneas para su relación con otras tablas de la base de datos.

Otro aspecto interesante de manejar la información como documentos XML es que se facilitan los procesos envío de información vía correo electrónico, importación y exportación de archivos. Esto de especial utilidad en SIEXCO – TB ya que el transporte de información, tal como informes trimestrales, análisis de cohorte, tratamientos, reglas de la base de conocimiento, tarjetas individuales, fórmulas médicas, entre otros, es fundamental para llevar a cabo mucha de las actividades contempladas en el programa de atención de la tuberculosis.

SIEXCO – TB interactúa con otras aplicaciones suministrándoles información, de esta manera definir dicha información en un formato estándar y difundido como XML, promueve la compatibilidad de las aplicaciones facilitando procesos de comunicación y acceso a información compartida.

#### 4.4.1.2.3 Java Foundation Classes (JFC/Swing)

JFC [24] constituye un conjunto de librerías de clases Java suministradas con la edición estándar de Java (J2SE) para soportar la construcción de interfaces gráficas de usuario que corren sobre diferentes plataformas tales como Microsoft Windows, Linux, y Mac OSX.

JFC contiene:

*AWT (Abstract Window Toolkit).* Es un API que permite a los programas integrarse en el sistema de ventanas nativo, incluyendo APIs para Drag and Drop (Arrastrar y soltar) entre aplicaciones Java.

*Java 2D.* Permite a los desarrolladores incorporar fácilmente gráficos 2D de alta calidad, texto, e imágenes en aplicaciones y applets Java.

*Componentes Swing.* Extiende AWT para proveer una librería de componentes GUI rica y extensible.

*Accesibilidad.* Permite tecnologías asistivas como lectores de pantalla obtener información desde la interfase de usuario.

*Internacionalización.* JFC incluye soporte para crear aplicaciones que interactúan con los usuarios, usando su propio lenguaje y convenciones culturales.

## JAVA SWING

Es una librería de componentes para la construcción de interfaces gráficas de usuario. El paquete javax.swing de la plataforma J2SE usa la funcionalidad de AWT y Java 2D para proveer componentes de Interfaces gráficas sofisticados y extensibles que cumple con la especificación del modelo de JavaBeans.

Los componentes de Swing están categorizados como:

Controles atómicos: JLabel, JButton, JComboBox, etc.

Componentes de datos Complejos: Jtable, Jtree, Jlist, etc.

Componentes de Texto: JTextField, JOptionPane, JPasswordField, etc.

Componentes de Menú: JMenuBar, Jmenu, JMenuItem, etc.

Contenedores Layout: Jpane, JscrollPane, Jtoolbar, etc.

Componentes de ventanas: Jwindow, JFrame, JinternalFrame, etc.

Los componentes de Swing usan otras características de JFC, tales como drag and drop, los APIs de accesibilidad, y el soporte para internacionalización de J2SE creando una amplia funcionalidad para la implementación de interfaces gráficas.

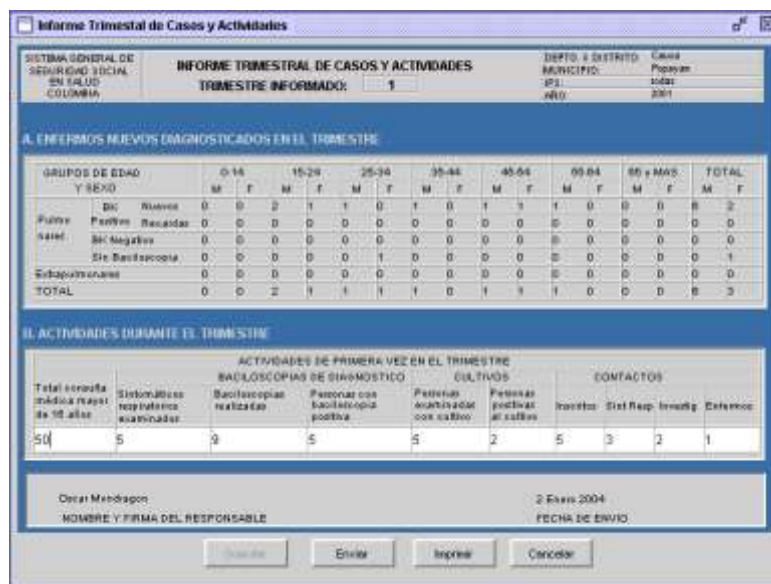


Figura 10 GUI Mostrada para el Informe Trimestral de Casos y Actividades

SIEXCO – TB aprovecha muchas de las características de Swing, utilizando todas las categorías de componentes swing para obtener una variedad de interfaces de usuario sofisticadas que

hacen que la interacción del usuario con el sistema sea muy agradable. Abajo se muestran algunos ejemplos de las interfaces gráficas de usuario ofrecidas por SIEXCO-TB.

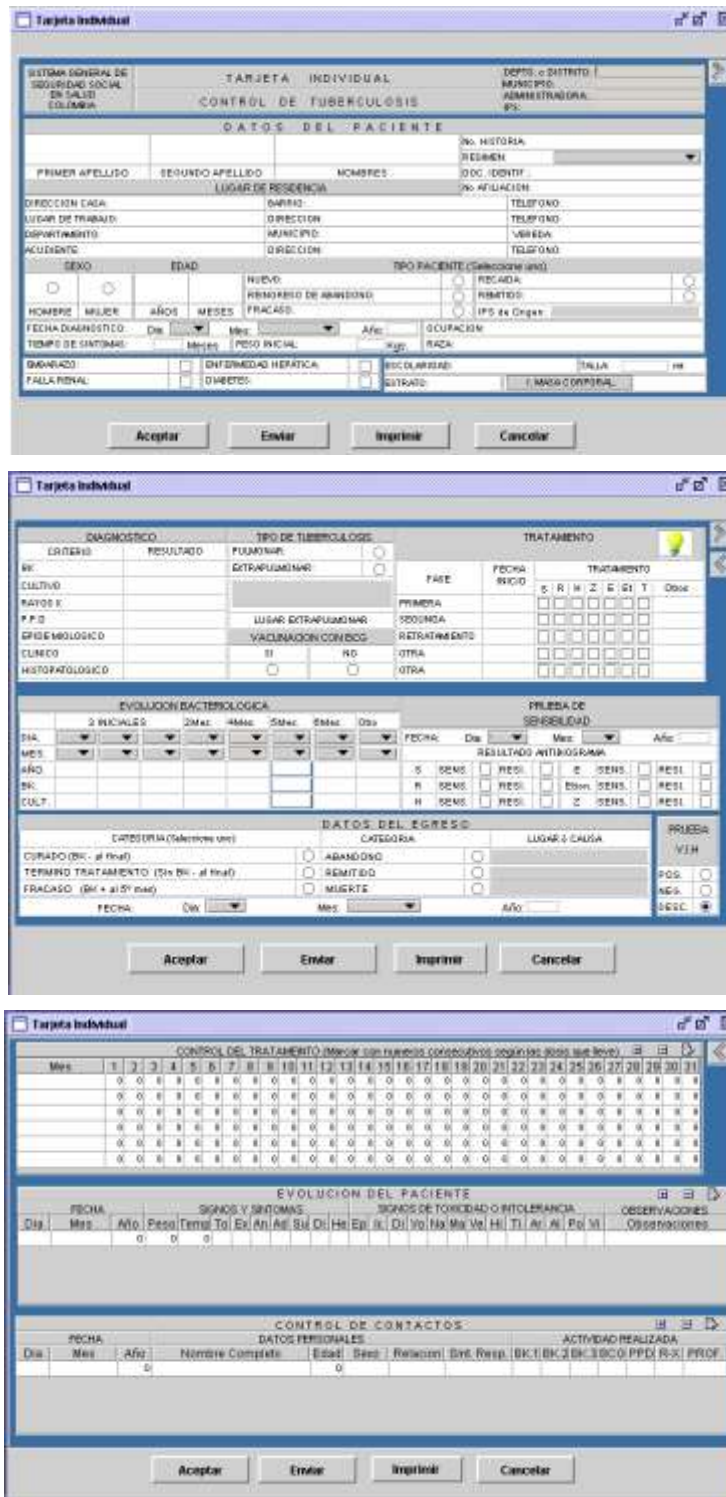


Figura 11 GUIs usadas para desplegar la tarjeta individual del paciente

#### 4.4.1.2.4 JavaMail

El API JavaMail es un API diseñado para adicionar, fácilmente, capacidades de correo electrónico a cualquier aplicación en Java. Incluye clases que encapsulan funciones comunes de correo y protocolos. JavaMail interactúa de manera sencilla con otros APIs de Java para facilitar su uso en diferentes aplicaciones.

JavaMail es usado para construir aplicaciones cliente de correo electrónico similares a Eudora, Pine y Microsoft Outlook.

SIEXCO-TB hace uso de Java Mail, para enviar correos electrónicos con archivos que contienen ya sea una tarjeta individual, un informe trimestral ó un análisis de cohorte como archivos adjuntos. Esto facilita el proceso de entrega de dichos informes entre los diferentes organismos involucrados en el programa de control de tuberculosis.

En la figura 12 se muestra la interfaz gráfica de usuario desplegada por SIEXCO – TB cuando se desea enviar algún tipo de información vía correo electrónico.

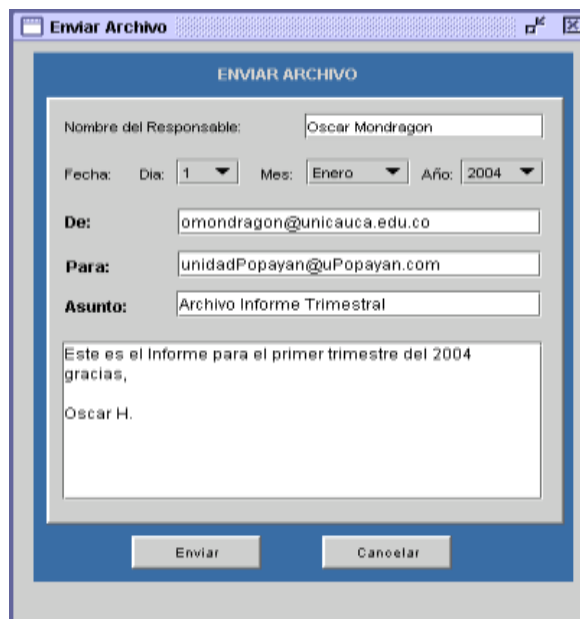


Figura 12 GUI para enviar vía correo electrónico los archivos generados por SIEXCO- TB

#### 4.4.1.2.5 JDBC

El API JDBC [25] es el estándar de la industria para conectividad entre el lenguaje de programación Java y un amplio rango de bases de datos. El API JDBC provee acceso basado en SQL a bases de datos.

Este API provee 3 funcionalidades básicas:

- Establecer una conexión con una base de datos o acceder a cualquier fuente de datos tabular.
- Enviar sentencias SQL
- Procesar el resultado

SIEXCO – TB usa JDBC para el manejo de la base de datos relacional en Mysql y para la alimentación de la base de datos de estadísticas que alimenta a Epi Info para realizar los diferentes análisis de la información manejada por el sistema.

Para la comunicación con la base de datos de SIEXCO – TB (La construida en Mysql), se utilizó un driver JDBC puro tipo 4, mientras que para la comunicación con la base de datos de Epi Info, la cual está construida en Microsoft Acces, se utilizo un driver JDBC-ODBC tipo 1, el cual es el único aceptado por este tipo de bases de datos.

La interacción con las bases de datos mencionadas, se logra a través de un JavaBean llamado *BeanConector*. Este Bean usa JDBC para realizar todas las funciones de actualización y búsqueda en la base de datos.

#### 4.4.1.2.6 JavaHelp

JavaHelp es un API para la construcción de ayudas en JAVA. Provee funcionalidades para acceso, navegación y búsqueda de información en dichas ayudas. Las ayudas construidas con JavaHelp tienen: Una tabla de contenido, un índice, y un mecanismo de búsqueda para fácil navegación sobre el contenido de las ayudas. Además provee un conjunto de palabras claves para búsqueda rápida de contenidos.

JavaHelp permite además imprimir el contenido de las ayudas desplegadas.



SIEXCO – TB hace uso de JavaHelp para:

- Mostrar las ayudas de operación del sistema, útiles para los todos los tipos de usuarios de SIEXCO -TB.
- Mostrar ayudas de diligenciamiento de la tarjeta individual y los informes.
- Mostrar la guía completa para el control de la tuberculosis expedida por el ministerio de protección social.

En la figura 13 se muestra una de las ayudas construidas con JavaHelp, para SIEXCO-TB.

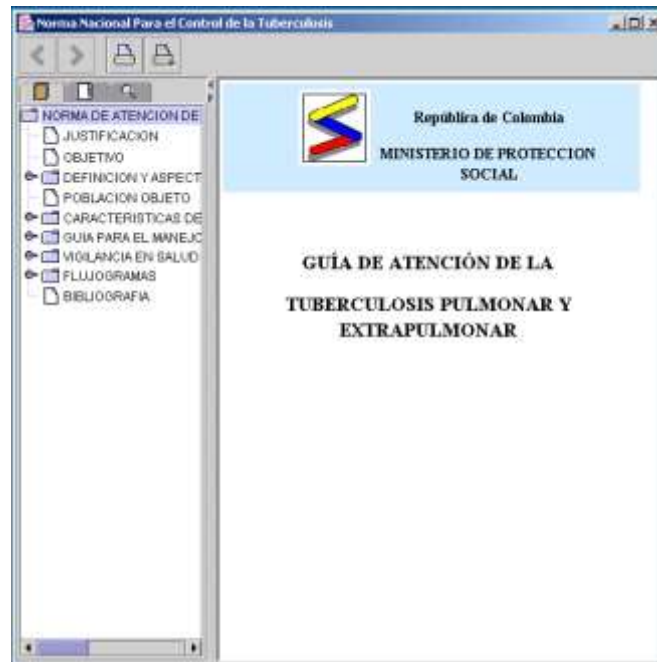


Figura 13 Ayudas de SIEXCO - TB

#### 4.4.1.2.7 Mandarax

Es una librería JAVA de fuente abierta para el desarrollo de sistemas expertos basados en reglas de producción. El módulo experto definido en la arquitectura de SIEXCO –TB, hace uso de Mandarax para la representación, edición y queryng de las reglas. En la sección 4.5.2. se profundiza acerca de esta API y su empleo en SIEXCO-TB

#### 4.4.2 XML (eXtensible Markup Language)

XML es un estándar que especifica un conjunto de reglas para definir etiquetas para organizar un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

XML provee las bases para una variedad de lenguajes. Ejemplos de estos son MathML (Matemactical Markup Language), ebXML (Electronic Business XML), y VXML (Voice Markup Language ). Esto se muestra en la figura 18.

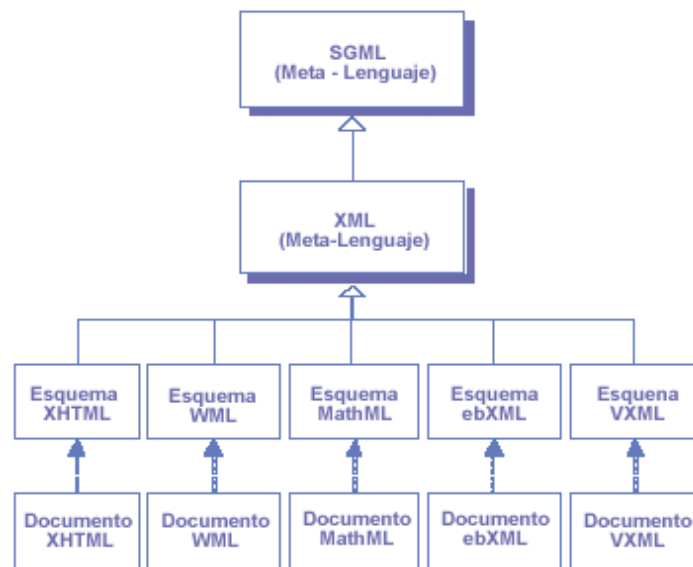


Figura 14 Jerarquía de Lenguajes XML

La flexibilidad con la que XML representa los datos permite el fácil envío de información entre aplicaciones, y la transformar los datos de un formato a otro. Hoy en día XML es el estándar más utilizado por las empresas para intercambiar datos. La información manejada en los e-business tal como precios, listas de productos, transacciones, es representada en XML y transferida a través de Internet usando protocolos y estándares abiertos.

Cada lenguaje XML define su propia gramática: Un conjunto específico de reglas controlan el contenido y la estructura de los documentos escritos en ese lenguaje. Por ejemplo el elemento precio podría ser válido en un documento ebXML pero no tener significado en un documento MathML. Cualquier *parser* puede validar la estructura de cualquier documento XML dadas las reglas de su lenguaje.

Usar XML como una base común para lenguajes de más alto nivel facilita el intercambio de datos entre componentes, sistemas y empresas. Las herramientas de *parsing* y traducción construidas para manejar cualquier tipo de datos XML pueden emplearse para crear y manipular datos en forma uniforme sin importar el lenguaje específico.

Una infraestructura basada en XML permite altos niveles de reuso de componentes e interoperabilidad de aplicaciones. Además, con XML se logran interfaces simples que facilitan su mantenimiento y extensión. Ya que XML es un estándar de la industria, puede ser utilizado por las aplicaciones sin preocuparse de la dependencia de un vendedor específico.

#### 4.4.2.1 APIs de Java para XML

Para evitar el desarrollo de APIs Java para XML aislados no compatibles Sun Microsystem está coordinando los diferentes esfuerzos en este tema. De esta manera los desarrolladores están estandarizando y simplificando los diferentes APIs de Java para XML.

Los APIs de Java para XML (JAX) constituyen actualmente una familia de especificaciones, la cual se muestra en la tabla 4.

APIs Java para XML	Descripción Funcional
Java Api for XML parsing <b>JAXP</b>	Provee una interfaz común para el acceso a parsers XML y procesadores XSLT
Java Document Object Model <b>JDOM</b>	Provee una implementación Java del framework DOM
Java Api for Binding <b>JAXB</b>	Permite el mapeo XML de objetos java almacenados en formato XML
Long Term Java – Beans Persistence	Es similar a JAXB y permite la serialización XML de JavaBeans
Java API for XML messaging <b>JAXM</b>	Permite el uso de mensajería SOAP en aplicaciones Java
<b>JAX-RPC</b>	Es un API de Java para realizar llamadas a procedimientos remotos usando XML
Java API for XML repositories <b>JAXR</b>	Provee acceso repositorios XML tales como ebXML y UDDI

Tabla 4 APIs de Java para XML

#### 4.4.2.2 Uso de XML en SIEXCO-TB

En SIEXCO – TB el uso XML para representar cierta información del sistema es el sustento fundamental para soportar algunas de sus características, tales como:

- La portabilidad de la información
- La compatibilidad con otras herramientas (en especial las demás herramientas SINCO-TB)
- La facilidad de exportación / importación de información de pacientes y de tratamientos e informes del sistema.
- La simplicidad de la base de datos relacional.

#### DOCUMENTOS XML

Para soportar cada una de estas características, SIEXCO –TB define una variedad de documentos XML, para facilitar la compatibilidad a nivel de datos con cualquier aplicación (tanto Java como no Java). Dichos documentos son:

##### *La tarjeta Individual del Paciente*

Este es el archivo XML más extenso manejado por SIEXCO – TB. En este documento se estructura toda la información de la tarjeta individual del paciente. Dicha tarjeta contiene:

- Información general de la institución prestadora del servicio
- Datos personales del paciente
- Datos de diagnóstico
- Evolución bacteriológica del paciente
- Datos de Egreso
- Datos del Control del Tratamiento
- Efectos colaterales producidos por los medicamentos
- Control de Contactos

##### *Análisis de Cohorte*

Este documento XML contiene indicadores epidemiológicos y operativos del programa. Incluye:

- Información sobre la negativización de las baciloscopias al segundo mes de la cohorte a evaluar.
- los resultados del tratamiento en la cohorte de casos con BK positivo.

#### *Informe Trimestral de Casos y Actividades*

Maneja la siguiente información:

- Los enfermos nuevos diagnosticados en el trimestre, clasificados en: Casos pulmonares y extrapulmonares, nuevos y recaídas, BK positivos y BK negativos.
- Las actividades desarrolladas durante el trimestre: Sintomáticos respiratorios examinados, baciloscopias realizadas, personas positivas al cultivo, etc.

#### *Grupo de Indicadores*

Este documento contiene un *JavaBean* serializado con información sobre los diferentes indicadores y parámetros usados para obtener información del paciente útil al sistema experto.

#### *Fórmula Médica*

Es un *JavaBean* serializado que contiene la siguiente información:

- Medicamentos que se le deben suministrar al paciente, con sus dosis correspondientes
- información relacionada con el organismo prestador del servicio de salud
- Información del paciente
- Información del médico tratante

#### *Repositorio*

Este XML relaciona un grupo de indicadores con una base del conocimiento específica. Además contiene una descripción del repositorio e información acerca de experto que lo creo.

### *Tratamiento*

Es un *JavaBean* serializado y contiene:

- Objetos *BeanFase* serializados representando las fases de cada tratamiento.
- Cada *BeanFase* se conforma de objetos *BeanMedicamento* serializados que almacenan información tal como el nombre del medicamento, la presentación comercial y la dosis asociada al tratamiento.

### *Usuario*

Este documento XML contiene la siguiente información de los usuarios del sistema:

- Nombre
- Apellidos
- Teléfono
- Dirección
- Tipo de usuario
- Login

### *Configuración de la Base de datos*

Este XML contiene la siguiente información de acceso al servidor de bases de datos:

- Dirección IP del servidor
- Login de la base de datos
- Password

### *Reglas de la Base de Conocimiento*

Son archivos XKB que contienen información sobre las reglas usadas por el módulo experto para obtener los diferentes esquemas del tratamiento. Para más información de estos archivos dirigirse a la sección 4.5.2.7 de este documento.

## SIEXCO – TB usa JDOM

Para el manejo de cierto tipo de información, tales como tarjetas individuales, informes del sistema, archivos de configuración, entre otros, SIEXCO – TB manipula estructuras XML simples. Los APIs Java que podrían utilizar SIEXCO - TB para la interacción con documentos XML son JAXP y JDOM.

JAXP y JDOM ofrecen un modelo similar. Ya que JDOM se construye sobre la arquitectura de JAXP. Sin embargo, JDOM no depende de esta arquitectura, permitiendo el uso de otras opciones de proveedores para el parsing de documentos.

Esta característica hace que JDOM sea una mejor opción que JAXP, además de presentar otras propiedades interesantes tales como:

- Es de libre distribución.
- Lectura / escritura eficiente y sencilla de documentos XML. JDOM.
- Aprovecha las características de Java al hacer uso de clases de Java ya familiares a los programadores para manipular la información extraída desde un XML.

SIEXCO – TB usa el método SAX soportado por JDOM, ya que consume menos recursos que DOM, es más rápido y facilita la manipulación de documentos grandes, tales como la tarjeta individual del paciente ó el informe trimestral de casos y actividades.

## 4.5 HERRAMIENTAS DE DESARROLLO

En esta sesión se pretende realizar una descripción de las diferentes herramientas de desarrollo utilizadas en la implementación del proyecto SIEXCO-TB.

### 4.5.1 Oryx

Oryx [17] es una herramienta desarrollada para proporcionar un entorno grafico que permita el diseño y mantenimiento de bases de conocimiento implementadas en Mandarax. El editor de reglas esta diseñado para proporciona una interfaz grafica agradable y de fácil uso que puede

ser utilizada por los usuarios finales sin mayores complicaciones. Oryx es un producto de fuente abierta con dos tipos de licencia (GPL y Comercial).

Oryx es una aplicación java standalone que tiene dos componentes principales, los cuales soportan dos herramientas importantes en el ciclo de vida para el desarrollo de bases de conocimientos basadas en reglas:

- **El editor de bases de conocimientos:** Es una interfaz grafica diseñada para ser utilizada por el experto del dominio, es decir alguien que tiene la experiencia y conocimiento necesarios para alimentarla. El objetivo principal del editor de bases de conocimiento es proporcionar al experto una herramienta no técnica para la representación del conocimiento, en la que las reglas, los hechos y otros objetos se representan como oraciones y secuencias del lenguaje común.
- **Repositorio:** En el repositorio se representa la estructura y los meta datos de una base de conocimientos. El repositorio contiene además la descripción de los predicados, funciones, conexiones de base de datos y otros elementos de una base de conocimientos. En el repositorio también se encuentra la descripción de la representación de las reglas y de los hechos. El mantener actualizado el repositorio es obligación del administrador de repositorio, quien conoce la estructura de la base de conocimientos, el modelo de objetos y la base de datos de su organización. El administrador no necesita ser un experto en programación.

Oryx trabaja con formatos basados en XML, dando soporte al estándar RuleML y a los formatos de Mandarax XKB y ZKB.

#### 4.5.2 Mandarax

Mandarax [18] es una librería java de fuente abierta que facilita el desarrollo de sistemas expertos en los que el conocimiento se encuentra representado por reglas de producción. Este incluye la representación, intercambio, manejo y procesamiento de la base de reglas. El objetivo principal de Mandarax es proporcionar una plataforma orientada a objetos para desarrollar sistemas basados en reglas de producción.



En términos generales una regla asocia uno o varios prerrequisitos con una conclusión. Algunas veces los prerrequisitos son llamados el cuerpo de la regla, mientras que la conclusión es llamada cabecera. Los prerrequisitos y las conclusiones son hechos los cuales a su vez están formados por términos y predicados que asocian los términos. Ej.: “*mayor de 50*” en este hecho “*mayor de*” es el predicado y “*50*” es el termino (En la sesión 4.3.3 se hace una amplia descripción de reglas de producción). Hablando en términos de orientación a objetos los términos son objetos y los predicados representan la relación entre ellos.

Mandarax utiliza tres tipos de términos:

- **Términos constantes:** los términos constantes son objetos concretos como “5%”.
- **Términos variables:** Un ejemplo de termino variable puede ser un “*cliente*”. Este tipo de término puede ser remplazado por un término constante si es necesario, esta es una característica importante por que es posible utilizar las mismas reglas para diferentes “*clientes*”.
- **Términos complejos:** Esta clase de términos puede estar compuesto por otros términos. Los términos que componen un término complejo pueden ser constantes, variables o complejos.

SIEXCO-TB Utiliza términos variables y términos constantes, los cuales se encuentran relacionados por Jpredicados, de los que se dan más detalles en la sección 4.5.2.6 de este documento. Los Jpredicados y términos variables se utilizan para construir el cuerpo de la regla, mientras que la conclusión o cabecera esta representada por predicados simples y términos constantes. A continuación se muestra un regla extraído de la base de conocimientos de SIEXCO-TB.

➔ `if identificarGrupo(objetoGrupo,menorde15) and obtenerEstado(objetoGrupo,Clasificacion,FallaRenal) then Tratamiento(TAS9)`

En esta regla se distinguen los siguientes Jpredicados cada uno con sus respectivos términos entre paréntesis:

- `identificarGrupo(objetoGrupo, menorde15)` El Jpredicado es “`identificarGrupo`” y esta relacionando el termino variable “`objetoGrupo`” con el termino constante “`menorde15`”. La relación que plantea este Jpredicado se encarga de determinar si el paciente, representado por el termino variable (`objetoGrupo`), es menor de 15 años, representado por el termino constante (`menorde15`).

- obtenerEstado(*objetoGrupo*, **Clasificacion**, **FallaRenal**) En este prerrequisito la relación que plantea el Jpredicado “obtenerEstado” se encarga de identificar si la clasificación, termino constante (**Clasificacion**), del paciente (*objetoGrupo*) corresponde a falla Renal (constante **FallaRenal**).
- Tratamiento(**TAS9**) A diferencia de los Jpredicados mencionados anteriormente, este representa la conclusión de la regla y esta compuesto por un predicado simple (Tratamiento) que indica un tratamiento en particular (constante **TAS9**). En este caso la conclusión indica la utilización del tratamiento básico (TAS9) para un paciente menor de 15 años con falla renal.

En este ejemplo es importante notar que todos los prerrequisitos actúan sobre la misma variable *objetoGrupo*, que representa las características de un paciente. Esta variable en el momento de realizar la consulta se reemplaza por un paciente en particular. Esta es una característica importante porque toda la información del paciente se encuentra representado por el *objetoGrupo* facilitando la realización de consultas ya que prácticamente el único hecho que hay que adicionar en la consulta es el *objetoGrupo*, del resto se encargan los Jpredicados.

#### 4.5.2.1 Clause Sets

Cuando se tienen reglas y hechos juntos estos son llamados *clauses*. Muchos sistemas utilizan una lista de clauses para representar el conocimiento. La utilización de clauses puede traer algunas complicaciones, porque generalmente los hechos se obtienen desde bases de datos. Las bases de datos pueden estar en archivos locales o fuentes Web, por esto se debe replicar la base de datos en la base de conocimiento, lo que se convierte en un gran problema si la base de datos es grande.

Para solucionar este problema Mandarax utiliza *clause set* en vez de *clausulas*. Los *clause set* son básicamente rutinas que permiten el acceso a colecciones de clausulas. Los *clause set* pueden ser definidos alrededor de una consulta JDBC-SQL, esto es, La consulta retorna un resulset y el *clause set* construye hechos desde los registros en tiempo de consulta. Por lo tanto la replicación de la base de datos ya no es necesaria.

Una aplicación de los *clause sets* en mandarax es que permiten la construcción de reglas en las que los prerrequisitos se encuentran conectados por “or”. Por ejemplo la regla “*if A or B then C*”

es interpretada como un clause set que comprende dos reglas simples (*clauses*) “*if A then C*” y “*if B then C*”.

#### 4.5.2.2 Asignación de prioridades en la base de conocimientos

Mandarax provee métodos para priorizar las reglas. Por defecto las reglas se encuentran priorizadas en el orden en que se encuentran en la base de conocimientos, teniendo mayor prioridad la primera regla y menor prioridad la última. Sin embargo Mandarax permite introducir otras propiedades de comparación que permiten implementar las siguientes estrategias de ordenamiento:

- Preferir hechos y SQL clause sets.
- Preferir hechos con menos variables.
- Preferir reglas con más prerequisites.
- Preferir reglas con menos prerequisites negativos.

Estas estrategias expresan básicamente el conocimiento que se considera más exacto o el conocimiento que tiene preferencia sobre el conocimiento vago. SIEXCO-TB utiliza el ordenamiento por defecto, es decir la prioridad de las reglas se encuentra asignada de acuerdo al orden en que se encuentran en la base de conocimiento.

#### 4.5.2.3 El proceso de inferencia

El motor de inferencia de Mandarax usa razonamiento hacia atrás. En la implementación de este tipo de razonamiento se utilizó orientación a objetos para realizar una versión de razonamiento hacia atrás similar al algoritmo usado en Prolog. Por otra parte muchos sistemas de reglas comerciales como ILOG, CLIPS y JESS usan razonamiento hacia adelante, en particular el popular algoritmo llamado RETE. El algoritmo RETE es un rápido igualador de patrones que obtiene su velocidad del almacenamiento de información sobre los cambios realizados en cada ciclo de búsqueda en una red de reglas. Mandarax utiliza razonamiento hacia atrás por las siguientes razones.

- En el razonamiento hacia adelante se dificulta la integración eficaz de hechos que se encuentran en diferentes bases de datos. Esto se debe a la utilización de software adicional

para programar cambios en las base de datos que permitan mantener actualizado la estructura de derivación. Esto puede ser difícil en particular si se utilizan bases de datos relacionales, debido a que los cambios no se propagan en sus datos y si las conclusiones se requieren en tiempo real, antes de emitir cualquier conclusión se debe sincronizar la base de datos y la base de hechos en un tiempo muy corto.

- Muchos sistemas de reglas actúan sobre grandes bases de hechos y generalmente pequeñas base de reglas (alrededor de 100 reglas). En las bases de reglas grandes se dificulta el entendimiento, introducción y mantenimiento de conocimiento.

En las bases de reglas pequeñas no existe mayor diferencia entre el razonamiento hacia adelante o hacia atrás, Mandarax toma las ventajas existentes en el razonamiento hacia atrás para manejar grandes bases de hechos.

Por otra parte Mandarax tiene implementados 4 motores de inferencia.

- ResolutionInferenceEngine
- ResolutionInferenceEngine2
- ResolutionInferenceEngine3
- ResolutionInferenceEngine4

La razón por la cual Mandarax soporta estos motores de inferencia se debe principalmente a que es una herramienta reciente y en cada nuevo proyecto de desarrollo se han hecho mejoras a los motores de inferencia anteriores. Al mantener los antiguos motores de inferencia Mandarax pretende seguir dando soporte a las aplicaciones desarrolladas en base a motores de inferencia anteriores a la última versión.

SIEXCO-TB se desarrollo con la última versión del motor de inferencia de Mandarax. Es decir con “ResolutionInferenceEngine4”

#### **4.5.2.4 Trabajando con resulset**

Al realizar una consulta en Mandarax el motor de inferencia retorna una estructura muy similar a los resulset usados en JDBC. A través de los métodos next() y previous() se puede navegar

fácilmente en el resulset. En este resulset se encuentran los resultados de las consultas así como las reglas que se utilizaron para llegar a las conclusiones.

SIEXCO-TB aparte de utilizar las conclusiones de las consultas también hace uso de las reglas utilizadas para generar la herramienta explicativa del sistema experto, la cual permite indicarle al usuario el razonamiento utilizado por el sistema experto para llegar a la solución final.

#### 4.5.2.5 Conexión a bases de datos

Cuando se integra el conocimiento desde bases de datos relacionales, primero se debe establecer la conexión a la base de datos. Mandarax utiliza fuentes de datos en vez de simples conexiones y por lo tanto necesita del API javax.sql. Para manejar las bases de datos Mandarax cuenta con predicados SQL, Clause Set SQL y funciones SQL. Estas herramientas facilitan en gran medida el trabajo con bases de datos.

- **Predicados SQL:** Los predicados SQL son básicamente tablas de bases de datos o columnas de tablas, estos cuentan con las siguientes componentes:
  - fuente de datos: Contiene información de cómo conectarse a la base de datos.
  - Nombre: Nombre del predicado SQL.
  - Consulta: Consulta a la base de datos. Por lo general no tienen la cláusula *where*
  - Tipo de mapeo: Describe como fijar los valores en el resulset.
- **Clause Set SQL:** Son similares a los predicados SQL mas la cláusula *where*, Esto permite restringir los renglones en el resulset.
- **Funciones SQL:** Las funciones SQL son funciones basadas en consultas SQL. El objetivo principal de las funciones SQL es tomar ciertos parámetros de entrada para realizar una consulta en la base de datos, esta consulta debe retornar un único renglón, y a partir de este renglón se genera un objeto. En pocas palabras las funciones SQL funcionan de la misma manera que un método que retorna un valor.

#### 4.5.2.6 Integración del modelo de objetos java

Mandarax permite la integración con objetos java por medio de funcionalidades como Jfunctions, Jpredicados y AutoFacts.

SIEXCO-TB hace un amplio uso de los Jpredicados en la implementación de la base de conocimientos, por esta razón se hará una explicación mas profunda sobre este tema.

- **Jfunctions:** Los métodos java son considerados como Jfunctions en Mandarax. Estas funciones toman objetos (parámetros) como entradas y retornan otros objetos
- **AutoFacts:** Son clause set genéricos que permiten generar hechos a partir de un conjunto de datos y integrarlos a la base de conocimientos en tiempo de consulta.

**Jpredicados:** Los Jpredicados se pueden considerar como métodos que siempre retornan un valor booleano. Cada Jpredicado recibe como parámetros el objeto que contiene el método y los parámetros asociados al método.

En SIEXCO-TB existen dos tipos de Jpredicados y la mayoría de los prerequisites de las reglas se encuentran contruidos en base a ellos. A continuación se explica como se construyeron y en que consiste cada uno.

##### 1. Jpredicado identificarGrupoPaciente

```
public Predicate identificarGrupoPaciente() {
    Method m = null;
    try {
        m = BeanGrupoPaciente.class.getMethod("identificar", new Class[]
{    String.class });
    } catch (SecurityException e) {
        System.out.println("Error creating predicate");
    } catch (NoSuchMethodException e) {
        System.out.println("Error creating predicate");
    }
    return new JPredicate(m, "El paciente es");    }
```

Este es el método que permite definir el Jpredicado `identificarGrupo`, La parte principal de este método se encuentra resaltada en negrita. En este renglón se obtiene el método que compone el Jpredicado, para esto es necesario indicar la clase donde se encuentra el método su nombre y los parámetros que recibe. En este caso el método se encuentra en la clase `BeanGrupoPaciente` su nombre es `identificar` y recibe como parámetros un objeto de tipo `String`. En el último renglón de este método se instancia el Jpredicado, el constructor de esta clase recibe como parámetros el método (`m`) y el nombre ("`El paciente es`") con el cual se conocerá el Jpredicado en la base de conocimientos. Finalmente se retorna un objeto de tipo `JPredicate`.

La función de este Jpredicado es identificar si el paciente es mayor o menor de 15 Años. Para un mejor entendimiento acerca de su utilización a continuación se muestra un prerequisite extraído de la base de conocimiento de SIEXCO-TB que utiliza un Jpredicado de este tipo.

→ if El paciente es(*paciente*, **mayor de 15 años**)

Este prerequisite está compuesto por un Jpredicado del tipo `El paciente es`, Como se explicó anteriormente se debe indicar en los términos de entrada el objeto donde se encuentra el método junto con sus parámetros de entrada. En este caso el término variable `paciente` puede ser reemplazada por un objeto del tipo `BeanGrupoPaciente` y el término constante **mayor de 15 años** es el parámetro de entrada del método `identificar` que se encuentra definido en la clase `BeanGrupoPaciente`.

Como se mencionó anteriormente los Jpredicados únicamente retornan tipos booleanos, por lo tanto el método `identificar` que está representado por este Jpredicado retorna `true` si el paciente es menor de 15 años (término **mayor de 15 años**) de lo contrario retorna `false`. Vale la pena aclarar que el prerequisite es satisfecho únicamente si es verdadero.

## 2. Jpredicado `obtenerEstado`

```
public Predicate obtenerEstado() {
    Method m = null;
    try {
        m = BeanGrupoPaciente.class.getMethod("getEstado", new Class[] {
String.class, String.class });
    } catch (SecurityException e) {
        System.out.println("Error creating predicate" );
    } catch (NoSuchMethodException e) {
```

```

        System.out.println("Error creating predicate");
    }
    return new JPredicate(m, "presenta");
}

```

La construcción de este Jpredicado es similar a la del Jpredicado `identificarGrupo` explicada anteriormente, la única diferencia es que el método representado por este Jpredicado se llama *presenta* y recibe como parámetros dos objetos de tipo String.

La función de este Jpredicado es identificar si el paciente presenta alguna situación especial. A continuación se muestra un ejemplo de un prerrequisito construido a partir de un Jpredicado de este tipo.

```

→ if presenta(paciente, Clasificacion, FallaRenal)

```

Este prerrequisito es similar al explicado en el Jpredicado *El paciente es y es satisfecho o verdadero* si el paciente (termino variable *paciente*) presenta alguna situación especial (Termino Constante **Clasificacion**) igual a falla renal (Termino Constante **FallaRenal**), de lo contrario es falso.

La mayor ventaja de la utilización de Jpredicados es que permiten la generación de predicados de acuerdo a las necesidades particulares del diseño del sistema experto. En SIEXCO-TB permiten la realización de consultas directamente sobre la estructura de datos del paciente (*paciente*).

En la figura 10 se muestra como esta estructurada la información del paciente utilizada por SIEXCO-TB, aquí se puede observar que el termino **Clasificacion** corresponde a un indicador y el termino **FallaRenal** corresponde a un parámetro. La estructuración de la información junto con la utilización de Jpredicados facilita enormemente el entendimiento, construcción y mantenimiento de la base de conocimientos, porque los nombres de los Jpredicados pueden ser definidos con un gran contenido semántico que facilita el entendimiento de las reglas, la información del paciente se encuentra estructurada adecuadamente para facilitar las consultas y posibles modificaciones futuras (Adicionar o Remover indicadores y parámetros), el árbol esta construido en base a beans serializables, lo que facilita el manejo y almacenamiento de la información en la base de datos, etc.



#### 4.5.2.7 Base de conocimiento en XML

Mandarax cuenta con un paquete llamado XKB que le permite dar soporte a la serialización de bases de conocimiento a documentos XML, este hace uso de la librería JDOM de la que se dan mas detalles en la sección 4.4.2.2. Este paquete tiene drivers que permiten la utilización de formatos como RuleML y a los formatos de Mandarax XKB y ZKB.

SIEXCO-TB hace uso del driver XKB para la serialización a XML de la base de conocimientos. Para el almacenamiento de los documentos XML se utilizan siete archivos XKB con el fin de generar base de conocimiento con un número pequeño de reglas que facilite el entendimiento y la construcción de estas y mejore el rendimiento del sistema. Para una mejor comprensión de los beneficios que trae la utilización de varios archivos a continuación se explica como y por que se implementaron.

A pesar la existencia de siete archivos diferentes, estos se pueden clasificar en dos grupos:

- Grupo 1: Esta conformado por el archivo *Reglas\_Generale.XKB*, en este archivo se encuentran las reglas generales que permiten identificar un tratamiento base, es decir reglas que permiten identificar la edad del paciente, su peso, su esquema y en general datos del paciente que realizan modificaciones relevantes al tratamiento base y por lo tanto ameritan la generación de un nuevo tratamiento. Además de este tipo de reglas también se encuentran las reglas que permiten identificar las recomendaciones que se deben tener con cada paciente de acuerdo a sus características particulares. A continuación se muestran dos reglas de este grupo.

→ if El paciente es(*paciente*,**mayor de 15 años**) and presenta(*paciente*,**Clasificacion**,**FallaRenal**) then Tratamiento(**TAS9**)

→ if El paciente es(*paciente*,**mayor de 15 años**) and presenta(*paciente*,**Esquema**,**Fracaso**) then sugerir(**ACCpruSen**)

- Grupo 2: Antes de utilizar las reglas de este grupo se debe identificar el tratamiento base por medio de las reglas del grupo 1. Estos tratamientos (archivos XKB) van desde el *TAS1* hasta el *TAS6* y cada uno esta conformado por las reglas que se deben utilizar de acuerdo al tratamiento base seleccionado previamente. Las reglas de este grupo se encargan de identificar las modificaciones que se deben hacer al tratamiento base de acuerdo a las características mas particulares de cada paciente, estas modificaciones tienen que ver con la duración del tratamiento, los medicamentos a suministrar y sus dosis. A continuación se muestra una regla de este grupo.

→ if El paciente es(*paciente*,**mayor de 15 años**) and presenta(*paciente*,**Clasificacion,Silicosis**) then modificar(**dosisFase2:63**)

Los beneficios de la utilización de varios archivos se debe principalmente a que en cada archivo se encuentran reglas que representan conocimiento específico acerca de un tratamiento previamente definido (reglas del grupo 2), o conocimiento aplicable de forma general a cualquier paciente independiente de la presencia de características más particulares (reglas del grupo uno). De esta forma la construcción de reglas y el entendimiento de la base de conocimiento se hace mucho más fácil porque al momento de adicionar o modificar alguna regla se sabe a que grupo y a que tipo de archivo pertenece. Además con pequeñas bases de conocimiento (archivos XKB) el número de prerequisites a tener en cuenta es menor lo que implica reglas cortas, fáciles de entender y construir.

Otro aspecto importante de segmentar la base del conocimiento, tiene que ver con la eficiencia del sistema, ya que en una consulta determinada no es necesario acceder a la base de conocimiento completa sino a una porción de esta, lo que hace más rápidas las búsquedas al ser más directas.

#### 4.5.2.8 XKB versus RuleML

El driver XKB no es compatible con el estándar RuleML. La razón principal es que RuleML actualmente soporta muy pocas características de las implementadas por Mandarax, como son:

- Términos Complejos y Funciones.
- Clase set (RuleML soporta solamente hechos simples)
- Integración de fuentes de datos SQL.
- Prerequisites conectados por OR.

A pesar de esto y teniendo en cuenta que RuleML ya es un estándar, los niveles superiores de Mandarax se diseñaron de tal forma que sean compatibles con RuleML 0.8. De esta manera se pretende facilitar la migración a RuleML y sus versiones futuras.

### 4.5.3 MySql

MySql [19] es un servidor de bases de datos relacionales muy rápido y robusto. Es software libre, publicado bajo la licencia GPL (GNU Public License) y mantenido por la compañía sueca MySql AB. Es uno de los más indicados para desarrollar aplicaciones que requieran muchas lecturas y pocas escrituras y no necesiten de características muy avanzadas, como en el caso de aplicaciones Web.

SIEXCO-TB cuenta con una base de datos relacional implementada en MySql. En esta base de datos se almacena información relacionada con: datos del paciente, datos de usuario, datos administrativa, información sobre los medicamentos, estructura de los meta datos e información de los diferentes repositorios.

A pesar de que se maneja una alta cantidad de información, la estructura de la base de datos de SIEXCO-TB es muy simple. Esto se debe a que la información se almacena en formato XML. Por lo general una de las columnas de cada tabla contiene información en XML y las demás tienen las claves foráneas que permiten relacionar las diferentes tablas, de esta forma la mayoría de datos están almacenados en una sola columna en texto XML lo cual facilita la creación de tablas sencillas con muy pocas columnas a pesar de la gran cantidad de datos diferentes.

Esta forma de almacenar información puede generar un poco de redundancia debido a que los datos utilizados como claves también se encuentran en el documento XML. Sin embargo esto no representa un problema relevante en cuanto a consistencia de la información se refiere, por que cada vez que se modifique o ingresen nuevos registros en la base de datos el sistema se verifica que la información concuerde y sea almacenada correctamente.

#### 4.5.3.1 Diseño de la base de datos relacional de SIEXCO-TB.

Como ya se ha dicho anteriormente, el uso del estándar XML ha simplificado mucho el diseño y construcción de la base de datos, a pesar del gran número de datos que se maneja en la tarjeta individual y en los diferentes informes. Como se puede ver en la siguiente figura las tablas tienen pocas columnas debido a que la mayoría de información se encuentra representada en XML y de esta forma es posible almacenarla en un solo campo. El resto de columnas de las tablas corresponde a las claves primarias y a claves foráneas que permiten establecer relaciones entre ellas.

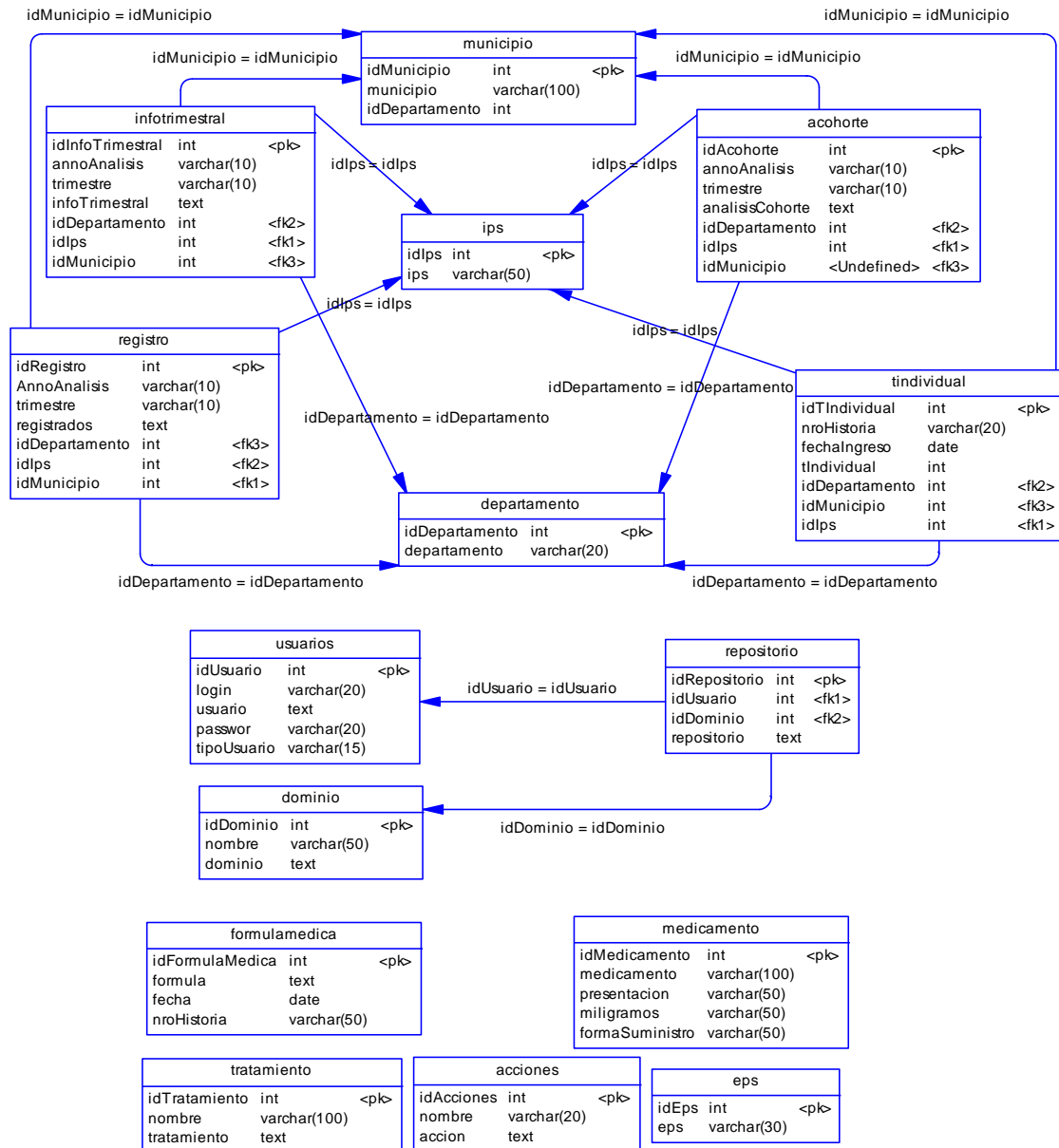


Figura 15. Diagrama Base de Datos

A continuación se hace una breve descripción del contenido de cada tabla.

- **Tabla infotrimestral:** contiene los datos de los informes trimestrales. Esta información se encuentra representada en un archivo XML, que está almacenado en el campo *infoTrimestral*.
- **Tabla acohorte:** contiene los datos de los análisis de cohorte. Esta información se encuentra representada en un archivo XML, que está almacenado en el campo *analisisCohorte*.

- **Tabla registro:** Contiene los datos del registro de pacientes. Esta información se encuentra representada en un archivo XML, que se esta almacenado en el campo *registrados*.
- **Tabla tindividual:** contiene los datos de los pacientes correspondientes a la tarjeta individual. A partir de estos datos se generan el análisis de cohorte, el informe trimestral y el registro de pacientes. El almacenamiento de esta información se realizo mediante documentos XML y el campo que los contiene es *tIndividual*.
- **Tabla usuarios:** Contiene información correspondiente a los usuarios que tienen acceso al sistema. De la misma forma que en los casos anteriores estos datos se encuentran en un archivo XML y en esta tabla están almacenados en el campo *usuario*.
- **Tabla repositorio:** Esta tabla contiene información que permite relacionar un repositorio o base de conocimiento con el usuario que lo creo y con el dominio al cual pertenece. Además contiene información correspondiente al nombre del repositorio y su descripción, estos datos se encuentran representados en un documento XML y están almacenados en el campo *repositorio*.
- **Tabla dominio:** En esta tabla se almacena los datos correspondientes a los dominios o tipos de enfermedades de las cuales el sistema experto tiene conocimiento. En otras palabras contiene una instancia de la estructura de los metadatos. Actualmente solamente contiene la estructura para la tuberculosis pulmonar y extrapulmonar. Estos datos están representados es un documento XML que esta almacenados en el campo *dominio*.
- **Tabla formulamedica:** contiene los datos correspondientes a las formulas médicas que se generaron y guardaron en el sistema. En esta tabla cada formula medica esta asociado a un numero de historia con el fin de determinar a que paciente corresponde cada una. La información correspondiente a al formula medica se encuentra representada en un documento XML que esta almacenado en el campo *formula*.
- **Tabla medicamento:** Contiene información de los diferentes medicamentos que se deben suministrar en el tratamiento de pacientes afectados por la tuberculosis.
- **Tabla tratamiento:** En esta tabla se encuentran los diferentes tratamientos básicos para el control de la tuberculosis que fueron definidos en la etapa de construcción de la base de conocimiento del sistema experto. Estos son los tratamientos que se cargan y luego se modifican según el resultado de las consulta al sistema experto. Esta información se encuentra representada mediante objetos serializados en documentos XML, y están almacenados en el campo *tratamiento*.
- **Tabla acciones:** contiene las recomendaciones que el sistema experto despliega en la interfaz grafica despendiendo del resultado de las consultas al sistema experto.

- **Tabla eps:** contiene el nombre de las administradoras de salud que se registren en el sistema.
- **Tabla ips:** contiene el nombre de las ips que se registren en el sistema.
- **Tabla municipio:** contiene el nombre de los municipios que se registren en el sistema. Los municipios están asociados a un departamento.
- **Tabla departamento:** contiene el nombre de los departamentos que se registren en el sistema.

#### 4.5.4 Epi\_Info

Epi\_Info [20] es un programa de dominio público diseñado por el Centro para el Control de Enfermedades de Atlanta (CDC). Es un sistema de fácil uso y ampliamente utilizado en el área de medicina para la vigilancia de brotes epidémicos. Este programa permite entre otras cosas, el análisis de datos con las estadísticas de uso básico en epidemiología. Estos datos estadísticos pueden ser visualizados mediante una gran variedad de gráficos y mapas.

##### 4.5.4.1 SIEXCO TB y Epi Info.

SIEXCO-TB genera información compatible con los formatos manejados por Epi\_Info, la cual esta relacionada con los efectos colaterales producidos por los medicamentos suministrados a los pacientes e indicadores epidemiológicos y operativos del programa de atención de pacientes con tuberculosis. Para un mejor entendimiento de que tipo de análisis estadísticos se pueden realizar con esta información a continuación se explica como se genera y de que datos se compone.

La información brindada por SIEXCO-TB que es compatible con los formatos manejados por Epi\_Info se encuentra almacenada en una base de datos Access llamada **EstSincotb**. Esta base de datos esta compuesta por las siguientes tablas *signosSintomas*, *signosToxicidad*, *TblPacientes* y cada una se genera de la siguiente manera:

- *signosSintomas*: Esta tabla se actualiza automáticamente y la información almacenada se obtiene de la tarjeta individual, la mayoría de datos corresponden a los “SIGNOS Y SÍNTOMAS” de la tabla “EVOLUCIÓN DEL PACIENTE”. Los datos almacenados en esta tabla se muestran en la siguiente figura.

signosSintomas
id
escolaridad
indice_masa_corporal
estrato
edad
talla
peso
tiempo_tratamiento
temperatura
tos
expectoracion
anorexia
adinamia
sudoracion
disnea
hemoptisis
No_Historia

**Figura 16** Tabla de signos y síntomas

En las columnas correspondientes a signos y síntomas (las que van desde *tos* hasta *hemoptisis*) se almacena *si* o *no* para indicar que el paciente presenta o no el signo o síntoma indicado por el nombre de la columna. De esta forma se pueden realizar análisis estadísticos en los que se pueden relacionar los diferentes signos y síntomas con los demás datos almacenados en esta tabla como son: estrato, edad, peso, tiempo\_tratamiento, etc.

- *signosToxicidad*: Esta tabla y la fuente de sus datos es similar a la mencionada en ítem anterior la diferencia es que la mayoría de los datos ahora corresponden a “SIGNOS DE TOXICIDAD O INTOLERANCIA” de la tabla “EVOLUCIÓN DEL PACIENTE”. Los datos almacenados en esta tabla se muestran en la siguiente figura.

signosToxicidad
id
tiempo_consumo_medicamento
peso
sexo
estreptomycin
ethambutol
isoniazida
rifampicina
pirazinamida
etionamida
tioacetazona
No_Historia
asociacion_r_h
edad

**Figura 17** Tabla de signos de toxicidad

Como se puede ver en la figura anterior el nombre de la mayoría de las columnas de esta tabla corresponde a los medicamentos que pueden incluirse en el tratamiento de un paciente afectado por tuberculosis. Los datos que se almacenan en estas columnas corresponden a los signos de toxicidad o intolerancia que el paciente presenta cuando esta consumiendo el medicamento. Por ejemplo si el paciente presento vómito cuando estaba consumiendo el medicamento estreptomycin, en la columna estreptomycin se registra vómito. De esta manera se pueden realizar fácilmente análisis estadísticos que permiten relacionar los signos de toxicidad o intolerancia con los medicamentos que el paciente consume. Además también se pueden relacionar los diferentes signos de toxicidad o intolerancia asociados a un determinado medicamento con el resto de datos, como son: peso, sexo, edad y tiempo\_consumo\_medicamento.

- *TblPacientes*: La información de esta tabla se actualiza automáticamente y se obtiene de la tarjeta individual. Los datos almacenados en esta tabla permiten generar análisis estadísticos epidemiológicos y operativos del programa de atención de pacientes con tuberculosis. En la siguiente grafica se muestra el tipo de información almacenado en esta tabla.

Id
Cedula
Anno
Departamento
Municipio
Administradora
Ips
Nombre
Apellido1
Apellido2
Genero
Edad
Regimen
TipoPaciente
TipoTuberculosis
LugarExtraPulmonar
FechaDiagnostico
Raza
TiempoSintomas
BkIniciales
Mes2
Mes4
Mes5
Mes6
CondicionEgreso
Escolaridad
Estrato
MasaCorporal
Talla
NroHistoria

**Figura 18** Tabla para análisis epidémico



#### 4.5.4.2 Análisis de datos con Epi Info

El análisis de datos brindado por Epi\_Info proporciona herramientas fáciles de utilizar para producir muchas formas de análisis estadístico útil para los epidemiólogos y otros profesionales de salud pública.

La información ha ser analizada estadísticamente puede ser recogida usando directamente Epi\_Info o pueden provenir de otro tipo de base de datos, tales como: DBase, FoxPro, Excel, Access, etc. En el caso de SIEXCO\_TB, como ya se mencionó antes, la base de datos compatible con Epi\_Info se encuentra implementada en Access.

Después de tener la información almacenada en un formato compatible con Epi\_Info es posible realizar, entre otros, los siguientes análisis estadísticos:

- **Estadísticas simples:**
  - Distribución de frecuencias.
  - Gráficas de las distribuciones de frecuencias.
  - Medias descriptivas.
  - Tablas.
  - Mapas
  
- **Estadísticas avanzadas:**
  - Frecuencia de muestreo complejo.
  - Tablas de muestreo complejo.
  - Medias de muestreo complejo
  - Regresiones lineales.

Las estadísticas mencionadas anteriormente se pueden aplicar a los datos almacenados en las tablas descritas en el numeral 4.5.4.1 SIEXCO\_TB y Epi\_Info. De esta manera SIEXCO\_TB brinda información estadística de gran utilidad para la realización de estudios epidemiológicos en tuberculosis, y además facilita el suministro de indicadores de efectos colaterales producidos por los medicamentos, que contribuyen a prevenir trastornos futuros en los pacientes, los cuales son factores influyentes en el incremento de los abandonos terapéuticos.

## CONCLUSIONES

- Mediante el uso de sistemas expertos, es posible proponer soluciones a problemas de toma de decisiones teniendo como base información obtenida desde fuentes como libros y revistas, además de la información adquirida de expertos en el dominio de la aplicación. Además, los sistemas expertos son lo suficientemente flexibles como para soportar actualizaciones continuas al conocimiento del dominio, debido tanto a los adelantos en investigación en el tratamiento de la enfermedad, así como a las nuevas experiencias adquiridas por los expertos encargados de alimentar al sistema. SIEXCO – TB aprovecha todas estas características para brindar una solución óptima de apoyo al control de la Tuberculosis.
- Para decidirse por el uso del paradigma de los sistemas expertos para construir una aplicación es necesario que se cumplan algunos criterios para el éxito del proyecto, como son: la adecuada delimitación del Dominio, la necesidad y el deseo de tener un sistema experto, la presencia de al menos un especialista dispuesto a cooperar y que parte del conocimiento sea heurístico ó empírico.
- El análisis de decisión está siendo gradualmente reconocido como una técnica poderosa para seleccionar las estrategias óptimas en problemas de decisión clínica. Muchos esfuerzos en el área de los sistemas de decisión basados en conocimiento pretenden automatizar el proceso de análisis de decisión usando técnicas de inteligencia artificial. Los sistemas expertos de apoyo en la toma de decisiones clínicas representan una aproximación importante a la solución de problemas en el área de la medicina, tales como el diagnóstico, tratamiento y control de enfermedades.
- SIEXCO - TB es una herramienta que ha sido concebida como una primera aproximación a partir de la cual se pretende construir toda una familia de herramientas para el control de enfermedades de alta prevalencia. Es por esto que el sistema pretende ser muy flexible y extensible. Es por eso que SIEXCO – TB trabaja con una jerarquía de meta datos a partir de la cual se construye un modelo de indicadores específico que es usado por cada sistema experto implementado.

- En SIEXCO- TB el conocimiento esta representado mediante reglas de producción, las cuales están clasificadas, lo que vuelve al sistema modular, sencillo y eficiente. De esta manera la adquisición de nuevo conocimiento, así como su mantenimiento se simplifica de manera notable.
- A nivel de implementación, SIEXCO-TB además de estar soportado en el paradigma de sistemas expertos para su construcción, se hace fuerte gracias al uso de dos tecnologías ampliamente difundidas gracias a la larga lista de características que ofrecen. Estas son JAVA y XML. Con estas tecnologías se logran dos objetivos fundamentales: Código portable y datos portables.
- XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Así pues, XML juega un papel importantísimo en este mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología que permitirá compartir la información de una manera segura, fiable y fácil. Además, XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.
- SIEXCO – TB aprovecha muchas de las características de Swing, utilizando todas las categorías de componentes swing para obtener una variedad de interfaces de usuario sofisticadas que hacen que la interacción del usuario con el sistema sea muy agradable. Este es un aspecto muy importante ya que la riqueza de las interfaces hace que el usuario de SIEXCO – TB aproveche la funcionalidad ofrecida por la aplicación de la mejor manera.
- En SIEXCO – TB el uso XML para representar cierta información del sistema es el sustento fundamental para soportar algunas de sus características, tales como: La portabilidad de la información, la compatibilidad con otras herramientas, la facilidad de exportación / importación de información de pacientes y de tratamientos e informes del sistema y la simplicidad de la base de datos relacional.
- La elaboración de bases de conocimiento segmentadas incrementa la eficiencia del sistema, ya que en una consulta determinada no es necesario acceder a la base de conocimiento

completa sino a una porción de esta, lo que hace más rápidas las búsquedas al ser más directas. Por otra parte facilita la construcción, mantenimiento y entendimiento de esta, por que con un numero pequeño de reglas en cada segmento, el número de prerrequisitos a tener en cuenta es menor lo que implica reglas cortas, fáciles de entender y construir.

- La utilización de herramientas que incorporan el paradigma orientado a objetos para la construcción de sistemas expertos permiten diseñar bases de conocimiento de forma parecida a como es percibido el conocimiento en la realidad. De esta forma se simplifica en gran medida la construcción y mantenimiento de la base de conocimientos, ya que la persona poseedora del conocimiento y la experiencia lo podrá ingresar al sistema sin mayores dificultades.
- En el sector de la salud es muy importante el análisis de la información para obtener nuevas conclusiones, por ejemplo, identificar los posibles efectos colaterales que puede presentar un paciente por el suministro de un medicamento determinado puede contribuir a prevenir trastornos futuros en dichos pacientes. SIEXCO-TB genera esta información en un formato compatible con Epi\_Info la cual permite generar análisis estadísticos, que les permitirán a los profesionales de la salud identificar claramente diferentes indicadores asociados con el paciente y el tratamiento suministrado.
- Unificar las acciones diagnósticas y los esquemas terapéuticos para el control de la tuberculosis es de vital importancia para controlar la epidemia y evitar la aparición de cepas resistentes a múltiples drogas, SIEXCO-TB permite a los profesionales de la salud ceñirse a los esquemas de tratamiento recomendados nacional e internacionalmente y garantizar a las Instituciones de la salud un mejor control de la enfermedad.

## RECOMENDACIONES

- La base de conocimiento de SIEXCO – TB ha sido construida teniendo en cuenta información científico / médica producto de la investigación y la experiencia en el control de la tuberculosis. Se recomienda que las personas encargadas de manipularla sean verdaderos expertos en el tema. De la correcta manipulación de esta información depende la efectividad en los resultados arrojados por el sistema. No olvidarse que se esta tratando con personas enfermas cuya curación puede verse comprometida por el manejo irresponsable de la aplicación.
- El diseño modular a nivel de reglas y de datos permite que SIEXCO – TB se pueda extender a aplicaciones similares para el tratamiento de enfermedades de alta prevalencia. Vale la pena tener en cuenta el trabajo realizado para seguir profundizando en el tema y desarrollando productos que apoyen el manejo de este tipo de enfermedades.
- El trabajo multidisciplinario entre la facultad de Electrónica y Telecomunicaciones y la dirección departamental de Salud del Cauca fue fundamental para la obtención de resultados positivos en el proyecto. Vale la pena fortalecer los lazos con este tipo de instituciones que permitan la ejecución de proyectos útiles, ajustados a las necesidades de la región.

## BIBLIOGRAFÍA

- [1] Galvis Ramírez, Virgilio. **Guía de Atención de la Tuberculosis Pulmonar y Extrapulmonar**. Ministerio de Salud, Colombia. 2002.
- [2] Joseph giarratano. **Sistemas Expertos Principios y Programación**. Thomson Editores, 2001.
- [3] **Sistemas expertos**. Disponible en <http://www.esi2.us.es/~dco/sistemas.htm>
- [4] **Introducción a los sistemas expertos**. Disponible en [http://ingenieroseninformatica.org/recursos/tutoriales/sist\\_exp/index.php](http://ingenieroseninformatica.org/recursos/tutoriales/sist_exp/index.php)
- [5] Juan José Samper Márquez **Introducción a los sistemas expertos**. Disponible en <http://www.redcientifica.com/doc/doc199908210001.html>
- [6] Chantal Pérez Hernández **Esquemas de representación de conocimiento formalizados**.2002. Disponible en <http://elies.rediris.es/elies18/523.html>
- [7] **La tuberculosis sigue siendo un problema grave de salud en las Américas, señalan expertos en el Día de Mundial de la Tuberculosis**. OPS, Washintong, 24 de Marzo de 2003. Disponible en <http://www.paho.org/spanish/dd/pin/ps030324.htm>.
- [8] **Un Marco Ampliado de DOTS para el Control Eficaz de la Tuberculosis**. Organización Mundial de la Salud, Ginebra, 2002.
- [9] Perú. **Programa de control de la tuberculosis. ¿Por qué un ejemplo para el mundo?**. OPS, OMS, Ministerio de Salud, 1997.
- [10] **World Health Organization: Treatment of Tuberculosis. Guidelines for National Programmes**. Geneva, World Health Organization, 1997
- [11] Gutiérrez José Manuel. **Sistemas Expertos Basados en Reglas**. Departamento de matemática aplicada, Universidad de Cantabria. Disponible en <http://personales.unican.es/gutierjm/ cursos/expertos/Reglas.pdf>

- [12] **CLIPS Expert System Shell.** Disponible en <http://www.jsc.nasa.gov/~clips/CLIPS.html>.
- [13] **Jess. The Java Expert System Shell.** Disponible en <http://herzberg.ca.sandia.gov/jess/>
- [14] Figueira Filho Carlos. **JEOPS - The Java Embedded Object Production System.** Agosto de 2001. Disponible en <http://www.cin.ufpe.br/~jeops/>
- [15] **Mandarax.** Disponible en <http://www.jbdietrich.com/mandarax.html>
- [16] **ILOG JRules.** Disponible en <http://www.ilog.com/products/jrules/>
- [17] **Oryx Standalone.** Disponible en <http://www.jbdietrich.com>
- [18] Jens Dietrich. **The Mandarax Manual.** Disponible en [http://sourceforge.net/project/showfiles.php?group\\_id=50817](http://sourceforge.net/project/showfiles.php?group_id=50817)
- [19] **Overview of the MySQL Database Management System.** Disponible en <http://www.mysql.com/doc/en/What-is.html>
- [20] **Manual Epi\_Info.** Disponible en <http://www.cica.es/epiinfo/manual.htm>
- [21] **Java.** Disponible en <http://java.sun.com/docs/white/index.html>
- [22] **Sistema de verificación de componentes software.** Disponible en [http://www.di.uniovi.es/~cernuda/investig\\_ESP.html](http://www.di.uniovi.es/~cernuda/investig_ESP.html)
- [23] **JavaBeans.** Disponible en <http://www.programacion.com/java/tutorial/javaBeans>
- [24] **Java Foundation Classes.** Disponible en <http://java.sun.com/products/jfc/>
- [25] **Jdbc.** Disponible en <http://java.sun.com/products/jdbc/overview.html>
- [26] **JavaMail.** Disponible en <http://www.programacion.com/java/tutorial/javamail/1/>
- [27] **J2EE y XML.** Disponible en <http://benmira.free.fr/xml.htm>

[28] **XSD (XML Schema definitions)**. Disponible en [http:// www.w3c.org/XML/Schema](http://www.w3c.org/XML/Schema)

[29] **XLAN**. Disponible en <http://xml.apache.org/xalan-j>

[30] **JAXM**. Disponible en <http://java.sun.com/xml/jaxm/>

[31] **JAXR**. Disponible en <http://java.sun.com/xml/jaxr>