



**UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSI**

**FACULTAD DE INGENIERÍA**

**POSGRADO EN INGENIERÍA MECÁNICA  
OPCIÓN TERMOFLUIDOS**

**“DESARROLLO DE SISTEMA EXPERTO PARA  
MANTENIMIENTO PREVENTIVO”**

**Para obtener el título de  
MAESTRO EN INGENIERÍA**

**AUTOR  
ING. JUAN MANUEL SILVA CAMPOS**

**ASESOR  
DR. RICARDO ROMERO MÉNDEZ**

**SAN LUIS POTOSÍ, S. L. P.  
ENERO 2010**



# CONTENIDO

<b>INTRODUCCIÓN</b>	9
---------------------	---

<b>OBJETIVO</b>	12
-----------------	----

## **CAPÍTULO I ANTECEDENTES TEÓRICOS**

---

1.1	Concepto de Inteligencia Artificial	13
1.2	Historia de la Inteligencia Artificial	13
1.3	Desarrollo histórico de los sistemas expertos	16

## **CAPÍTULO II GENERALIDADES DE LOS SISTEMAS EXPERTOS**

---

2.1	¿Qué son los Sistemas Expertos?	19
2.2	Los Expertos Humanos	21
2.3	Diferencias entre un Experto humano y un experto humano	22
2.4	Ventajas de los SE	23
2.5	Diferencias entre un SE y un programa tradicional	23
2.6	Tareas que realizan los sistemas expertos	24
2.7	Campos de aplicación de los sistemas expertos	29
2.8	Componentes de un sistema experto	32

## **CAPITULO III ESTRUCTURA DE UN SISTEMA EXPERTO**

---

3.1	Introducción de los Sistemas Expertos en la empresa	35
3.2	Aspectos Generales	38
3.3	Metodología para la construcción de un SE	39
3.4	Lenguaje de Programación Visual Basic	42

## **CAPÍTULO IV TIPOS DE SISTEMAS EXPERTOS**

---

4.1	Sistema Expertos Determinísticos	46
4.1.1	Introducción	46
4.1.2	La Base de Conocimientos	46
4.1.3	Motor de Inferencia	47
4.2	Sistemas Expertos que manejan Incertidumbre	48
4.2.1	Introducción	48
4.2.2	Método de Factores de Certeza	48
4.2.3	Lógica Difusa	50
4.2.4	Probabilidad Bayesiana	51
4.3	Conceptos sobre grafos	53
4.3.1	Introducción	53
4.3.2	Conceptos básicos y definiciones	53
4.3.3	Grafos dirigidos y no dirigidos	54
4.3.4	Características de grafos dirigidos	55
4.4	Sistemas Expertos Basados en Probabilidad	59
4.4.1	Introducción	59
4.5	Sistemas Expertos Basados en Reglas vs. SE basados en Probabilidad	60

## **CAPÍTULO V DESCRIPCION DEL SISTEMA EXPERTO DESARROLLADO**

---

5.1	Estructura del sistema experto	62
5.1.1	Componentes	62
5.1.2	Conjuntos de reglas	63
5.1.3	Descripción de tipos de variables	64
5.2	Descripción de subsistema de aprendizaje	64
5.2.1	Creación y modificación de un conjunto de reglas	64
5.2.2	Creación y modificación de una regla	66
5.2.3	Creación y modificación de una variable	67

## CAPITULO VI APLICACIÓN DEL SISTEMA EXPERTO

---

6.1	Definición del problema. SISTEMA DEL TANQUE DE PRESIÓN (Caso 1)	75
6.2	Selección de variables	76
6.3	Conjunto de reglas utilizadas	77
6.4	Propagación de Evidencia	80
6.5	Programación de Reglas y Variables	81
6.6	Programación del Método Probabilístico en el Sistema Experto	83
6.7	Definición del problema. SISTEMA DE DISTRIBUCIÓN DE ENERGÍA (Caso 2)	86

## CAPITULO VII RESULTADOS

---

7.1	SISTEMA DEL TANQUE DE PRESIÓN (Caso 1)	92
7.2	SISTEMA DE DISTRIBUCIÓN DE ENERGÍA (Caso 2)	94

### CONCLUSIONES

### APENDICE A: MOTOR DE INFERENCIA REGLAS EXACTAS

### APENDICE B: MOTOR DE INFERENCIA POR LÓGICA DIFUSA

### APENDICE C: MOTOR DE INFERENCIA PARA SISTEMA EXPERTO POR METODO DE CERTEZAS

### APENDICE D: MOTOR DE INFERENCIA PARA SISTEMA EXPERTO PROBABILISTICO, METODO EXACTO

### APENDICE E: MOTOR DE INFERENCIA PARA SISTEMA EXPERTO PROBABILISTICO, METODO ESTOCASTICO

### APENDICE F SISTEMAS DIFUSOS Y OPERADORES

### BIBLIOGRAFIA

# AGRADECIMIENTOS

Primeramente quiero agradecer a Dios por darme la vida y por todo lo que me ha brindado hasta hoy, por darme la fuerza para terminar este trabajo.

A mis padres J. Manuel Silva Hdz. y Carmen Campos por su apoyo incondicional y sus consejos para conseguir siempre lo que deseamos. A mis hermanas Magdalena, Angélica y Elba.

A mi esposa Verónica por su apoyo y comprensión en todo momento.

A mi asesor Dr. Ricardo Romero Méndez que sin su apoyo este trabajo no se hubiera realizado.

Un agradecimiento muy grande a los integrantes del comité de revisión de este trabajo, por sus consejos y observaciones.

Y finalmente, un agradecimiento a todos mis compañeros de generación, que de manera directa o indirecta colaboraron conmigo para llegar hasta este momento, a todos ¡Gracias por su apoyo!

# LISTA DE FIGURAS Y TABLAS

Tabla 1.1. Sistemas Expertos y sus aplicaciones.

Tabla 2.1. Diferencias entre un experto y un no experto humano.

Tabla 2.2. Diferencias entre un Sistema Experto y un Experto Humano.

Tabla 2.3. Diferencias entre un Sistema Experto y un Programa Tradicional.

Figura 2.1. Estructura de un Sistema Experto según De Miguel.

Figura 2.2. Estructura de un Sistema Experto ideal para Samper

Figura 3.1. La base de conocimientos es vista como un objeto contenedor de otros objetos usando la Metodología Orientada a Objetos.

Figura 4.1. Ejemplo de Grafo o Red.

Figura 4.2. Ejemplos de un grafo dirigido (a) y uno No dirigido (b).

Figura 4.3. Padres e hijos del nodo E.

Figura 4.4. Familias asociadas a los nodos de un grafo.

Figura 4.5. Ascendientes y descendientes del nodo E.

Figura 4.6. Dos numeraciones ancestrales del mismo grafo.

Figura 4.7. Ejemplos de grafos dirigidos: árbol simple (a) y poliárbol (b).

Figura 4.8. Ejemplos de grafos dirigidos: grafo cíclico (a) y múltiplemente conexo (b).

Figura 5.1. Pantalla Inicial del Sistema Experto.

Figura 5.2. Tipos de conjuntos de Reglas que puede manejar el Sistema Experto.

Figura 5.3. Ventanas que muestran el proceso para agregar un nuevo conjunto de Reglas dentro del Sistema Experto.

Figura 5.4. Ventanas que muestran el proceso para modificar un conjunto de Reglas dentro del Sistema Experto.

Figura 5.5. Despliegue de las características que pertenecen a una regla.

Figura 5.6. Ventanas que muestran el proceso para añadir y/o modificar una Regla dentro del Sistema Experto.

Figura 5.7. Despliegue de la características de una Variable.

Figura 5.8. Ventanas que muestran el proceso para añadir y/o modificar una Variable dentro del Sistema Experto.

Figura 5.9. Ventanas que muestran un ejemplo de lo que es el Subsistema de Explicación.

Figura 5.10. Identificación de los elementos más importantes para crear una regla perteneciente a un conjunto del tipo probabilístico.

Figura 5.11. Motor de Inferencia para un conjunto de reglas del tipo probabilístico.

Figura 5.12. Subsistema de Aprendizaje para Método de Lógica Difusa.

Figura 5.13. Motor de Inferencia para un conjunto de reglas del tipo Lógica Difusa.

Figura 5.14. Motor de Inferencia para un conjunto de reglas del tipo Explicaciones.

Figura 6.1. Diagrama del Sistema del tanque a presión.

Figura 6.2. Árbol de fallos del sistema del tanque a presión.

Figura 6.3. Reglas encadenadas para el sistema del tanque a presión.

Figura 6.4. Grafo dirigido correspondiente al tanque de presión.

Figura 6.5. Selección del conjunto de Reglas en el Sistema Experto para el sistema del tanque a presión.

Figura 6.6. Características del conjunto de Reglas para el sistema del tanque a presión.

Figura 6.7. Reglas que pertenecen al conjunto del sistema del tanque a presión.

Figura 6.8. Variables que pertenecen al conjunto del sistema del tanque a presión.

Figura 6.9. a) Elección del conjunto de reglas para el problema del Tanque a Presión, b) El conjunto de reglas ahora es del Tipo Probabilístico.

Figura 6.10. Ventana para crear reglas para el Método Probabilístico sistema del tanque a presión.

Figura 6.11. Motor de Inferencia que muestra las variables para el problema del tanque a presión.

Figura 6.12. Diagrama del Sistema de distribución de Energía.

Figura 6.13. Árbol de fallos para el motor 2.

Figura 6.14. Reglas encadenadas para el motor 2.

Figura 6.15. Grafo dirigido para el sistema de distribución de energía (motor 2).

Tabla 6.2. Probabilidades condicionales de fallo de las variables del sistema de distribución de energía (motor 2).

Figura 6.16. Conjunto de Reglas para el sistema de distribución de energía (motor 2).

Figura 6.17. Reglas con sus probabilidades marginales y condicionales pertenecientes al sistema de distribución de energía (motor 2).

Figura 7.1. Probabilidades marginales iniciales de los nodos (cuando no hay evidencia) para el tanque a presión.

Figura 7.2. Probabilidades condicionales de los nodos dada la evidencia  $F=f$  y  $D=d$  para el sistema del tanque a presión.

Figura 7.3. Probabilidades condicionales de los nodos dada la evidencia  $F=f$ ,  $D=d$  y  $A=a$  para el sistema del tanque a presión.

Figura 7.4. Probabilidades marginales de los nodos cuando no hay evidencia para el sistema de distribución de energía (motor 2).

Figura 7.5. Probabilidades condicionales de los nodos dada la evidencia  $K=k$ , para el sistema de distribución de energía (motor 2).

Figura 7.6. Probabilidades condicionales de los nodos dada la evidencia  $K=k$  e  $I=i$ , para el sistema de distribución de energía (motor 2).

Figura 7.7. Probabilidades condicionales de los nodos dada la evidencia  $K=k$  e  $G=g$ , para el sistema de distribución de energía (motor 2).

Figura 7.8. Probabilidades condicionales de los nodos dada la evidencia  $G=g$ , para el sistema de distribución de energía (motor 2).

Figura 7.9. Probabilidades condicionales de los nodos dada la evidencia  $I=i$ , para el sistema de distribución de energía (motor 2).



# INTRODUCCIÓN

Desde la aparición de las computadoras hasta nuestros días, los científicos han realizado grandes esfuerzos para tratar de dar una cierta capacidad de decisión a estas máquinas, incluso un cierto grado de inteligencia.

A principios de la década de los 70, los investigadores del área de la Inteligencia Artificial (IA) reconocieron que los métodos y técnicas para resolver problemas generales desarrollados los años anteriores eran insuficientes para abordar aplicaciones de la vida diaria.

Ellos comprendieron que era necesario un conocimiento específico y limitado del dominio de las aplicaciones de interés, más que uno general que abarcara muchos dominios. Fue así que construyeron los primeros sistemas expertos en áreas concretas del saber, tales como MYCIN para el diagnóstico médico, DENDRAL para deducir la estructura de una composición química, PROSPECTOR para la exploración mineral y muchos más que tuvieron mucho éxito.

Un Sistema Experto en sí no tiene verdadera Inteligencia Artificial; más bien, es un sistema basado en el conocimiento que, mediante el buen diseño de su base de información y un adecuado motor de inferencias para manipular dichos datos proporciona una manera de determinar resoluciones finales dados ciertos criterios.

Los Sistemas Expertos son una herramienta poderosa en el apoyo o guía de los usuarios en los procesos que tienen una secuencia de pasos definida, pero que puede ser configurable.

En todo México, el uso de Sistemas Expertos en empresas es muy pobre, especialmente en el área de mantenimiento. Las empresas siguen utilizando el viejo paradigma de atención a los usuarios de manera presencial (personal, telefónica) o semipresencial (correos electrónicos, etc.). Estos modos de operación restringen seriamente la labor analítica de los programadores.

A continuación se presenta de manera general el contenido de la presente tesis.

En el Capítulo I (*Antecedentes y Conceptos Básico*) se detallan los conceptos básicos de la Inteligencia Artificial y Sistemas Expertos. Asimismo, se presenta detalladamente la historia de la Inteligencia Artificial, desde la propuesta inicial de Turing sobre la inteligencia de las máquinas y la propuesta del modelo de la neurona del cerebro humano por Mc Culloch y Pitts hasta los primeros Sistemas Expertos, que fueron los primeros resultados de la Inteligencia Artificial con aplicación en problemas más cotidianos.

En el Capítulo II (*Generalidades de los Sistemas Expertos*), se presenta una descripción de los Sistemas Expertos, así como también se muestran algunas comparaciones importantes entre diferentes tipos de expertos. Se menciona algunas de las ventajas y desventajas de estos tipos de expertos. Se muestran una serie de áreas en las que los Sistemas Expertos tienen presencia debido a las características y necesidades inherentes a ellas. Presenta también una descripción de los procesos en los cuales se involucran los Sistemas Expertos, tales como la interpretación, el diagnóstico, el diseño, la planificación, el control, la corrección, la simulación, entre otros. Finalmente, se da una idea de los componentes que integran un Sistema Experto, así como una descripción detallada de cada uno de estos componentes.

El Capítulo III (*Estructura de un Sistema Experto*), inicia con la descripción de algunas consideraciones que debe tomarse en cuenta antes de intentar arrancar un proyecto de implementación de un Sistema Experto en una empresa. Muestra un bosquejo general de las preguntas claves que los profesionales de la programación deben hacerse para evaluar la situación de la empresa respecto a este paradigma de solución de problemas. Continúa con una descripción de cómo se implementan los principales componentes de un Sistema Experto, así como dos de las principales metodologías para la creación de Sistemas Expertos. El capítulo continúa con una breve descripción de los principales lenguajes usados en la creación de los Sistemas Expertos concluyendo con una descripción de varias herramientas para el desarrollo de estos.

En el Capítulo IV (*Tipos de Sistemas Expertos*) se realiza una descripción detallada de los principales tipos de Sistemas Expertos: *Sistemas Basados en Reglas*, *Sistemas que manejan Incertidumbre* y *Sistemas Basados en Probabilidad*; finalizando el capítulo con la explicación de la base de conocimientos, el motor de inferencia y el subsistema de explicación para cada tipo de sistema.

En el Capítulo V (*Descripción de Sistema Experto Desarrollado*) explica el desarrollo de un Sistema Experto y del Motor de Inferencia en base a la propagación de evidencia y los poliárboles. Continuando con la estructura del sistema experto, así como los componentes y conjunto de reglas que se programaron. El capítulo concluye con la descripción del subsistema de aprendizaje y el subsistema de explicación.

En el Capítulo VI (*Aplicación del Sistema Experto*) se explica el problema a resolver, la selección de las variables y el conjunto de reglas utilizadas para la solución. Continuando con la explicación de la programación del método probabilístico y la propagación de evidencia.

En el Capítulo VII (*Resultados*) se presentan los resultados obtenidos por el Sistema Experto, utilizando el algoritmo de propagación de evidencia para poliárboles. Se presentan algunos casos de solución para cuando en el problema a solucionar se tiene o no evidencia disponible.

# OBJETIVO

*El objetivo general* de la realización de esta Tesis es la del estudio y aplicación de los Sistemas Expertos del Área de la Inteligencia Artificial, debido a que son una herramienta cuya utilidad ya está comprobada; y que sin embargo muchas personas desconocen y otras más aún no las aceptan. Se desea presentar los aspectos generales de un Sistema Experto (en adelante SE), sus orígenes, componentes, construcción y aplicaciones.

El *objetivo específico* que presenta esta Tesis es el de desarrollar un SE aplicado al área de mantenimiento a nivel industrial, en base al desarrollo de un software (programa) que utilice las técnicas de solución que manejan los diferentes tipos de SE; lo cual implica un nuevo algoritmo para la resolución de problemas de los ya existentes, y que podría significar mejoras en la obtención de resultados en lo que se refiere al tiempo de respuesta.

# I. ANTECEDENTES Y CONCEPTOS BÁSICOS

## 1.1 CONCEPTO DE INTELIGENCIA ARTIFICIAL

No existe una definición concreta de Inteligencia Artificial, pero la mayoría de los autores coinciden en que es, en esencia, lograr que una máquina tenga inteligencia propia, para algunos autores, la Inteligencia Artificial es el estudio de cómo hacer que los ordenadores hagan cosas que, en estos momentos, hace mejor el hombre<sup>1</sup>. Para otros, la Inteligencia Artificial (IA) es una ciencia que intenta la creación de programas para máquinas que imiten el comportamiento y la comprensión humana, que sean capaces de aprender, reconocer y pensar<sup>2</sup>.

Pero sea cual sea el concepto, la inteligencia artificial está orientada a conseguir que las máquinas realicen trabajos donde se aplique la inteligencia, el razonamiento y el conocimiento del ser humano.

## 1.2 HISTORIA DE LA INTELIGENCIA ARTIFICIAL

Desde tiempos muy remotos en la historia, el hombre ha soñado con crear máquinas con inteligencia propia, muestra de esto pueden ser: el mito del coloso de Rodas entre los griegos, el Turco de Von Kempelen que era una cabina de madera con un maniquí que simulaba a una persona jugando ajedrez, y la “máquina analítica” de Charles Babbage que calculaba logaritmos; sin embargo estos intentos resultaron infructuosos, hasta que en 1943 la base de la IA fue asentada, gracias a Warren McCulloch y Walter Pitts<sup>3</sup>, quienes propusieron un modelo de neurona de cerebro humano y animal. Estas neuronas nerviosas informáticas proporcionaron una representación simbólica de la actividad cerebral. Un tiempo después, Norbert Wiener tomó estas y otras ideas y las elaboró dentro de un mismo campo que se llamo Cibernética, a partir de la cual nacería, la Inteligencia Artificial.

<sup>1</sup> DE ÁVILA Ramos, Jorge. **Sistemas Expertos**. URL([http://www.lafacu.com/apuntes/informatica/sist\\_expe/](http://www.lafacu.com/apuntes/informatica/sist_expe/))

<sup>2</sup> CRIADO Briz, José Mario. **Introducción a los Sistemas Expertos**.  
URL([www.ingenieroseninformatica.org/recursos/tutoriales/sist\\_exp/index.php](http://www.ingenieroseninformatica.org/recursos/tutoriales/sist_exp/index.php))

<sup>3</sup> HURTADO Vega, José de Jesús. **Inteligencia Artificial**.  
URL(<http://www.itlp.edu.mx/publica/boletines/actual/inteligencia.html>)

Una vez que se asentaron las bases de la IA, los investigadores de dicha ciencia se mostraron optimistas sobre el futuro de la nueva ciencia que acababa de nacer; algunos de ellos expresaron diversas predicciones acerca de lo que podía desarrollarse dentro de la IA. Por ejemplo, Herbert Simon<sup>4</sup> en 1958 predijo que en un lapso de 10 años una computadora llegaría a ser campeona de ajedrez, y que sería posible realizar mediante la máquina la demostración de un nuevo e importante teorema matemático. Sin embargo, el obstáculo que enfrentó la mayoría de los proyectos de investigación en IA consistió en que aquellos métodos que demostraban su funcionamiento en uno o dos ejemplos sencillos, fallaban cuando se utilizaban en problemas más variados o de mayor dificultad:

1. El reconocimiento de que el pensamiento puede ocurrir fuera del cerebro, es decir, en máquinas.
2. La presuposición de que el pensamiento puede ser comprendido de manera formal y científica.
3. La presuposición de que la mejor forma de entenderlo es a través de computadoras digitales.<sup>5</sup>

Los primeros programas de IA fallaron debido a que contaban con poco o ningún conocimiento de la materia objeto de estudio, un ejemplo de esto es el programa ELIZA de Weizenbaum<sup>6</sup>, el cual aparentemente podía entablar una conversación seria sobre cualquier tema, sin embargo, lo único que hacía era tomar prestadas y manipular las oraciones que mediante un teclado le proporcionaba una persona. Otro problema muy común en los primeros programas que intentaban simular el pensamiento humano era la intratabilidad de muchos de los problemas que estaban intentando resolver mediante IA.

---

<sup>4</sup> ELGUEA, Javier. **Inteligencia artificial y psicología: la concepción contemporánea de la mente humana.** URL([http://www.hemerodigital.unam.mx/ANUIES/itam/estudio/estudio10/sec\\_16.html](http://www.hemerodigital.unam.mx/ANUIES/itam/estudio/estudio10/sec_16.html))

<sup>5</sup> HURTADO Vega, José de Jesús. *Op. cit.*

<sup>6</sup> ELGUEA, Javier. *Op. cit.*

La mayoría de los primeros programas de IA se basaban en la presentación de las características básicas de un problema y se sometían a prueba diversos pasos, hasta que se llegara a encontrar aquella combinación de estos que produjeran la solución esperada. Sin embargo, si una combinación sencilla no llegaba a la solución del problema a resolver, los primeros programas de IA no eran capaces de probar con combinaciones más complejas, por lo que se limitaban a mostrar un mensaje que indicaba que el problema no se podía resolver, cuando lo que en verdad sucedía era que el programa no estaba dotado con la suficiente “inteligencia” para resolver dicho problema.

Entonces algunos investigadores cambiaron el enfoque del problema: ahora se dedicaban a resolver problemas sobre un área específica intentando simular el razonamiento humano. En vez de dedicarse a computarizar la inteligencia general, se centraron en dominios de conocimientos muy concretos. De esta manera nacieron los Sistemas Expertos.

El programa DENDRAL de Buchanan<sup>7</sup> constituye uno de los primeros ejemplos de este enfoque. Fue diseñado en Stanford, donde Ed Feigenbaum, Bruce Buchanan y Joshua Lederberg colaboraron en la solución del problema de inferir una estructura molecular a partir de la información proporcionada por un espectrómetro de masas. El programa se alimentaba con la fórmula elemental de la molécula ( $C_6H_{13}NO_2$ ). La primera versión del programa generaba todas las posibles estructuras que correspondieran a la fórmula, luego predecía el espectro de masa que se observaría en cada caso, y comparaba estos con el espectro real. Así fue como comenzaron los primeros desarrollos e investigaciones sobre los sistemas expertos.

---

<sup>7</sup> ELGUEA, Javier. **Inteligencia artificial y psicología: la concepción contemporánea de la mente humana.**  
URL([http://www.hemerodigital.unam.mx/ANUIES/itam/estudio/estudio10/sec\\_16.html](http://www.hemerodigital.unam.mx/ANUIES/itam/estudio/estudio10/sec_16.html))

### 1.3 DESARROLLO HISTÓRICO DE LOS SISTEMAS EXPERTOS

Los Sistemas Expertos como tales, surgen a mediados de los años 60's; en esos tiempos, se creía que bastaban unas pocas leyes de razonamiento junto con potentes ordenadores para producir resultados brillantes. Los primeros investigadores que desarrollaron programas basados en leyes de razonamiento fueron Alan Newell y Herbert Simon, quienes desarrollaron el GPS (General Problem Solver)<sup>8</sup>. Este sistema era capaz de resolver problemas como el de las torres de Hanoi y otros similares, a través de criptoaritmética. Sin embargo, este programa no podía resolver problemas más “cotidianos” y reales, como, por ejemplo, dar un diagnóstico médico.

El primer SE que se aplicó a problemas más reales fue desarrollado en 1965 con el fin de identificar estructuras químicas: el programa se llamó DENDRAL<sup>9</sup>. Lo que este SE hacía, al igual que lo hacían los expertos de entonces, era tomar unas hipótesis relevantes como soluciones posibles, y someterlas a prueba comparándolas con los datos. El nombre DENDRAL significa *árbol* en griego. Debido a esto, el programa fue bautizado así porque su principal tarea era buscar en un árbol de posibilidades la estructura del compuesto.

El siguiente SE que causó gran impacto fue el MYCIN, en 1972, pues su aplicación era detectar trastornos en la sangre y recetar los medicamentos requeridos. Fue tal el éxito de MYCIN que incluso se llegó a utilizar en algunos hospitales. Para 1973, se creó TIERESIAS, cuya función era la de servir de intérprete o interfaz entre los especialistas que manejaban MYCIN cuando introducían nuevos conocimientos. TIERESIAS entraba en acción cuando MYCIN cometía un error en un diagnóstico, por la falta de información o por alguna falla en el árbol de desarrollo de teorías, corrigiendo la regla que generaba el resultado o bien destruyéndola.<sup>10</sup>

<sup>8</sup> RUSSELL, Stuart y Peter Norving. **Inteligencia Artificial: Un enfoque moderno.**

<sup>9</sup> CRIADO Briz, José Mario. **Introducción a los Sistemas Expertos.**

<sup>10</sup> SAMPER Márquez, Juan José. **Sistemas Expertos: El conocimiento al poder.**



Para los años comprendidos entre 1980 y 1985 se crearon diversos sistemas expertos, tales como DELTA<sup>11</sup> de la General Electric Company, el cual se encargaba de reparar locomotoras diesel y eléctricas, o como “Aldo en Disco”, que reparaba calderas hidrostáticas giratorias usadas para eliminación de bacterias.

En esa misma época surgen las empresas dedicadas a desarrollar Sistemas Expertos, las cuales supusieron en conjunto una inversión total de más de 300 millones de dólares. Los productos más importantes que creaban estas nuevas compañías eran las “máquinas LISP”<sup>11</sup>, las cuales consistían en unos ordenadores que ejecutaban programas LISP con la misma rapidez que en un ordenador central.

---

<sup>11</sup> ROOS, Rita. **Historia de los Sistemas Expertos.**  
URL(<http://www.monografias.com/trabajos10/exper/exper.shtml>).

A continuación la Tabla 1.1 presenta de manera resumida los primeros Sistemas Expertos y sus aplicaciones.

<b>Sistema</b>	<b>Fecha</b>	<b>Autor</b>	<b>Aplicación</b>
DENDRAL	1965	Stanford	Deduce información sobre estructuras químicas
Macsyma	1965	MIT	Análisis matemático complejo
HearSay	1965	Carnegie-Mellon	Interpreta en lenguaje natural un subconjunto del idioma
PROLOG	1970	Universidad Aix-Marseille	Herramienta para desarrollo de SE
Mycin	1972	Stanford	Diagnóstico de enfermedades
Tieresias	1972	Stanford	Herramienta para la transformación de conocimientos
Prospector	1972	Stanford	Exploración mineral
Age	1973	Stanford	Herramienta para generar SE
OPS 5	1974	Carnegie-Mellon	Herramienta para desarrollo de SE
Cadeceus	1975	University of Pitsyburg	Diagnóstico para medicina interna
Rosie y XCON	1978	Rand	Desarrollo de SE
DELTA	1980	General Electric	Reparación de locomotoras diesel y eléctricas
CLIPS	1984	NASA	Producción y ejecución de SE
SEILUM	Década de los 90's	UNAM	SE para iluminación industrial
fuzzyCLIPS	2000	NASA	Extensión de CLIPS para tratar incertidumbre e imprecisión
EXCYSCORVID	2000 a la fecha	Knowledge Automation Expert System	Múltiples aplicaciones para NO expertos. Ejemplos: - Diagnóstico y reparación de equipos y procesos  - Configuración de sistemas complejos. Etc.

**Tabla 1.1. Sistemas Expertos y sus aplicaciones.**

## **II. GENERALIDADES DE LOS SISTEMAS EXPERTOS**

### **2.1 ¿QUÉ SON LOS SISTEMAS EXPERTOS?**

Se puede decir que los Sistemas Expertos son el primer resultado operacional de la Inteligencia Artificial, pues logran resolver problemas a través del conocimiento y raciocinio de igual forma como lo hace un experto humano.

Un Sistema Experto (SE), es básicamente un programa de computadora basado en conocimientos y raciocinio que lleva a cabo tareas que generalmente realiza un experto humano; es decir, es un programa que imita el comportamiento humano en el sentido de que se utiliza la información que le es proporcionada para poder dar una opinión sobre un tema en especial. Otros autores lo definen de distintas formas: un SE es un programa de computadora interactivo que contiene la experiencia, conocimiento y habilidad propios de una persona o grupos de personas especialistas en un área particular del conocimiento humano, de manera que permitan resolver problemas específicos de manera inteligente y satisfactoria. La tarea principal de un SE es tratar de aconsejar al usuario.

Los usuarios que introducen la información al SE son en realidad los expertos humanos, y tratan a su vez de estructurar los conocimientos que poseen para ponerlos entonces a disposición del sistema. Los SE son útiles para resolver problemas que se basan en conocimiento.

Desde su aparición, a mediados de 1960, los Sistemas Expertos se han definido como aquellos programas que se basan en el conocimiento y tratan de imitar el razonamiento de un experto para resolver un problema de problema definido. Su comportamiento se basa generalmente en reglas, es decir, se basa en conocimientos previamente definidos, y mediante estos conocimientos, los SE son capaces de tomar decisiones. Sería ilógico pensar que solo existe una definición de Sistemas Expertos, ya que tanto los SE como la propia IA han ido evolucionando a la par a través de los años.

En el Congreso Mundial de IA en 1991 Feigenbaum definió a los SE como:

- Un programa de computador inteligente que usa el conocimiento y procedimientos de inferencia para resolver problemas que son lo suficientemente difíciles como para requerir la intervención de un experto humano para su resolución.

Sin embargo con los avances conseguidos hasta ahora esta definición ha cambiado, actualmente un SE tiene la definición siguiente:

- Un SE es un sistema informático que simula los procesos de aprendizaje, memorización, razonamiento, comunicación y acción de un experto humano en una determinada rama de la ciencia.

Las características mencionadas en las definiciones anteriores le permiten a un SE almacenar datos y conocimientos, sacar conclusiones lógicas, ser capaces de tomar decisiones, aprender, comunicarse con expertos humanos o con otros SE, explicar el razonamiento de su decisión y realizar acciones como consecuencia a todo lo anterior.

Un problema se presta a ser resuelto usando un Sistema Experto cuando:

- Una solución del problema tiene una rentabilidad tan alta que justifica el desarrollo de un sistema, pues las soluciones son necesidades del área y no se ha trabajado en otros métodos para obtenerla.
- El problema puede resolverse sólo por un conocimiento experto que puede dar forma a los conocimientos necesarios para resolver el problema, y la intervención del experto dará al sistema la experiencia necesaria.
- Se tiene a un experto que puede dar forma a los conocimientos necesarios para resolver el problema. La intervención de este experto dará al sistema la experiencia que necesita.

- El problema puede o no tener una solución única. Los Sistemas Expertos funcionan mejor con problemas que tienen un cierto número de soluciones aceptables.
- El problema cambia rápidamente, o bien el conocimiento es el que cambia rápidamente, o sus soluciones son las que cambian constantemente.
- El desarrollo de un SE no se considera que está acabado una vez que funcione este, sino que continua desarrollando y actualizando tanto el conocimiento del sistema como los métodos de procesamiento, quedando reflejados los progresos o modificaciones en el campo, área o sistema.

## 2.2 LOS EXPERTOS HUMANOS

Un experto humano es una persona que es competente en un área determinada del conocimiento o del saber. Un experto humano es alguien que sabe mucho sobre un tema determinado y que puede dar un consejo adecuado. Esta experiencia sólo se adquiere tras un largo aprendizaje y a base de mucha experiencia. Los expertos humanos tienen las siguientes características generales:

- Son personas raras, tanto por su escaso número como su comportamiento poco “ortodoxo” e “incomprensible” frente a los problemas con los que se enfrentan.
- Son caros por dos motivos: por su escaso número y por necesitar un largo periodo de aprendizaje.
- No están siempre disponibles, pues son humanos y cuando se jubilan o se mueren se llevan con ellos todos sus conocimientos. Es por eso que tradicionalmente están acompañados de un “aprendiz”.

La forma más rápida de formar a un experto es mediante el aprendizaje formal o académico (“conocimiento profundo”) en un principio, posteriormente un aprendizaje informal o práctico (“conocimiento informal”).

En la tabla 2.1 se mencionan algunas características notables que destaca a una persona que tiene un dominio sobre un tema en particular (Experto) de una persona que no tiene tal dominio (No experto).

CARACTERÍSTICA	<i>Experto</i>	NO EXPERTO
Tiempo de Resolución	Pequeño	Grande
Eficacia Resolutiva	Alta	Baja
Organización	Alta	Baja
Estrategias y Tácticas	Sí	No
Búsqueda de Soluciones	Heurística	No Heurística
Cálculos Aproximados	Sí	No

**Tabla 2.1. Diferencias entre un experto y un no experto humano.**

### 2.3 DIFERENCIAS ENTRE UN SE Y UN EXPERTO HUMANO

En la tabla 2.2 se muestran las diferencias más importantes que existen entre una persona que se considera Experto y un Sistema Experto (software). Las definiciones de SE y de experto humano se vieron anteriormente.

CARACTERÍSTICA	SISTEMA EXPERTO	EXPERTO HUMANO
Conocimiento	Adquirido	Adquirido + Innato
Adquisición del conocimiento	Teórico	Teórico + Práctico
Campo	Único	Múltiples
Explicación	Siempre	A veces
Limitación de capacidad	Sí	Sí, no valuable
Reproducibile	Sí, idéntico	No
Vida	Infinita	Finita

**Tabla 2.2. Diferencias entre un Sistema Experto y un Experto Humano.**

## 2.4 VENTAJAS DE LOS SISTEMAS EXPERTOS

Las ventajas que se presentan a continuación son en comparación con los expertos humanos:

- Están siempre disponibles a cualquier hora del día y de la noche, y de forma ininterrumpida.
- Pueden duplicarse (lo que permite tener tantos SE como se necesiten).
- Pueden situarse en el mismo lugar donde sean necesarios.
- Son fáciles de reprogramar.
- Pueden perdurar y crecer en el tiempo de forma indefinida.
- Pueden ser consultados por personas u otros sistemas informáticos.

## 2.5 DIFERENCIAS ENTRE UN SE Y UN PROGRAMA TRADICIONAL

En la tabla 2.3 se muestran la diferencia que hay entre un Sistema Experto (que ya se definió anteriormente) y un Programa Tradicional, este último puede ser un *programa orientado a procedimientos o algoritmos*, un *programa interactivo* (Matlab) o bien un *programa típico de Windows* (Excel, Power Point, etc.). Las características y diferencia entre cada uno de estos últimos programas se ven en los capítulos siguientes.

Las características que se muestran tienen que ver con la *forma de ejecución* del programa.

CARACTERÍSTICA	SISTEMA EXPERTO	PROGRAMA TRADICIONAL
Conocimiento	En programa e independiente	En programa y circuitos
Tipo de datos	Simbólicos	Numéricos
Resolución	Heurística	Combinatoria
Def. problema	Declarativa	Procedimental

Control	Independiente. No secuencial	Dependiente. Secuencial
Conocimientos	Imprecisos	Precisos
Modificaciones	Frecuentes	Raras
Explicaciones	Sí	No
Solución	Satisfactoria	Optima
Justificación	Sí	No
Resolución	Área limitada	Específico
Comunicación	Independiente	En programa

**Tabla 2.3. Diferencias entre un Sistema Experto y un Programa Tradicional.**

## 2.6 TAREAS QUE REALIZAN LOS SISTEMAS EXPERTOS

### *INTERPRETACIÓN, DIAGNÓSTICO Y MONITOREO*

La interpretación consiste en encontrar el significado de los datos de entrada obtenidos por sensores o introducidos por el usuario.

Con frecuencia aparecen datos contradictorios, incompletos o inexactos, por lo que hay que dotar al SE de conocimiento para resolver un problema de este tipo, y que de todas formas llegue a una resolución. Existen dos tipos de interpretación:

- **Análisis:** La interpretación de datos se obtiene mediante la separación o distinción de las partes que forman los datos.
- **Síntesis:** La interpretación de los datos se obtiene mediante la combinación de los mismos.

El diagnóstico consiste en identificar las causas internas que provocan un problema, disfunción a partir de una serie de datos o síntomas que son consecuencia de la misma y que son observables.



Los SE en diagnóstico encuentran múltiples dificultades a la hora de realizar su tarea como son las siguientes:

- **Manifestaciones nuevas.** Es decir, síntomas que no se habían observado con anterioridad.
- Causas nuevas.
- Manifestaciones debidas a varias causas.
- Datos Inaccesibles, caros o de obtención peligrosa.
- Fallos de aparición intermitente.
- Existencia de varios fallos simultáneos o en cadena.

La monitorización es un caso particular de la interpretación, y consiste en la comparación continua de los valores de las señales o datos de entrada y unos valores que actúan como criterios de normalidad o estándares.

En el campo del mantenimiento predictivo los Sistemas Expertos se utilizan fundamentalmente como herramienta de diagnóstico. Se trata de que el programa pueda determinar en cada momento el estado de funcionamiento de sistemas complejos, anticipándose a los posibles incidentes que pudiera acontecer. Así, usando un modelo computacional del razonamiento de un experto humano, proporciona los mismos resultados que alcanzaría dicho experto.

## *DISEÑO Y SELECCIÓN*

El diseño se puede realizar de distintas formas:

- ✓ El diseño en ingeniería es el uso de principios científicos, información técnica e imaginación en la definición de una estructura mecánica, máquina o sistema que ejecute funciones específicas con el máximo de economía y eficiencia.
- ✓ El diseño industrial busca rectificar las omisiones de la ingeniería, es un intento consistente de traer forma y orden visual a la ingeniería de hardware donde la tecnología no provee estas características.

- ✓ Diseño es el proceso de especificar una descripción de un artefacto que satisface varias características desde un número de fuentes de conocimiento.

Los SE en diseño ven este proceso como un problema de búsqueda de una solución óptima o adecuada. Las soluciones alternas pueden ser conocidas de antemano (problemas de derivación) o se pueden generar automáticamente (problemas de formulación).

Los SE prueban distintos diseños para verificar cuáles de ellos cumplen con los requerimientos solicitados por el usuario, ésta técnica es llamada “generación y prueba”, por lo tanto estos SE son llamados de selección. En áreas de aplicación, la prueba se termina cuando se encuentra la primera solución; sin embargo, existen problemas más complejos en los que el objetivo es encontrar la solución más óptima.

La utilidad de los programas basados en conocimientos aplicados a la ciencia y a la ingeniería no se limita al análisis sofisticado, muchos programas recientes han empezado a trabajar en la síntesis.

Existen programas que diseñan dispositivos simples y después buscan oportunidades de reducción de costos o de reducción de componentes.

### *PLANIFICACIÓN*

La planificación es la realización de planes o secuencias de acciones y es un caso particular de la simulación. Está compuesto por un simulador y un sistema de control. El efecto final es la ordenación de un conjunto de acciones con el fin de conseguir un objetivo global (costo mínimo, tiempo mínimo, etc.).

Los problemas que presenta la planificación mediante un SE son los siguientes:

- ✗ Existen consecuencias no previsibles, de forma que hay que explorar y explicar varios planes.

- ✗ Existen muchas consideraciones que deben ser valoradas o incluirles un factor de paso.
- ✗ Trabajo frecuente con incertidumbre, pues la mayoría de los datos con los que se trabaja son más o menos probables pero no seguros.
- ✗ Es necesario hacer uso de fuentes diversas tales como bases de datos.

### *CONTROL*

Un sistema de control participa en la realización de las tareas de interpretación, diagnóstico y reparación de forma secuencial. Con ello se consigue guiar un proceso o sistema.

Los sistemas de control son complejos debido al número de funciones que deben manejar y el gran número de factores que deben considerar; esta complejidad creciente es otra de las razones que apuntan al uso del conocimiento, y por tanto de los SE.

Cabe aclarar que los sistemas de control pueden ser de lazo abierto, si en el mismo la retroalimentación o el paso de un proceso a otro lo realiza el operador, o en lazo cerrado si no tiene que intervenir el operador en ninguna parte del mismo.

### *REPARACIÓN, CORRECCIÓN O TERAPIA*

La reparación, corrección, terapia o tratamiento consiste en la proposición de las acciones correctivas necesarias para la resolución de un problema.

Los SE en reparación tiene que cumplir diversos objetivos, como son:

- Reparación lo más rápida y económicamente posible.
- Orden de las reparaciones cuando hay que realizar varias.
- Evitar los efectos secundarios de la reparación, es decir la aparición de nuevas fallas por la reparación.

## *SIMULACIÓN, PRONÓSTICO O PREDICCIÓN*

La simulación es una técnica consistente en crear modelos basados en hechos, observaciones e interpretaciones, sobre la computadora, a fin de estudiar el comportamiento de los mismos mediante la observación de las salidas para un conjunto de entradas.

Las técnicas tradicionales de simulación requieren modelos matemáticos y lógicos que describen el comportamiento del sistema en estudio.

El empleo de los SE para la simulación viene motivado por la principal característica de los SE, que es la capacidad para la simulación del comportamiento de un experto humano, que es un proceso complejo.

En la aplicación de los SE para simulación hay que diferenciar tres configuraciones posibles:

1. Un SE puede disponer de un simulador con el fin de comprobar las soluciones y en su caso rectificar el proceso que sigue.
2. Un sistema de simulación puede contener como parte del mismo a un SE y por lo tanto el SE no tiene que ser necesariamente de simulación.
3. Un SE puede controlar un proceso de simulación, es decir que el modelo está en la base de conocimiento del SE y su evolución es función de la base de los hechos, la base de conocimientos y el motor de inferencia, y no de un conjunto de ecuaciones aritmético-lógicas.

Los sistemas de pronóstico deducen consecuencias posibles a partir de una situación. Su objetivo es determinar el curso del futuro en función de información sobre pasado y presente. Esto abarca diversos problemas, tales como predicciones meteorológicas, predicciones demográficas, etc.

## *INSTRUCCIÓN*

Un sistema de instrucción, realiza un seguimiento del proceso de aprendizaje de un estudiante. El sistema detecta errores de los estudiantes e identifica la solución adecuada, es decir, desarrolla un plan de enseñanza para facilitar el proceso de aprendizaje y la corrección de errores.

## *RECUPERACIÓN DE INFORMACIÓN*

Los Sistemas Expertos, con su capacidad para combinar información y reglas de actuación, han sido vistos como una de las posibles soluciones de tratamiento y recuperación de información, no sólo documental.

Lo que diferencia a estos sistemas de un sistema tradicional de recuperación de información es que estos últimos son capaces de recuperar lo que existe explícitamente, mientras que un SE debe ser capaz de generar información no explícita razonando con los elementos que se le dan. Pero la capacidad de los SE en el ámbito de la recuperación de la información no se limita a la recuperación. Pueden utilizarse para ayudar al usuario, en selección de recursos de información, en filtrado de respuestas, etc. Un SE puede actuar como un intermediario inteligente que guía y apoya el trabajo del usuario final.

## **2.7 CAMPOS DE APLICACIÓN DE LOS SISTEMAS EXPERTOS**

### *MEDICINA*

Los SE enfrentan tareas tales como la resolución de problemas, razonamiento automático y aprendizaje automático. Es típico el estudio de estos sistemas inteligentes en dominios específicos del conocimiento, como la medicina.

Los programas en esta área se pueden clasificar en:

- Métodos de contestación prefijada, formados por algoritmos aritméticos lógicos, en los cuales el control y el conocimiento están juntos y están escritos en lenguajes procedimentales.
- Métodos estadísticos que se clasificaban en Bayesianos, de análisis discriminantes y análisis secuencial.

### *INDUSTRIA*

Los SE en la industria se aplican principalmente en:

- Diagnóstico de control de calidad.
- Detección y actuación en caso de alarmas y emergencias.
- Configuración de equipos y sistemas bajo demanda.
- Control de procesos industriales.
- Mantenimiento de equipo.

### *ELECTRÓNICA, INFORMÁTICA Y TELECOMUNICACIONES*

Las aplicaciones principales de los SE son:

- Diseño de circuitos de alto grado de integración.
- Sistemas inteligentes de autodiagnóstico.
- Configuración de equipos y sistemas.
- Control de redes de comunicación.
- Programación automática.
- Ajuste de equipos y sistemas.
- Optimización de programas de computadoras.

### *MILITAR*

Las aplicaciones se centran en:

- Guiado de vehículos y proyectiles de forma semiautomática.
- Planificación estratégica.
- Reconocimiento automático de blancos y valoración de los mismos.
- Reconocimiento de planes del enemigo.
- Optimización de carga.

## *ROBÓTICA*

Aun cuando los robots no son como se muestran en las películas, realmente pueden llegar a realizar actividades sorprendentes, sobre todo si son utilizados en la fabricación de productos, donde las tareas son repetitivas y aburridas.

Los robots son muy solicitados en ambientes peligrosos para el ser humano, como en el manejo de explosivos, altas temperaturas, atmósfera sin la cantidad adecuada de oxígeno y en general bajo cualquier situación donde se pueda deteriorar la salud.

La mayoría de los robots tienen un brazo con varias uniones móviles, donde todos sus elementos son controlados por un sistema de control programado para realizar varias tareas bajo una secuencia de pasos preestablecidos. Los investigadores de IA pretenden adicionar al robot métodos y técnicas que le permitan actuar como si tuviera un pequeño grado de inteligencia, lo cual pretenden lograr con la conjunción de todas las áreas de la IA.

## *OTROS CAMPOS DE APLICACIÓN*

- Aeronáutica.
- Agricultura.
- Arqueología.
- Educación.
- Meteorología.
- Geología.

- Química.
- Transportes.
- Ventas.

## 2.8 COMPONENTES DE UN SISTEMA EXPERTO

Los dos componentes principales de cualquier Sistema Experto son una BASE DE CONOCIMIENTOS y un PROGRAMA DE INFERENCIA, o también llamado MOTOR DE INFERENCIA.

1) La base de conocimientos contiene el conocimiento especializado extraído del experto en el dominio. Es decir, contiene conocimiento general sobre el dominio en el que se trabaja. El método más común para representar el conocimiento es mediante reglas de producción. El dominio de conocimiento representado se divide en pequeñas fracciones de conocimientos o reglas.

Una característica muy importante es que la base de conocimientos es independiente del mecanismo de inferencia que se utiliza para resolver los problemas. De esta forma, cuando los conocimientos almacenados se han quedado obsoletos, o cuando se dispone de nuevos conocimientos, es relativamente fácil añadir reglas nuevas, eliminar las antiguas o corregir errores existentes. No es necesario reprogramar todo el Sistema Experto.

Las reglas suelen almacenarse en alguna secuencia jerárquica lógica, pero esto no es estrictamente necesario. Se pueden tener en cualquier secuencia y el motor de inferencia las usará en el orden adecuado que necesite para resolver el problema.

La base de datos o base de hechos es una parte de la memoria del ordenador que se utiliza para almacenar los datos recibidos inicialmente para la resolución de un problema. Contiene conocimiento sobre el caso concreto en que trabaja. También se registrarán en ella conclusiones intermedias y los datos generados en el proceso de



inferencia. Al memorizar todos los resultados intermedios, se puede utilizar para explicar las deducciones y el comportamiento del sistema.

2) El motor de inferencias es un programa que controla el proceso de razonamiento que seguirá el SE. La interfaz de usuario permite que el usuario pueda describir el problema al SE. Interpreta sus preguntas, los comandos y la información ofrecida. A la inversa, formula la información generada por el sistema incluyendo respuestas a las preguntas, explicaciones y justificaciones. También puede solicitar más información al SE si le es necesaria. En algunos sistemas se utilizan técnicas de tratamiento del lenguaje natural para mejorar entre el usuario y el SE.

3) 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 basándose en el consejo del SE. Además, de esta forma, y con el tiempo suficiente, los usuarios pueden convertirse en especialistas de la materia, al asimilar el proceso de razonamiento seguido por el sistema. El subsistema de explicación también puede usarse para depurar el SE durante su desarrollo.

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

Muchos investigadores hacen representaciones visuales de los componentes de un SE. Por ejemplo, De Miguel<sup>12</sup> muestra su diagrama de un SE en la figura 2.1.

---

<sup>12</sup> CRIADO Briz, José Mario. **Introducción a los Sistemas Expertos.**  
[URL\(www.ingenieroseninformatica.org/recursos/tutoriales/sist\\_exp/index.php\)](http://www.ingenieroseninformatica.org/recursos/tutoriales/sist_exp/index.php)

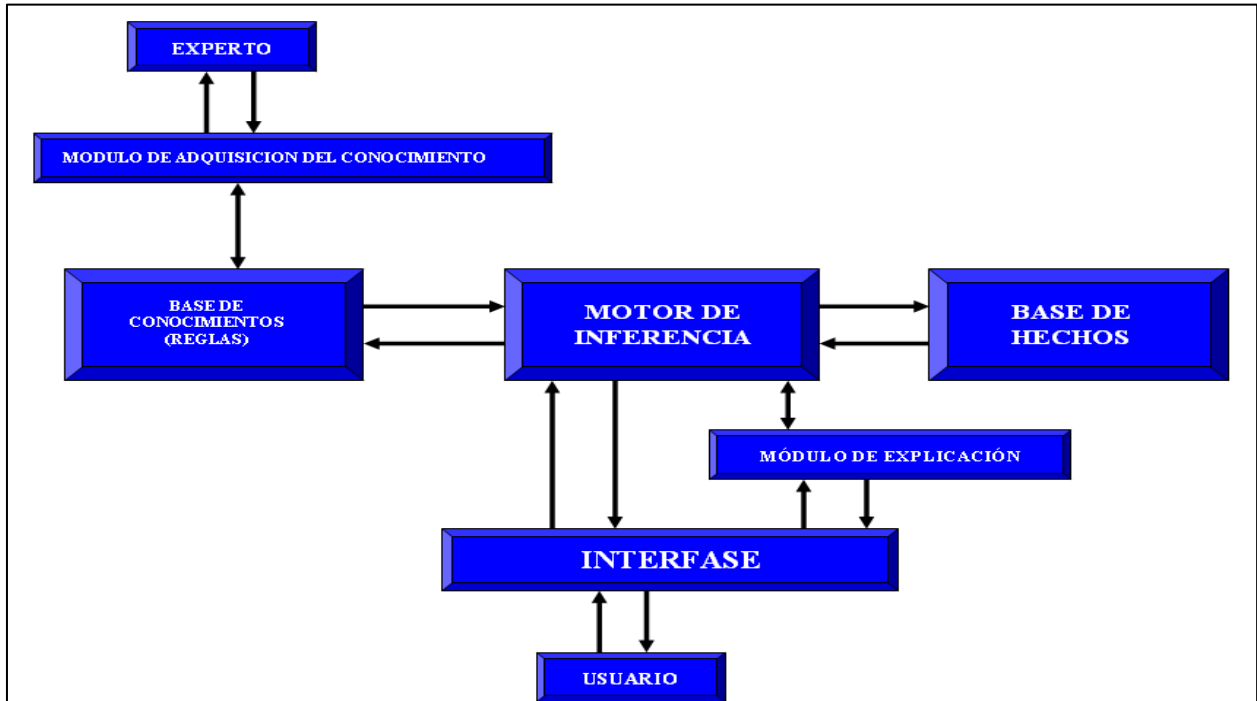


Figura 2.1. Estructura de un Sistema Experto según De Miguel.

Otra representación de componentes de un Sistema Experto es la de Samper<sup>13</sup> quien nos dice que no existe una estructura común para un Sistema Experto. Sin embargo, la mayoría de los sistemas expertos tienen ciertos componentes básicos: base de conocimientos, motor de inferencia, base de datos e interfaz con el usuario. Muchos tienen, además, un módulo de explicación y un módulo de adquisición del conocimiento. La figura 2.2 muestra la estructura de un SE ideal.

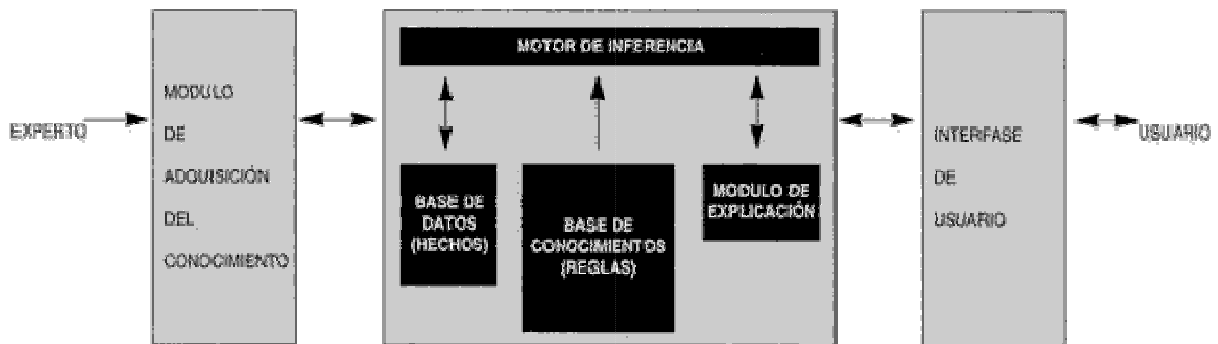


Figura 2.2. Estructura de un Sistema Experto ideal para Samper.

<sup>13</sup> **Sistemas Expertos, Principios y programación** 3ª Ed. Joseph Giatarrano, Gary Riley International Thomson Editores

### **III. ESTRUCTURA DE UN SISTEMA EXPERTO**

La construcción de un SE no es una tarea sencilla, debido a que involucra mucha participación de distintas personas, cada una de las cuales aportará algo para el SE a desarrollar sea robusto y fácil de usar y mantener. Además se deben hacer varias elecciones en cuanto al desarrollo del Sistema Experto.

La primera decisión consiste en determinar si se comenzará el SE desde cero, se deberá entonces determinar qué metodología utilizar, es decir, determinar la guía para el desarrollo del SE, cómo se implementará la base de conocimientos y el motor de inferencia, principalmente; y como complemento se debe elegir el lenguaje que se va a utilizar para el proyecto.

#### **3.1 INTRODUCCIÓN DE LOS SISTEMAS EXPERTOS EN LA EMPRESA**

Para empezar como su nombre lo indica, un SE está encaminado a los conocimientos de un experto. Un experto como se ha explicado anteriormente es aquél que domina perfectamente un tema por elemental que este sea.

La identificación de un problema en la empresa puede hacerse buscando a los expertos cuya desaparición causaría a la empresa una pérdida.

Sin llegar al extremo de la pérdida del experto. Se pueden mencionar algunos rasgos que pueden implicar una especie de pérdida del experto por parte de la empresa:

1. La experiencia existe pero la difusión se distribuye mal ya que el experto difícilmente esta accesible.
2. Varios expertos son necesarios y no se puede confrontar a todos ellos al mismo tiempo.
3. Los métodos de resolución de problemas o la aplicación de reglas difieren según los sectores y las personas, que las interpretan a su manera.

Finalmente, se puede proceder a una encuesta en todos los niveles de la empresa (agrupando por sector, jerarquía, etc.); una vez hecho esto, se debe definir el tipo de problema que se tiene. Existen muchos tipos como:

Problemas que tienen solución pero con tiempos largos de espera. Un SE puede mejorar notablemente estos procesos. Cuando la información es muy superficial o intuitiva un SE no es una buena opción. Y, además, existe el tipo de problemas que requieren del manejo de grandes cantidades de conocimiento claro y cualitativo (este es el tipo de problemas más aprovechados en la elección de un SE).

Una vez que se detectó el problema, simplemente debe justificarse ante los intereses de la empresa. Debemos cuestionarnos si los conocimientos que se manejarán constituyen un patrimonio para la empresa tal que amerite el desarrollo del sistema de información y si realmente perjudica a la empresa que alguno de los problemas que se intentan resolver mediante sistemas expertos no se resuelve rápida o adecuadamente.

Además de las cuestiones anteriores, se puede ver la viabilidad de la aplicación de un SE si el proceso en el cual queremos implantar un SE cumple las siguientes condiciones:

1. Evitar fallos en labores rutinarias complejas.
2. Ampliar de forma más rápida los conocimientos de los especialistas.

Diagnosticar fallos con mayor rapidez y conseguir tareas de planificación más completas y consistentes.

Antes de introducir un Sistema Experto en una empresa se deben responder una serie de preguntas referentes a los siguientes aspectos:

## QUIÉN

- ¿A quién concierne?
- ¿Quién está implicado en la decisión?
- ¿Quién va a crear el sistema (expertos, ingenieros de conocimiento, usuarios,...)?
- ¿Quién lo empleará (Área, función y nivel)?

## QUÉ

- ¿Para que sirve el sistema?
- ¿Cuál es su finalidad?
- ¿Va a remplazar a los operadores humanos o por el contrario revalorizará el trabajo de los que asista en su tarea (por ejemplo, haciéndoles que no olviden nada y presentándoles información oportuna)?

## DÓNDE

- ¿Dónde va a ser utilizado?
- ¿Va a ser repartido en varias copias o se utilizará localmente?
- ¿Se utilizará en el interior de la empresa (estudios, oficinas, etc.) o en el exterior (representantes, clientes, etc.)?

## CÓMO

- ¿Cómo va a utilizarse?
- ¿Se utilizará como un servicio libre o por personas que tendrán esa tarea exclusivamente?
- ¿Funcionará de forma autónoma?
- ¿Va a ser utilizado por expertos para mejorar su rendimiento?
- ¿Va a ser utilizado por personal especialmente preparado para manejarlo?
- Si trabajará en tiempo real, ¿qué carga representará para el servidor y el cliente?
- ¿Cuáles serán sus relaciones con otros sistemas: bases de datos, tableros, etc.?
- ¿Existen problemas de datos confidenciales?
- ¿Funcionará con medios tradicionales?

## CUÁNDO

- ¿En qué plazo desea que se realice el sistema?
- ¿Se empleará escasa o frecuentemente?

Debido a que la estructuración e implementación del conocimiento del experto requiere una gran cantidad de trabajo, sólo valdrá la pena realizar el esfuerzo de crear un Sistema Experto cuando un conocimiento sea válido durante un largo periodo de tiempo y vaya por el mayor número de personas.

#### CUÁNTO

- ¿Cuánto costará?
- ¿Cuánto aportará?, ¿Se puede cifrar su uso en términos de productividad, de reducción de costos de mantenimiento, de mejora de la condiciones de trabajo y de la calidad de sus productos?, ¿Resultará en una mejora en el servicio para el cliente o el usuario?

### 3.2 ASPECTOS GENERALES

#### El Equipo de Desarrollo

Para desarrollar un SE primero se debe conocer el equipo de gente necesario, después los métodos que utiliza ese equipo de gente y por último cómo prueban y construyen prototipos de software para terminar en el sistema final.

Las personas que componen un grupo o un equipo, como en todos los ámbitos deben cumplir ciertas características y cada uno de ellos dentro del equipo desarrolla un papel distinto. A continuación se detalla cada componente del equipo dentro del desarrollo y cuál es la función de cada uno:

- a) **El Experto.** La función del experto es la de poner sus conocimientos especializados a disposición del Sistema Experto.
- b) **El ingeniero del conocimiento.** Es el ingeniero que plantea las preguntas al experto, estructura sus conocimientos y los implementa en la base de conocimientos.
- c) **El usuario.** El usuario aporta sus deseos y sus ideas, determinando especialmente el escenario en el que debe aplicarse el Sistema Experto.

En el desarrollo de un Sistema Experto, el ingeniero del conocimiento y el Experto trabajan muy unidos. El primer paso consiste en elaborar los problemas que deben ser resueltos por el sistema. Precisamente en la primera fase de un proyecto es de vital importancia determinar correctamente el ámbito delimitado de trabajo. Aquí se incluye ya el usuario posterior, o un representante del grupo de usuarios. Para la aceptación, y en consecuencia para el éxito, es de vital y suma importancia tener en cuenta los deseos y las ideas del usuario.

Una vez delimitado el dominio, se alimenta el sistema con los conocimientos del experto. El experto debe comprobar constantemente que su conocimiento haya sido transmitido de la forma más conveniente. El ingeniero del conocimiento es responsable de una implementación correcta, pero no de la exactitud del conocimiento. La responsabilidad de esta exactitud recae en el experto.

De ser posible, el experto deberá tener comprensión para los problemas que depara el procesamiento de datos. Ello facilitará mucho el trabajo. Además, no debe ignorarse nunca al usuario durante el desarrollo, para que al final se disponga de un sistema que le sea de máxima utilidad.

La estricta separación entre usuario, experto e ingeniero del conocimiento no deberá estar siempre presente. Pueden surgir situaciones en las que el experto puede ser también el usuario. Este es el caso, cuando exista un tema muy complejo cuyas relaciones e interacciones deben ser determinadas una y otra vez con un gran consumo de tiempo. De esta forma el experto puede ahorrarse trabajos repetitivos.

### **3.3 METODOLOGÍA PARA LA CONSTRUCCIÓN DE UN SE**

En el desarrollo de Sistemas Expertos se plantean dos importantes riesgos:

1. No existen implementaciones similares que puedan servir de orientación al encargado del desarrollo en casi la totalidad de los casos.
2. En muchos puntos, los requisitos necesarios están esbozados con muy poca Precisión.

El diseño y la especificación requieren una temprana determinación de la interfaz del software y de la funcionalidad de los componentes. Durante el desarrollo, resulta apropiado empezar con implementaciones tipo test para encontrar el camino hacia una solución definitiva y para hacerlas coincidir con las necesidades del usuario.

Un método efectivo es la implementación de un prototipo de Sistema Experto que permita llevar a cabo las funciones más importantes de éste, aunque con un esfuerzo de desarrollo considerablemente inferior al de la implementación convencional.

La programación orientada a objetos (OOP) es un tipo de programación que provee una manera de modularizar programas estableciendo áreas de memoria distribuidas para datos y procedimientos, que pueden ser usadas como plantillas para crear copias de tales módulos conforme se requieran.

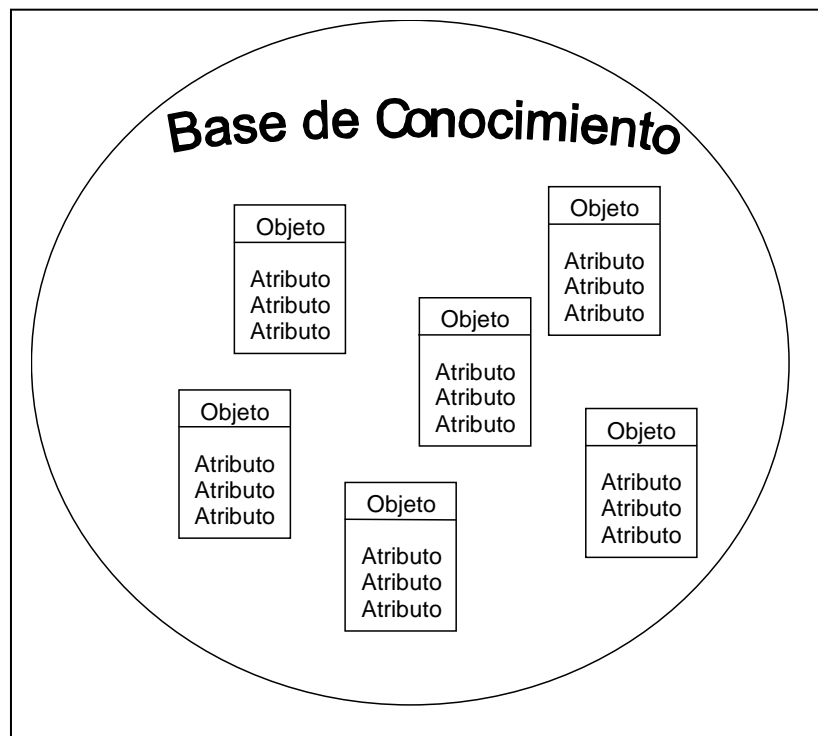
Cada área de memoria a la que se refiere en el párrafo anterior es conocida como *clase*, mientras que las copias creadas a partir de la clase son llamadas *objetos*. La OPP tiene varias propiedades, entre las cuales destacan:

- **Abstracción.** Permite enfocarse en la solución general del problema, sin preocuparse de los detalles.
- **Encapsulación.** Es el concepto de que un objeto debería tener separada su interfaz de su implementación. Es decir, un objeto es visto como una “caja negra”.
- **Herencia.** Significa que se puedan crear clases que “hereden” el comportamiento de una o más clases padre; y que además añadan su propio comportamiento.

El concepto de encapsulación evita que se pueda modificar el estado interno de un objeto por el código cliente que lo utiliza, es decir solamente el código mismo del objeto puede modificar el estado interno del mismo. Esto se puede aprovechar en la construcción de SE creando clases que correspondan a los componentes de un SE.



Por lo tanto si se utiliza la metodología orientada a objetos a la construcción de SE, se debe crear una clase que contenga el código referente a la base de conocimiento. Si dicha base es un conjunto de objetos con sus propiedades, entonces dicha clase será una clase contenedor de objetos de otra clase, los cuales contendrán el nombre del objeto y una lista de atributos. La figura 3.1 muestra lo expuesto.



**Figura 3.1. La base de conocimientos es vista como un objeto contenedor de otros objetos usando la Metodología Orientada a Objetos**

Obviamente, la clase Base de Conocimientos deberá tener todo el código necesario para leer objetos, quizá una base de datos, quizá un archivo, guardar objetos, ingresar nuevos objetos o atributos, eliminar objetos, etc.

De manera similar, el motor de inferencia se implementa como una clase, la cual contiene encapsulada toda la lógica que hace que funcione el SE, es decir, si se implementa usando encadenamiento hacia delante, hacia atrás o reglas, dicha implementación estará contenida dentro de la clase.

Esta metodología tiene diversas ventajas, entre las cuales destaca el hecho que cualquier modificación o mantenimiento que se le quiera realizar a un determinado componente no afectará al otro y viceversa. Tello<sup>14</sup> propone crear especializaciones o jerarquías de clases (usando herencia) que incluyan métodos que permitan a los objetos modificarse a sí mismos, logrando con esto que un SE aprenda por sí mismo, modificando su base de conocimiento sin modificar los demás componentes. Otra ventaja de la OOP es que mediante la herencia, las clases hijas heredarán el comportamiento de las clases padres, evitando con esto, tener que incluir código redundante en las especializaciones.

### **3.4 LENGUAJES UTILIZADOS EN LA PROGRAMACIÓN DE UN SISTEMA EXPERTO**

#### *Lenguaje de programación Visual Basic*

*Visual Basic* es uno de los lenguajes de programación que más entusiasmo despiertan entre los programadores, tanto expertos como novatos. En el caso de los programadores expertos por la facilidad con la que se desarrollan aplicaciones complejas en muy poco tiempo (comparado con lo que cuesta programar en C++, por ejemplo). En el caso de los programadores novatos por el hecho de ver de lo que son capaces a los pocos minutos de empezar su aprendizaje.

Visual Basic es un lenguaje de programación visual y un programa basado en objetos, aunque no orientado a objetos como C++ o *Java*. La diferencia está en que Visual Basic utiliza objetos con propiedades y métodos.

---

<sup>14</sup> TELLO, Ernest. **Object oriented programming for artificial intelligence**

Por otra parte, los programas *orientados a eventos* son los programas típicos de *Windows*, tales como *Word, Excel, Netscape y Power Point*. Cuando uno de estos programas ha arrancado, lo único que hace es quedarse a la espera de la acción del usuario, que en este caso son llamados *eventos*. El usuario dice si quiere abrir y modificar un fichero existente, o bien comenzar a crear un fichero desde el principio. Estos programas pasan la mayor parte de su tiempo esperando las acciones del usuario (eventos) y respondiendo a ellas. La programación orientada a objetos es sensiblemente más complicada que la secuencial y la interactiva, pero *Visual Basic* la hace especialmente sencilla y agradable

*Visual Basic* está orientado a la realización de programas para *Windows*, pudiendo incorporar todos los elementos de este entorno de programación: ventanas, botones, cajas de diálogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús, etc.

Prácticamente todos los elementos de interacción con el usuario de los que dispone *Windows* pueden ser programados en *Visual Basic* de un modo muy sencillo.

## *LISP*

*LISP*, acrónimo de Lenguaje de Procesamiento de Listas, fue inventado por John McCarthy y su equipo en la Universidad de Stanford a finales de 1950. Originalmente fue creado como un modelo computacional de procesos matemáticos.

*LISP* actualmente está diseñado para manejar símbolos matemáticos (variables), por lo que es utilizado perfectamente para la investigación en IA, donde un símbolo puede representar cualquier cosa. *LISP* tiene dos características principales que lo hacen sobresalir entre los demás lenguajes para IA; primero, es altamente flexible, es decir, es posible escribir en programa *LISP* para producir cualquier comportamiento deseable de la computadora; segundo, es indefinidamente extensible, lo que significa que si como programador siente que a *LISP* le falta alguna característica, puede escribir un programa *LISP* que provea dicha característica y hacer que ese programa forme parte de su *LISP* personal.

LISP utiliza un ciclo llamado *leer – evaluar – imprimir*. Cuando el programador teclea algo en LISP, el lenguaje toma lo que se ha tecleado, intenta responder de cualquier forma y después despliega dicha respuesta en la pantalla. El término en LISP usado para “ver lo que se ha tecleado” es *leer*; el término usado para “tratar de responder de cualquier forma” es *evaluar*; y el término usado para “desplegar la respuesta en la pantalla” es *imprimir*. Si hubiera una persona en vez de una máquina en frente del programador, diríamos que dicha persona estaría escuchando, comprendiendo y respondiendo al programador.

### *CLIPS*

CLIPS es otra herramienta para el desarrollo de SE que ofrece un entorno completo para su construcción basado en reglas y objetos. CLIPS está siendo utilizado por numerosos usuarios como la NASA (que es su creadora), muchas 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. Así, al darle una portabilidad con programas en lenguaje C.

### *PROLOG*

Prolog es un lenguaje de programación que se centra alrededor de un conjunto pequeño de mecanismos, incluyendo reconocimiento de patrones, estructuras de datos basadas en árboles y backtraking (retroceso) automático. Este conjunto pequeño constituye una estructura de programación sorprendentemente poderosa y flexible. Prolog es ideal para resolver problemas que involucren objetos – en particular objetos estructurados – y relaciones entre ellos. Por ejemplo, un ejercicio muy sencillo en Prolog es expresar relaciones espaciales, de la forma: “la esfera azul detrás de la verde”.

Prolog nació en Europa, y fue implementado primeramente para dar soporte al Procesamiento del Lenguaje Natural (PLN). Prolog trabaja esencialmente en la lógica matemática. Además tiene cierto vigor híbrido en el sentido de que contiene características declarativas de la lógica computacional matemática. En vez de escribir

un procedimiento con una secuencia de pasos, un programador Prolog escribe un conjunto declarativo de reglas y hechos con sus respectivas relaciones. Debido a esto, los diagramas de flujo y las técnicas convencionales de programación no aplican en Prolog.

Walker<sup>15</sup> afirma que niños de 10 años de edad encuentran a Prolog como un lenguaje sencillo de aprender. Pero los programadores profesionales con muchos años de experiencia a menudo lo encuentran revuelto y confuso. Lo revuelto de Prolog es la simplicidad y que no tiene las características de los lenguajes comunes.

### *C y C++*

C es uno de los lenguajes de programación más populares en uso. Proporciona un esqueleto estructurado sin límites para la creatividad del programador, una de las ventajas de C sobre todos los lenguajes usados para la investigación en IA es que es un lenguaje estructurado y además, si su aplicación no requiere usar la técnica Backtracking ni los recursos de una base de datos, estos no se convierten en un obstáculo extra que debe soportar la aplicación.

No hay una sola técnica de IA que no pueda ser desarrollada en un lenguaje procedimental como lo es C. De hecho, el desarrollo de ciertas rutinas es en verdad más claro en C que un lenguaje de IA.

En cuanto a C++ se puede decir que es una extensión orientada a objetos de C, la cual permite utilizar la metodología orientada a objetos para la creación de diversos sistemas y programas, incluso los de IA.

---

<sup>15</sup> WALKER Adrian, Michael Mc Cord, John Sowa y Walter Wilson. **Knowledge – Based Systems and Prolog**

## IV. TIPOS DE SISTEMAS EXPERTOS

### 4.1 SISTEMAS EXPERTOS DETERMINÍSTICOS

#### 4.1.1 Introducción

En nuestra vida diaria encontramos muchas situaciones complejas gobernadas por reglas deterministas: sistemas de control de tráfico, sistemas de seguridad, transacciones bancarias, etc. Los sistemas basados en reglas son una herramienta eficiente para tratar estos problemas. Las reglas deterministas constituyen la más sencilla de las metodologías utilizadas en sistemas expertos. La base de conocimiento contiene el conjunto de reglas que definen el problema y el motor de inferencia saca las conclusiones aplicando la lógica clásica a estas reglas.

#### 4.1.2 La Base de Conocimiento

En los sistemas basados en reglas intervienen dos elementos importantes: la base de conocimiento y los datos. Los datos están formados por la evidencia o los hechos conocidos en una situación particular. Este elemento es dinámico, es decir, puede cambiar de una aplicación a otra.

En situaciones deterministas, las relaciones entre un conjunto de objetos pueden ser representadas mediante un conjunto de reglas. El conocimiento se almacena en la base de conocimiento y consiste en un conjunto de objetos y un conjunto de reglas que gobiernan las relaciones entre esos objetos. La información almacenada en la base de conocimiento es de naturaleza permanente y estática, es decir, no cambia de una aplicación a otra, a menos que se incorpore al sistema experto elementos de aprendizaje.

A continuación se dan unos ejemplos de reglas:

Regla 1: Si  $\text{nota} > 9$ , entonces  $\text{calificación} = \text{sobresaliente}$ .

Regla 2: Si  $\text{puesto} < 20$  o  $\text{nota} > 7$ , entonces  $\text{admitir} = \text{sí}$  y  $\text{Notificar} = \text{sí}$ .

Cada una de las reglas anteriores relaciona dos o más objetos y está formada por las partes siguientes:

- La *premisa* de la regla, que es la expresión lógica entre las palabras claves *si* y *entonces*. La premisa puede contener una o más afirmaciones objeto-valor conectadas con operadores lógicos *y*, *o*, o *no*. Por ejemplo, la premisa de la Regla 1 consta de una única afirmación objeto-valor, mientras que las premisas de la Regla 2 constan de dos afirmaciones objeto-valor conectadas por un operador lógico.
- La *conclusión* de la regla, que es la expresión lógica tras la palabra *entonces*.

Los ejemplos anteriores facilitan la definición siguiente de regla.

*Una regla es una afirmación lógica que relaciona dos o más objetos e incluye dos partes, la premisa y la conclusión. Cada una de estas partes consiste en una expresión lógica con una o más afirmaciones objeto-valor conectadas mediante operadores lógicos y, o, o no.*

#### 4.1.3 Motor de Inferencia

Como ya se mencionó existen dos tipos de elementos: los datos (hechos o evidencia) y el conocimiento (el conjunto de reglas almacenado en la base de conocimiento). El motor de inferencia usa ambos para obtener nuevas conclusiones o hechos. Por ejemplo, si la premisa de una regla es cierta, entonces la conclusión de la regla también debe ser cierta. Los datos iniciales se incrementan incorporando las nuevas conclusiones. Por ello, tanto los hechos iniciales o datos de partida como las conclusiones derivadas de ellos forman parte de los hechos o datos de que se dispone en un instante dado.

Para obtener conclusiones se utilizan diferentes tipos de reglas y estrategias de inferencia y control.

Las estrategias de inferencia son:

- Encadenamiento de reglas.
- Encadenamiento de reglas orientado a un objeto.
- Compilación de reglas.

que son utilizadas por el motor de inferencia para obtener conclusiones

Tome en cuenta que ninguna de las estrategias anteriores, si se implementan solas, conduce a todas las conclusiones posibles. Por ello, deben implementarse varias reglas y estrategias en el sistemas experto para que el motor de inferencia sea capaz de obtener tantas conclusiones como sea posible.

## **4.2 SISTEMAS EXPERTOS QUE MANEJAN INCERTIDUMBRE**

### *4.2.1 Introducción*

La solución de muchos problemas que requieren de un comportamiento humano inteligente, tiene asociado algún grado de incertidumbre. Por ello, es natural esperar que los sistemas expertos también sean capaces de manejar incertidumbre. En este apartado describiremos los métodos más populares para tratar con incertidumbre: probabilidades bayesianas, factores de certeza y lógica difusa.

### *4.2.2 Factores de Certeza*

Durante el desarrollo de MYCIN, los investigadores descubrieron que el modelo bayesiano era inadecuado para manejar incertidumbre, porque en el área médica es difícil contar con la gran cantidad de datos que el Teorema de Bayes requiere. Además, los estudios indican que los médicos parecen razonar capturando evidencias que confirmen o nieguen sus hipótesis y éstas evidencias las consiguen independientemente, en contraposición con el proceso usado por la técnica bayesiana que usa probabilidades condicionadas. Por otro lado, existe la necesidad de representar el conocimiento médico



explícitamente (lo que no puede hacerse usando probabilidades), para que el sistema pueda explicar su razonamiento.

Estas razones hicieron que se desarrollara el formalismo de los factores de certeza (FC), el cual se basa vagamente en el análisis bayesiano y varía ligeramente de implementación a implementación. Describiremos esta técnica tal cual se implementó en MYCIN.

Los sistemas expertos implementan el formalismo de FC por medio de reglas:

SI evidencia

ENTONCES hipótesis (FC)

donde evidencia es uno o más hechos conocidos para soportar la derivación de la hipótesis. El valor de FC denota la creencia en la hipótesis, dada que la evidencia es observada. El formalismo FC usa tres valores distintos:

- 1) La medida de credibilidad MC (h,e), la cual es un valor entre 0 y 1 que representa el grado de creencia en que la hipótesis h es soportada por la evidencia e.
- 2) La medida de incredibilidad MI (h,e), la cual es un valor entre 0 y 1 que representa el grado de no creencia de que la hipótesis h es soportada por la evidencia e.
- 3) El factor de certeza FC, el cual es una combinación de MC y MI, su rango de variación fluctúa entre -1 (negación de h) y +1 (confirmación de h). Se define así:

$$FC = \frac{MC - MI}{1 - \min(MC - MI)} \quad (4.1)$$

Durante el proceso de inferencia, muchas reglas pueden derivar la misma hipótesis o conclusión. Como resultado debe existir un mecanismo para combinar los FC de cada regla y obtener un solo FC para esa hipótesis. Se usan las siguientes ecuaciones:

$$\begin{aligned}
 FC_{rev}(FC_{old}, FC_{new}) &= FC_{old} + FC_{new} (1 - FC_{old}) \quad \text{si } FC_{old} \text{ y } FC_{new} > 0 \\
 &= -FC_{rev}(-FC_{old}, -FC_{new}) \quad \text{si } FC_{old} \text{ y } FC_{new} < 0 \quad (4.2) \\
 &= (FC_{old} + FC_{new}) / (1 - \min(|FC_{old}|, |FC_{new}|)) \\
 &\quad \text{si uno de } FC_{old} \text{ y } FC_{new} < 0
 \end{aligned}$$

Los factores de certeza son la técnica más popular, debido a que su modelo computacional es simple y porque permite que el conocimiento representado mediante reglas pueda cuantificarse con incertidumbre. La obtención de FC es significativamente más fácil que la obtención de valores de otros métodos. No se requiere una base estadística, solamente se pregunta al experto para conseguir los valores.

Muchos sistemas, incluyendo MYCIN, han utilizado este formalismo y han tenido un alto grado de competencia en sus áreas de aplicación. Pero los detractores de este método consideran que estos éxitos se deben principalmente al conocimiento representado en las reglas más que a los valores de los FC asociados al conocimiento. Otras críticas a este modelo incluyen su débil fundamento teórico.

#### 4.2.3 *Lógica Difusa*

A mediados de los 60, Zadeh desarrolló la teoría de los “conjuntos difusos” y posteriormente la “lógica difusa” para tratar con numerosos conceptos usados en el razonamiento humano, los cuales son intrínsecamente vagos e imprecisos, tales como los adjetivos (ej.: “alto”, “viejo”), los cuantificadores del lenguaje natural (ej.: “muchos”) y las sentencias (ej.: “no es muy probable”). Puesto que muchos expertos expresan sus conocimientos de manera subjetiva e imprecisa, la lógica difusa parece aplicarse naturalmente a su formalización.

Mientras la lógica clásica define la pertenencia de los distintos elementos a un conjunto haciéndoles corresponder los valores 1 ó 0 según pertenezcan o no a un conjunto, Zadeh creó una función de posibilidad que les asocia valores en el intervalo [0,1], los cuales miden el grado de pertenencia a dicho conjunto. Los valores 1 y 0 aseguran la pertenencia o no al conjunto, mientras que los valores intermedios se asocian a elementos de pertenencia dudosa. Por ejemplo, si nuestro predicado consiste en determinar la membresía de un individuo en cierta categoría étnica (ej.: mexicana, alemana, libanesa), tenemos que tomar en cuenta que esta categorización no es precisa, pues muchos individuos tienen un origen mestizo. Por lo tanto, nuestro predicado tendrá un valor numérico en el rango [0,1] para expresar el grado de mestizaje étnico de cada persona.

La lógica difusa ha sido aplicada en numerosos sistemas expertos incluyendo el diagnóstico médico, recuperación de información y detección de fallas, pero existe mucho debate acerca de su viabilidad debido a sus inherentes dificultades. El desarrollo de funciones de membresía no es trivial y a menudo toma más tiempo que el dedicado al desarrollo del sistema. A pesar de ello, esta técnica continúa generando interés y siendo aplicada en numerosos sistemas.

#### 4.2.4 Probabilidad bayesiana

Es la técnica más antigua y está basada en la teoría clásica de probabilidades. El teorema de Bayes proporciona un método para computar la probabilidad de que ocurra un evento ( $H_i$ , hipótesis) si antes se ha producido otro ( $E_j$ , evidencia) relacionado con el primero. Está definido así:

$$p(H_i/E_j) = \frac{p(E_j/H_i) * p(H_i)}{p(E_j)} \quad (4.3)$$

Por ejemplo, si queremos saber la probabilidad de que una persona tenga resfrío ( $H$ =resfrío) si sabemos que estornuda ( $E$ =estornudos), podemos aplicar la fórmula (4.3), junto con los datos probabilísticos de la población de estudio: “probabilidad de que una

persona estornude dado que está resfriada”, “probabilidad de que una persona tenga resfrío” y “probabilidad de que una persona estornude”.

La propagación de la incertidumbre se obtiene generalizando la fórmula (4.3), para tratar con ‘m’ hipótesis y ‘n’ piezas de evidencia, así:

$$p(H_i/E_1E_2\dots E_n) = \frac{p(E_1E_2\dots E_n/H_i) * p(H_i)}{p(E_1E_2\dots E_n)} \quad (4.4)$$

Esta probabilidad denota la probabilidad de que ocurra la hipótesis  $H_i$  dadas las evidencias  $E_1, E_2 \dots E_n$  (ej.: probabilidad de que un paciente tenga la enfermedad  $H_j$ , dados los síntomas  $E_i$ ). Si bien este método tiene el mejor fundamento teórico para el manejo de la incertidumbre, tiene la gran desventaja de necesitar una significativa cantidad de datos probabilísticos para construir la base de conocimientos (ej.: un sistema de diagnóstico de 50 enfermedades ( $p$ ) y 300 síntomas ( $q$ ) necesitaría  $15.050$  ( $p*q + p$ ) valores de probabilidad, asumiendo que las respuestas son excluyentes entre sí y que los síntomas son independientes. Si no lo son, se necesitan muchos más valores). Además, los sistemas basados en probabilidad no pueden explicar cómo llegaron a su conclusión (su base de conocimientos está formada por números), por lo que no dan mucha confianza a los expertos.

Antes de continuar con los Sistemas Expertos probabilísticos, es importante conocer algunos conceptos básicos de Grafos; para esto se presenta un breve repaso sobre este tema en la sección 4.3. Los lectores que estén familiarizados con estos conceptos pueden omitir esta sección e ir directamente a la sección 4.4.

## 4.3 CONCEPTOS SOBRE GRAFOS

### 4.3.1 Introducción

Los grafos son herramientas muy útiles para definir sistemas expertos y otros modelos utilizados en el área de la inteligencia artificial. Muchos de los resultados teóricos de la teoría de grafos pueden ser utilizados para analizar diversos aspectos de estos campos.

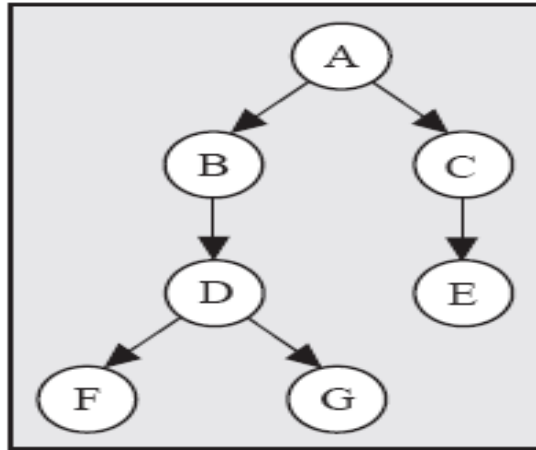
### 4.3.2 Conceptos Básicos y Definiciones

Supóngase un conjunto de objetos  $X = \{X_1, X_2, \dots, X_n\}$  que pueden relacionarse entre sí. El conjunto  $X$  puede ser representado gráficamente por una colección de *nodos* o *vértices*, asociando un nodo a cada elemento de  $X$ . Estos nodos pueden conectarse por *aristas*, indicando las relaciones existentes entre los mismos. Una arista entre los nodos  $X_i$  y  $X_j$  se denotaría mediante  $L_{ij}$ . Así mismo, el conjunto de todas las aristas se denotaría por  $L = \{L_{ij} \mid X_i \text{ y } X_j \text{ están conectados}\}$ . Por tanto, un grafo puede definirse de forma intuitiva mediante el conjunto de nodos,  $X$ , y las relaciones entre los mismos,  $L$ . En el siguiente ejemplo se ilustra esta idea intuitiva. A continuación se introduce una definición formal.

La Figura 4.1 es un ejemplo de un grafo compuesto de seis nodos  $X = \{A, B, \dots, G\}$  y de un conjunto de seis aristas,

$$L = \{LAB, LAC, LBD, LCE, LDF, LDG\}.$$

Los nodos están representados por círculos y las aristas por líneas que unen los nodos correspondientes.



**Figura 4.1. Ejemplo de Grafo o Red.**

*Un grafo es un par de conjuntos  $G = (X,L)$ , donde  $X = \{X_1, X_2, \dots, X_n\}$  es un conjunto finito de elementos (nodos), y  $L$  es un conjunto de aristas, es decir, un subconjunto de pares ordenados de elementos distintos de  $X$ . Los términos grafo y red se emplearan como sinónimos en este libro.*

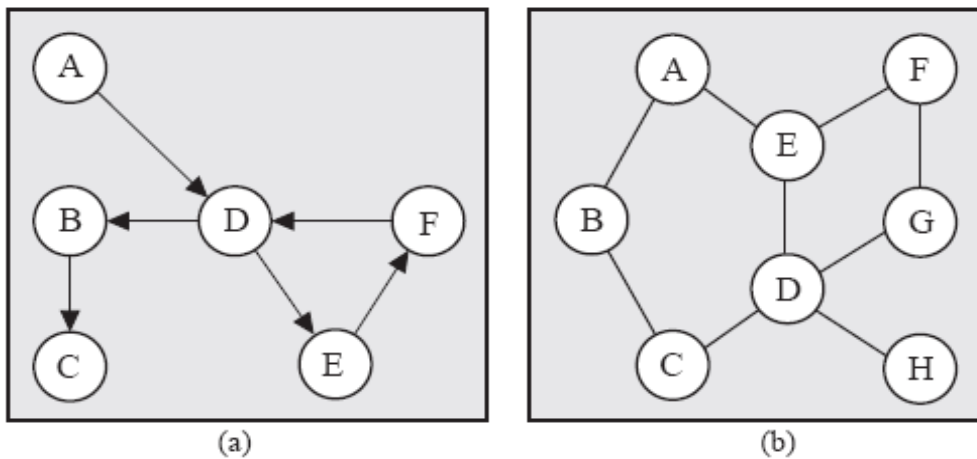
El concepto de grafo puede definirse de forma más general. Por ejemplo, puede permitirse que dos nodos estén conectados por más de una arista, o incluso que un nodo esté conectado consigo mismo. Sin embargo, en el campo de los sistemas expertos, los grafos se utilizan para representar un conjunto de variables (nodos), y unas relaciones de dependencia entre ellas (aristas). Por tanto, no es necesario que dos nodos estén unidos por más de una arista, o que una arista una un nodo consigo mismo.

Las aristas de un grafo pueden ser *dirigidas* o *no dirigidas*, dependiendo de si se considera o no, el orden de los nodos. En la práctica, esta distinción dependerá de la importancia del orden en que se relacionen los objetos.

#### 4.3.3 Grafos dirigidos y No dirigidos

*Un grafo en el cual todas las aristas son dirigidas se denomina grafo dirigido, y un grafo en el que todas sus aristas son no dirigidas se denomina no dirigido.*

Por tanto, en un grafo dirigido es importante el orden del par de nodos que definen cada arista, mientras que en un grafo no dirigido, el orden carece de importancia.



**Figura 4.2. Ejemplos de un grafo dirigido (a) y uno No dirigido (b).**

En las Figuras 4.2(a) y 4.2(b) se muestran ejemplos de un grafo dirigido y de un grafo no dirigido, respectivamente. El grafo de la Figura 4.2(a) está definido por:

$$X = \{A, B, C, D, E, F\},$$

$$L = \{A \rightarrow D, B \rightarrow C, D \rightarrow B, F \rightarrow D, D \rightarrow E, E \rightarrow F\},$$

mientras que para el grafo de la Figura 4.2(b) se tiene

$$X = \{A, B, C, D, E, F, G, H\},$$

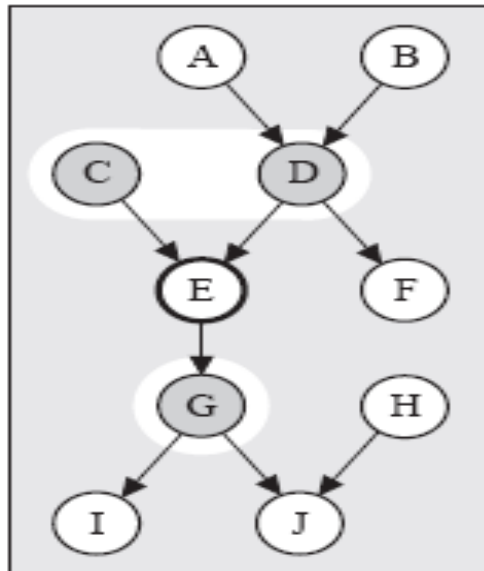
$$L = \{A - B, B - C, C - D, D - E, E - A, E - F, F - G, G - D, D - H\}.$$

#### 4.3.4 Características de Grafos Dirigidos

##### **Padre e hijo.**

*Cuando existe una arista dirigida,  $X_i \rightarrow X_j$ , del nodo  $X_i$  al nodo  $X_j$ , entonces se dice que el nodo  $X_i$  es un padre del nodo  $X_j$ , y que el nodo  $X_j$  es un hijo de  $X_i$ .*

El conjunto de los padres de un nodo  $X_i$  se denota mediante  $\Pi x_i$  o simplemente  $\Pi i$ . Por ejemplo, los nodos  $C$  y  $D$  son los padres del nodo  $E$  en el grafo de la Figura 4.3. En un grafo dirigido, el conjunto de hijos de un nodo coincide con el conjunto de nodos adyacentes.

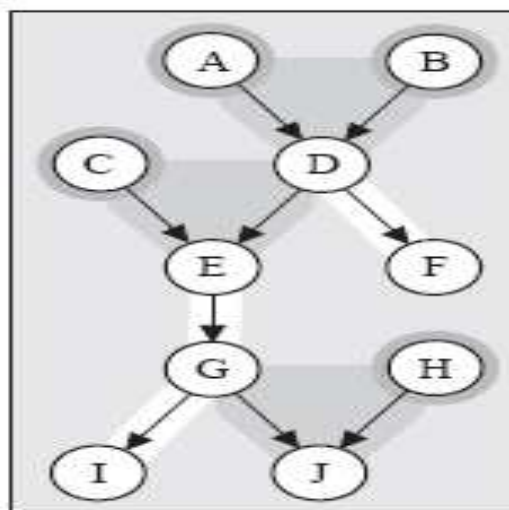


**Figura 4.3. Padres e hijos del nodo E.**

**Familia de un nodo.**

*El conjunto formado por un nodo y sus padres se denomina la familia del nodo.*

Por ejemplo, las distintas zonas sombreadas en el grafo de la Figura 4.4 muestran las distintas familias asociadas a este grafo. En este ejemplo se pueden observar familias con uno, dos y tres nodos. Las familias de un grafo tienen un papel muy importante en, pues la estructura de dependencias codificada en un grafo dirigido podría trasladarse a una función de probabilidad definiendo distribuciones de probabilidad locales sobre cada familia del grafo.



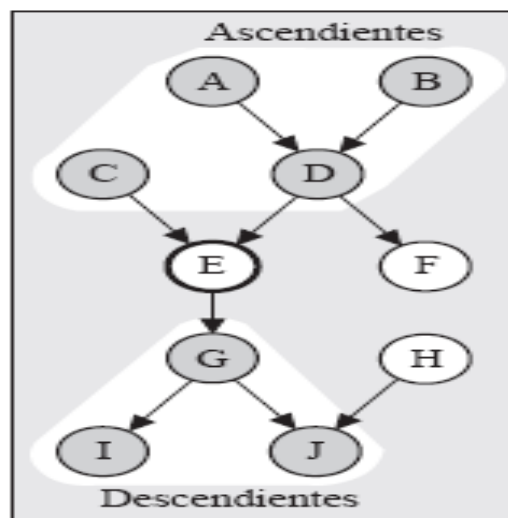
**Figura 4.4. Familias asociadas a los nodos de un grafo.**



**Ascendentes de un nodo.** Un nodo  $X_j$  se denomina ascendente del nodo  $X_i$  si existe un camino de  $X_j$  a  $X_i$ .

**Conjunto ancestral.** Un conjunto de nodos  $S$  se denomina un conjunto ancestral si contiene los ascendentes de todos sus nodos.

**Descendientes de un nodo.** Un nodo  $X_j$  se denomina descendiente del nodo  $X_i$  si existe un camino de  $X_i$  a  $X_j$ . La Figura 4.5 muestra los conjuntos de ascendentes y descendientes del nodo  $E$ .

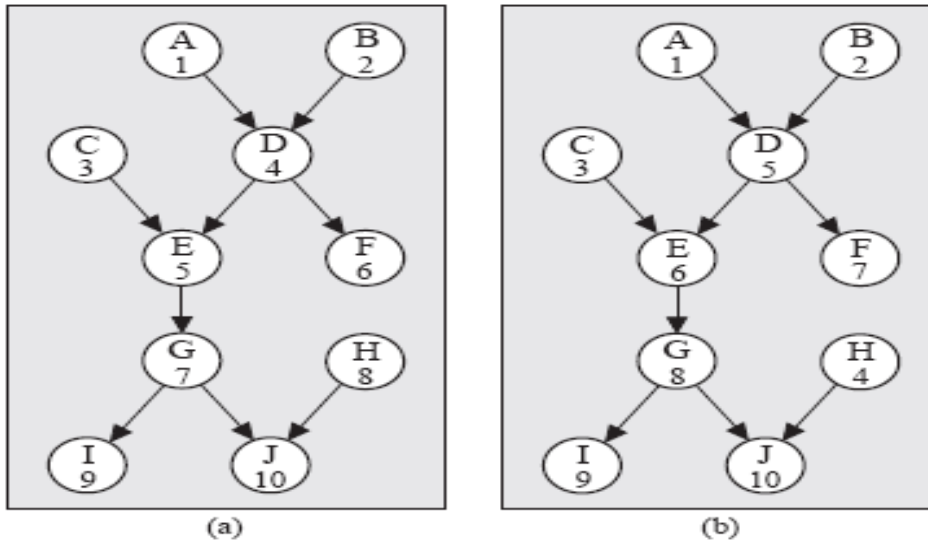


**Figura 4.5.** Ascendentes y descendientes del nodo  $E$ .

### **Numeración ancestral.**

Una numeración de los nodos de un grafo dirigido se denomina ancestral si el número correspondiente a cada nodo es menor que los correspondientes a sus hijos.

Por ejemplo, las dos numeraciones mostradas en las Figuras 4.6 son dos numeraciones ancestrales distintas del mismo grafo. Por tanto, este tipo de numeración no es necesariamente única. Por otra parte, existen grafos dirigidos que no admiten ninguna numeración ancestral.



**Figura 4.6. Dos numeraciones ancestrales del mismo grafo.**

### Árboles y grafos múltiplemente conexos.

*Un grafo dirigido conexo se denomina árbol si el grafo no dirigido asociado es un árbol; en caso contrario se denomina múltiplemente conexo.*

### Grafos cíclicos y acíclicos.

*Un grafo dirigido se denomina cíclico si contiene al menos un ciclo; en caso contrario se denomina grafo dirigido acíclico.*

### Grafos simples y poliárboles.

*Un árbol dirigido se denomina un árbol simple si cada nodo tiene como máximo un padre; en caso contrario se denomina un poliárbol.*

La Figura 4.7 muestra un ejemplo de un árbol simple y un ejemplo de un poliárbol. La Figura 4.8 muestra un grafo cíclico y uno múltiplemente conexo.

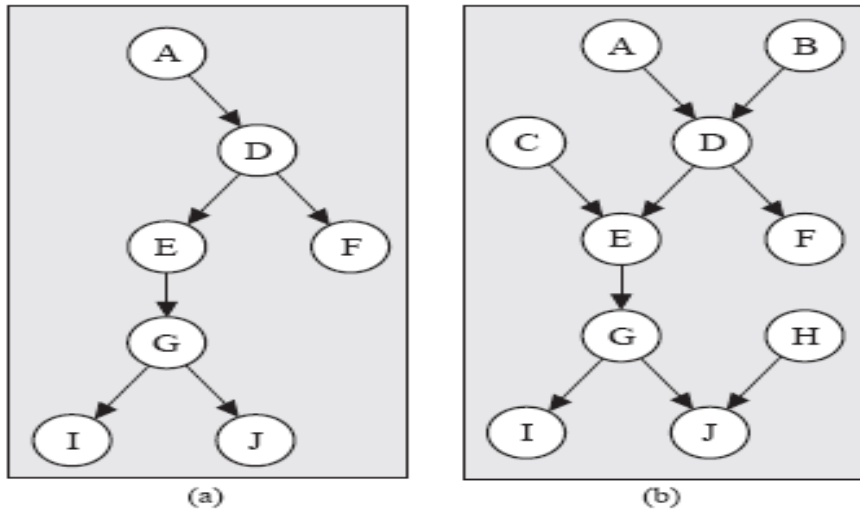


Figura 4.7. Ejemplos de grafos dirigidos: árbol simple (a) y poliárbol (b).

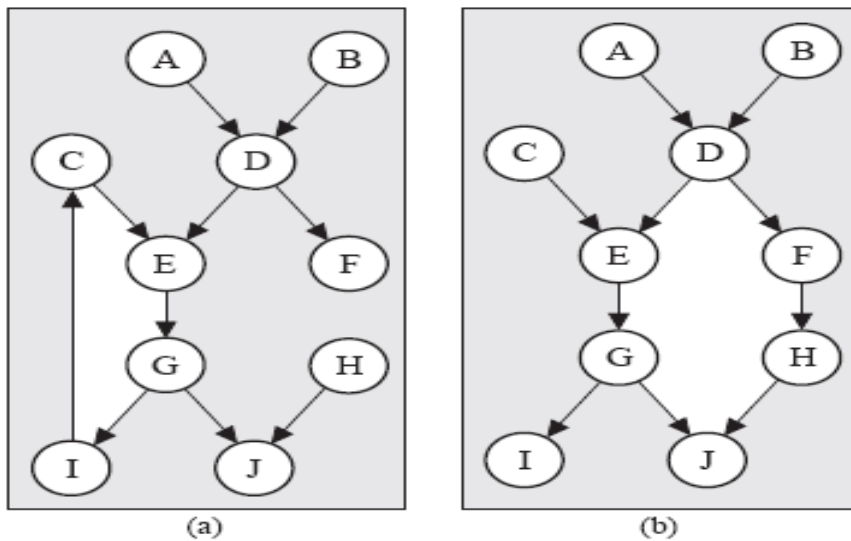


Figura 4.8. Ejemplos de grafos dirigidos: grafo cíclico (a) y múltiplemente conexo (b).

## 4.4 SISTEMAS EXPERTOS BASADOS EN PROBABILIDAD

### 4.4.1 Introducción

En los primeros sistemas expertos, se eligió la probabilidad como medida para tratar la incertidumbre. Pero, desgraciadamente, muy pronto se encontraron algunos problemas, debidos al uso incorrecto de algunas hipótesis de independencia, utilizadas

para reducir la complejidad de los cálculos. Como resultado, en las primeras etapas de los sistemas expertos, la probabilidad fue considerada como una medida de incertidumbre poco práctica. La mayoría de las críticas a los métodos probabilísticos se basaban en el altísimo número de parámetros necesarios, la imposibilidad de una asignación o estimación precisa de los mismos, o las hipótesis poco realistas de independencia. Consecuentemente, en la literatura de la época, surgieron medidas alternativas a la probabilidad, como los factores de certeza, las credibilidades, las plausibilidades, las necesidades o las posibilidades, para tratar la incertidumbre. Sin embargo, con la aparición de las redes probabilísticas, la probabilidad ha resurgido de forma espectacular, y es, hoy en día, la más intuitiva y la más aceptada de las medidas de incertidumbre. Lindley (1987), por ejemplo, dice:

*“La única descripción satisfactoria de la incertidumbre es la probabilidad. Esto quiere decir que toda afirmación incierta debe estar en forma de una probabilidad, que varias incertidumbres deben ser combinadas usando las reglas de la probabilidad, y que el cálculo de probabilidades es adecuado para manejar situaciones que implican incertidumbre. En particular, las descripciones alternativas de la incertidumbre son innecesarias.”*

En los sistemas expertos probabilísticos las relaciones entre las variables se describen mediante su función de probabilidad conjunta. Una vez definido el sistema de grafos que describe el problema, este se puede resolver ya sea por método exacto o métodos de simulación estocástica.

#### ***4.5 Sistemas Expertos Basados en Reglas vs. SE basados en Probabilidad***

Se concluye este capítulo con una breve comparación de los sistemas expertos basados en reglas con los sistemas expertos basados en probabilidad. Se discuten sus analogías y diferencias, y sus ventajas y desventajas.

*Motor de Inferencia:*

En los sistemas expertos basados en reglas las conclusiones se obtienen de los hechos aplicando las diferentes estrategias de inferencia, tales como Modus Ponens, Modus Tollens y encadenamiento de reglas.

Por ello, el motor de inferencia es rápido y fácil de implementar. En los sistemas expertos basados en probabilidad, el motor de inferencia es más complicado que en el caso de los sistemas expertos basados en reglas. El motor de inferencia de un sistema experto probabilístico se basa en la evaluación de las probabilidades condicionales utilizando uno o varios métodos propuestos por los diferentes tipos de sistemas expertos probabilísticos. El grado de dificultad depende del modelo seleccionado y varía desde baja, para los modelos de independencia, a alta, para los modelos de dependencia generales.

#### *Subsistema de Explicación:*

La explicación es fácil en el caso de los sistemas expertos basados en reglas, ya que se sabe que reglas han sido utilizadas para concluir en cada momento. El motor de inferencia sabe que reglas se han utilizado en el encadenamiento y han contribuido a obtener conclusiones y que reglas se han utilizado sin éxito. En el caso de los sistemas expertos basados en probabilidad, la información sobre que variables influyen en otras está codificada en la función de probabilidad conjunta. Por ello, la explicación se basa en los valores relativos de las probabilidades condicionales que miden los grados de dependencia. Una comparación de las probabilidades condicionales para diferentes conjuntos de evidencia permite analizar sus efectos en las conclusiones.

#### *Subsistema de Aprendizaje:*

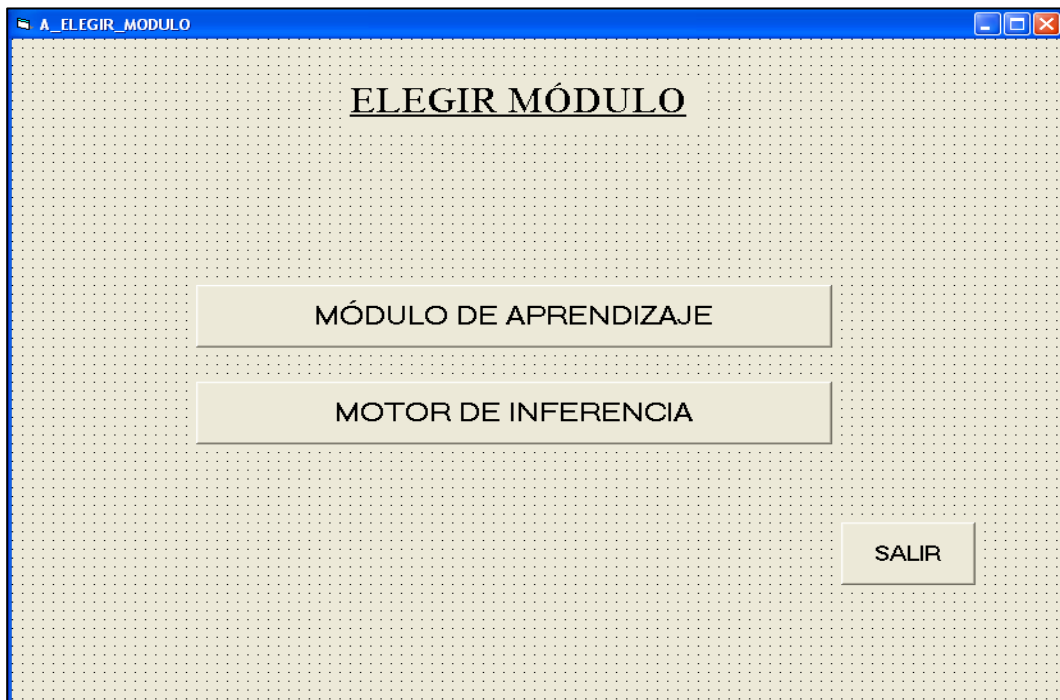
En los sistemas expertos basados en reglas, el aprendizaje consiste en incorporar nuevos objetos, nuevos conjuntos de valores factibles para los objetos, nuevas reglas o modificaciones de los objetos existentes, de los conjuntos de valores posibles, o de las reglas. En los sistemas expertos probabilísticos, el aprendizaje consiste en incorporar o modificar la estructura del espacio de probabilidad: variables, conjunto de posibles valores, o los parámetros (valores de las probabilidades).

# V. DESCRIPCIÓN DEL SISTEMA EXPERTO DESARROLLADO

## 5.1 ESTRUCTURA DEL SISTEMA EXPERTO

### 5.1.1 Componentes

La programación del Sistema Experto se llevo a cabo utilizando el lenguaje de Visual Basic versión 6.0. La estructura del programa estará formada por dos componentes principales: el Módulo de Aprendizaje y el Motor de Inferencia que aparecen en la ventana inicial de la Figura 5.1.



**Figura 5.1. Pantalla Inicial del Sistema Experto.**

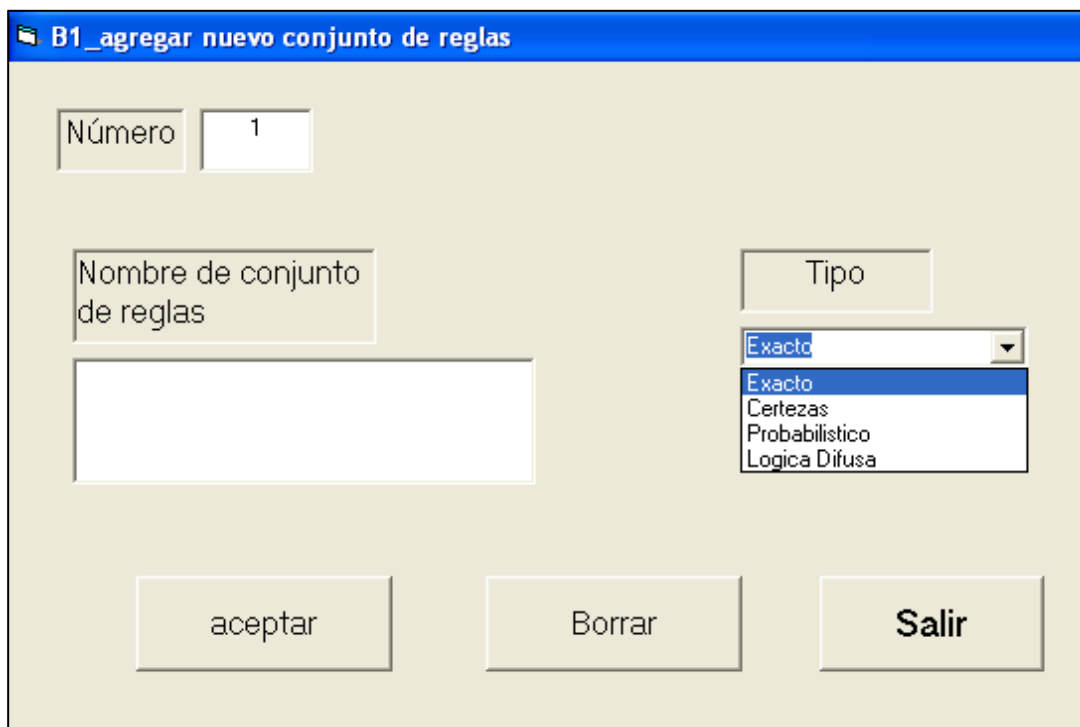
Esta ventana se identifica con el nombre de “ELEGIR MÓDULO”, en el cual se tienen dos botones principales : “MÓDULO DE APRENDIZAJE” y “MOTOR DE INFERENCIA”; el usuario deberá entrar primero a “MÓDULO DE APRENDIZAJE” en el cual tendrá que elegir el conjunto de reglas con el que desea trabajar; si el usuario oprime el botón “MOTOR DE INFERENCIA” el sistema no dejará que prosiga con la

ejecución del programa, ya que la indicará mediante una ventana emergente que no ha elegido ningún conjunto de reglas. El botón “SALIR” detiene totalmente la ejecución del programa.

Dentro del Módulo de Aprendizaje primeramente se elegirá o creará un Conjunto de Reglas

### 5.1.2 Conjunto de Reglas

Los tipos de conjuntos de reglas que puede resolver el Sistema Experto son: Determinísticos, por Método de Certeza, Probabilísticos y por Lógica Difusa (Ver Figura 5.2) , a continuación se explica brevemente en qué consiste cada uno de ellos.



**Figura 5.2. Tipos de conjuntos de Reglas que puede manejar el Sistema Experto.**

El método de programación para la solución para cada uno de estos conjuntos, son los siguientes:

Para el Método Determinístico (Método Exacto). Se utiliza la misma forma de razonamiento como los Sistemas Basados en Reglas (Ver sección 4.1.3).

El método de razonamiento y solución para los conjuntos de reglas por el método de Certezas se mencionó en la sección 4.1.3.

Para el método probabilístico se programó el algoritmo de propagación de evidencia para poliárboles (Ver Apéndice C).

Y por último para los conjuntos de reglas que utilizan lógica difusa, el método de razonamiento se mencionó en la sección 4.2.3.

### 5.1.3 Descripción de tipos de Variables

Las variables que manejan los conjuntos de reglas de tipo determinístico, probabilístico y certezas son de tipo *discretas*, es decir, pueden tomar por ejemplo valores de: *fallo o no fallo, alto, medio y/o bajo*, etc.

En el caso de las variables que utiliza el método de lógica difusa son *continuas* es decir, están dentro de un rango de valores y el programa las puede clasificar dependiendo de estos. Por ejemplo: si la variable presión se encuentra en un rango permitido de 5 a 100 psi, el programa deberá de conocer cuándo la presión está en el rango de baja, media, alta, máxima y no permitida, de acuerdo a los valores numéricos proporcionados.

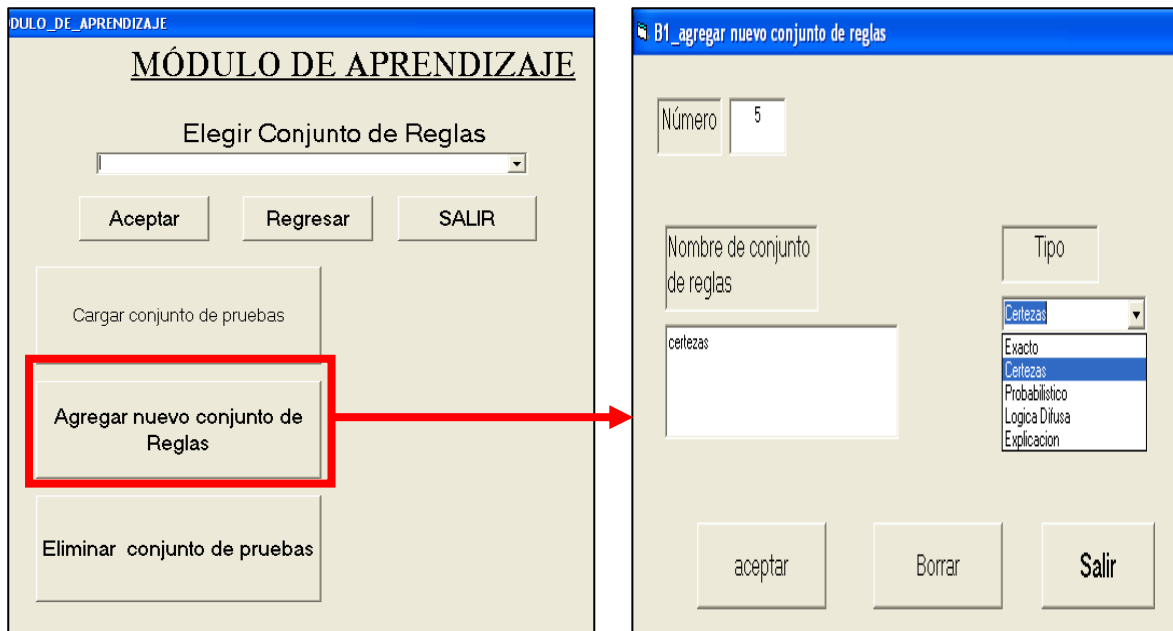
## 5.2 DESCRIPCIÓN DEL SUBSISTEMA DE APRENDIZAJE

### 5.2.1 Creación y modificación de un conjunto de Reglas

El sistema experto puede crear y modificar un conjunto de Reglas. En el caso de que se desea crear un nuevo conjunto, estando dentro del Módulo de Aprendizaje se encuentra la opción “*agregar un nuevo conjunto de reglas*” que inmediatamente después de darle esta instrucción aparece una ventana (Ver Figura 5.3) en la cuál se

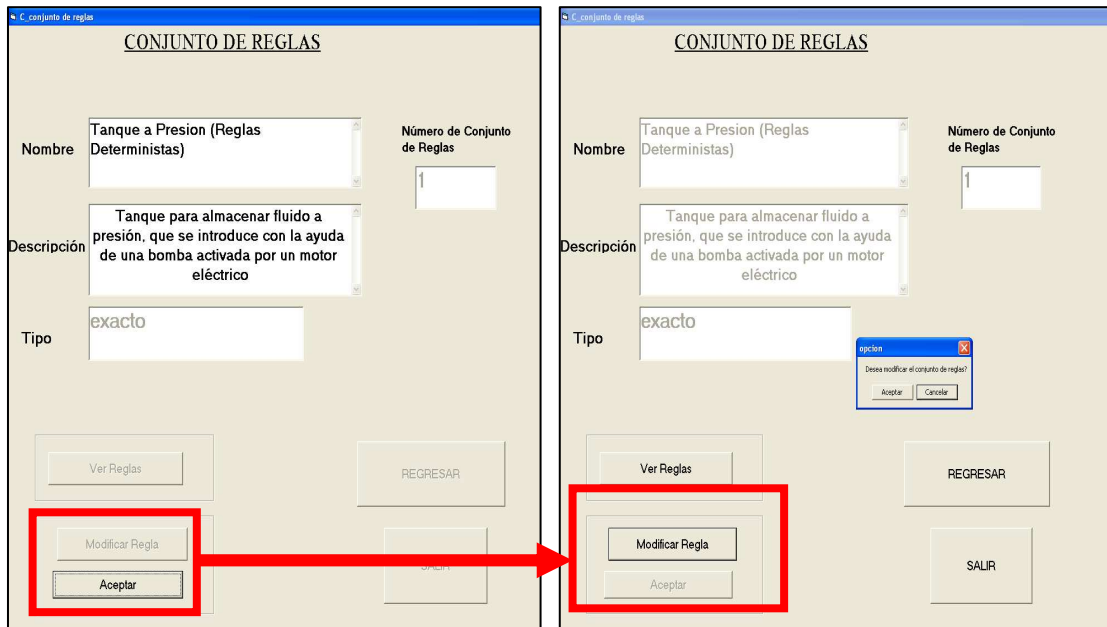


muestra el número de conjunto, se tendrá que dar un nombre a dicho conjunto y escoger el tipo de acuerdo a la solución que se requiera.



**Figura 5.3. Ventanas que muestran el proceso para agregar un nuevo conjunto de Reglas dentro del Sistema Experto.**

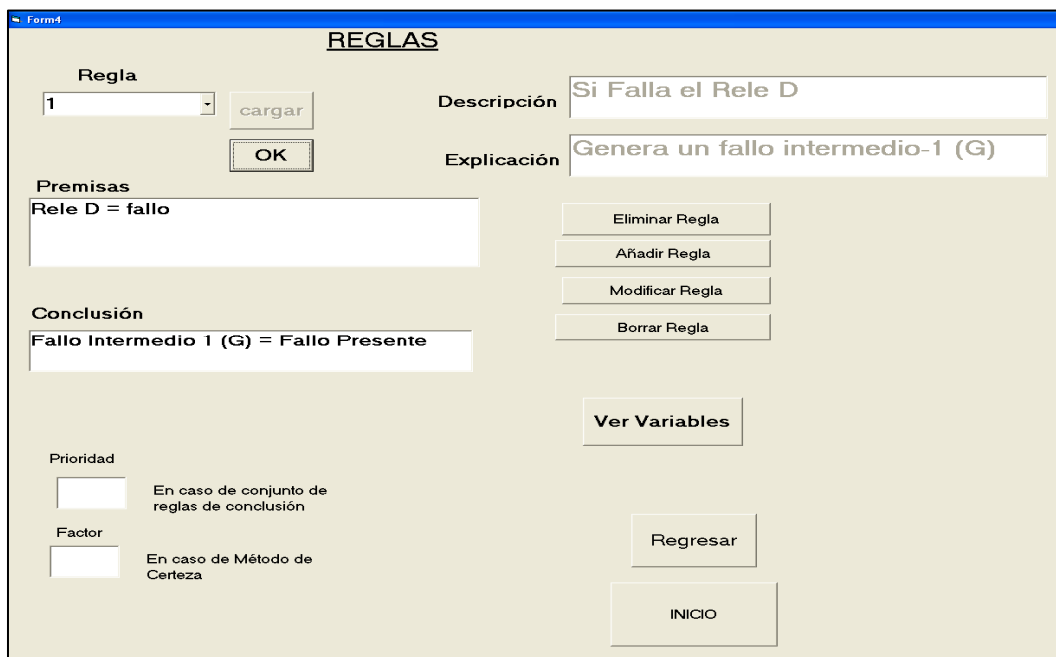
De forma similar, para modificar un conjunto de reglas dentro del sistema experto tenemos una ventana de trabajo que nos ayuda a realizar esta acción. Dentro del conjunto de reglas elegido aparece una opción “*modificar*” la cual permite realizar las acciones de modificación del conjunto de reglas dentro de esta misma ventana (Ver Figura 5.4). Únicamente se podrá modificar el nombre y la descripción del conjunto (No tipo y número).



**Figura 5.4. Ventanas que muestran el proceso para modificar un conjunto de Reglas dentro del Sistema Experto.**

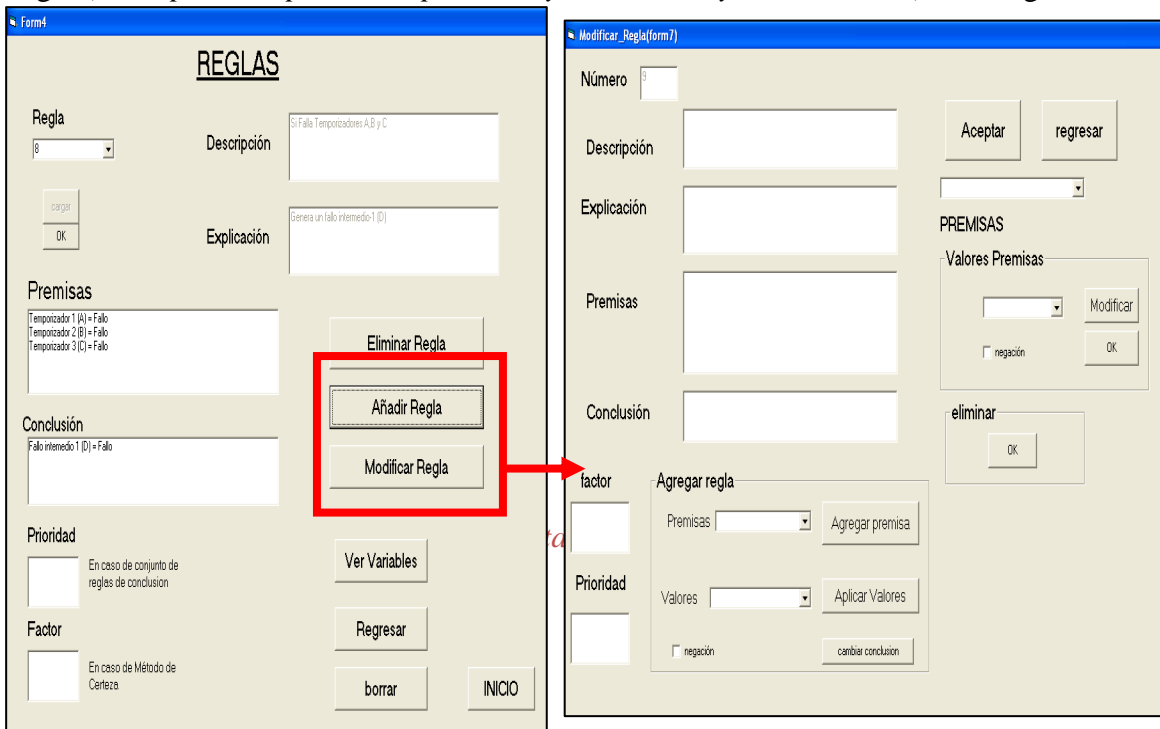
### 5.2.2 Creación y modificación de una Regla

La figura 5.5 muestra la forma de como se elige una regla que pertenece a un conjunto de reglas en particular. Una vez seleccionada la regla, se despliegan algunas de sus características en la pantalla de trabajo, como son: *descripción*, *explicación*, *premisas* y *conclusión*. La conclusión se llega de acuerdo a las reglas encadenadas (ver Figura 5.5).



**Figura 5.5. Despliegue de las características que pertenecen a una regla.**

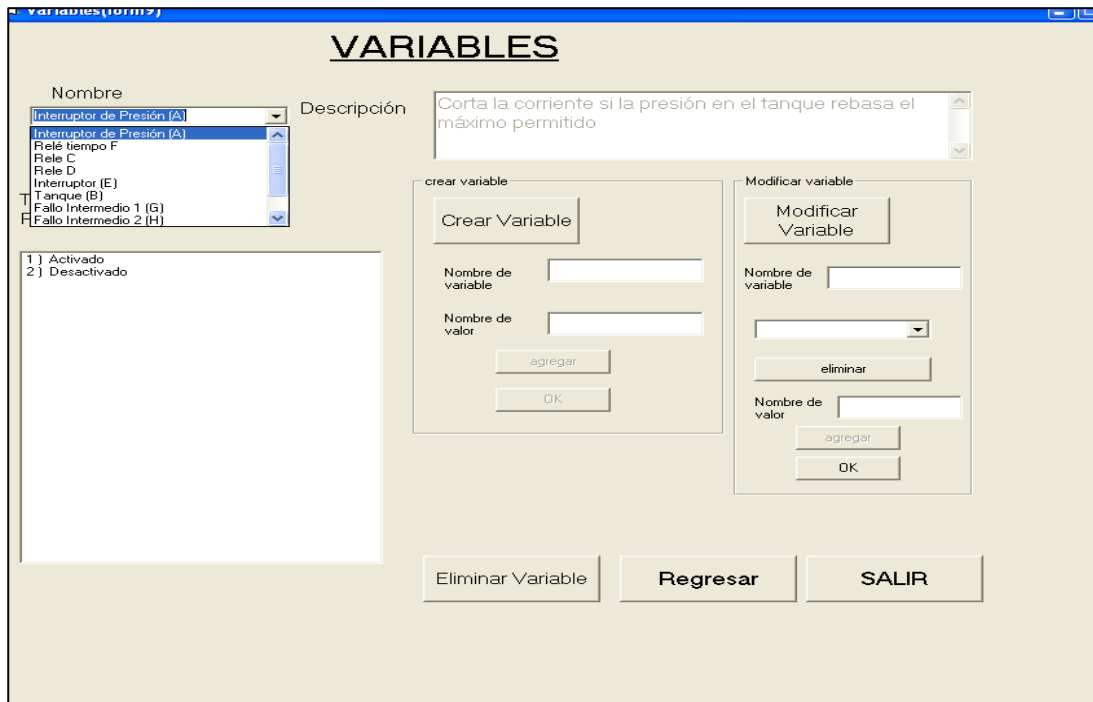
También podemos crear y modificar una o más reglas que pertenecen a un conjunto. Para la creación de una regla dentro de la ventana “Reglas” existe la opción “Añadir Regla” la cual nos permite crear una nueva regla dentro de un conjunto, las características que se deben proporcionar para cada regla añadida son: *descripción*, *explicación*, *premisas* y *sus valores* y la *conclusión* a la cual deberá llegar dicha regla. De igual manera para modificar una regla existe la opción “Modificar Regla” que también nos permite modificar las características mencionadas para la creación de una regla (*descripción*, *explicación*, *premisas* y *sus valores* y la *conclusión*). Ver Figura 5.6.



**Figura 5.6. Ventanas que muestran el proceso para añadir y/o modificar una Regla dentro del Sistema Experto.**

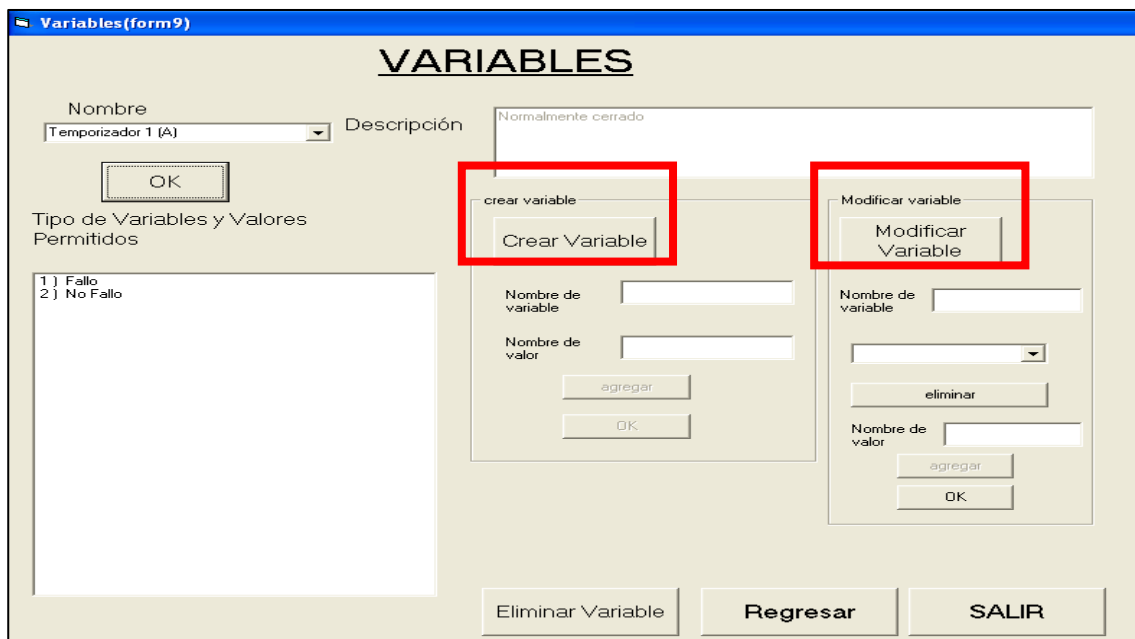
### 5.2.3 Creación y modificación de una Variable

La figura 5.6 muestra cómo se despliega la información que pertenece a una variable en particular la cual puede formar parte de una o más reglas dentro de un conjunto. Al seleccionar el botón de *Ver Variables*, enseguida la pantalla muestra una breve descripción de la variable así como los valores permitidos para ésta (Ver Figura 5.7).



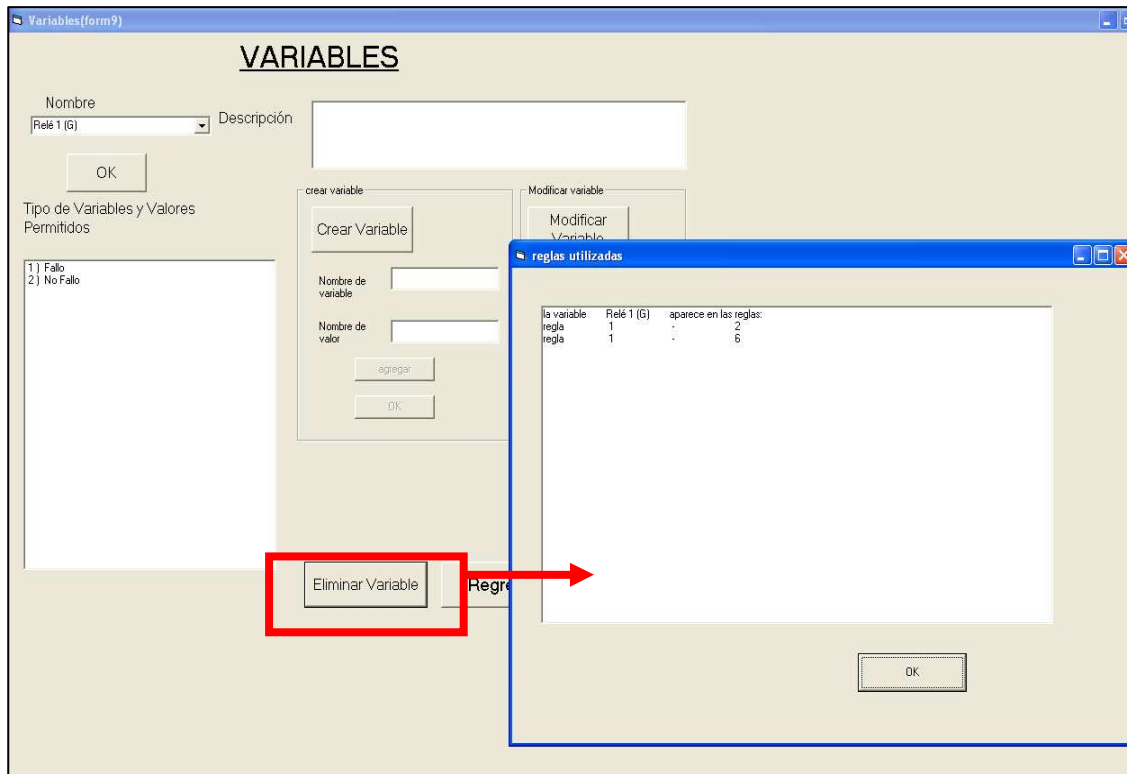
**Figura 5.7. Despliegue de la características de una Variable.**

De la misma forma que para un conjunto de Reglas y para una o más reglas, dentro del sistema experto podemos crear y modificar una o más variables. Dentro de la ventana VARIABLES aparecen las opciones “*Crear Variable*” y “*Modificar Variable*” que realizan dichas operaciones, las características que se deben de dar y/o modificar para una variable son: *descripción, nombre y valores permitidos*. Ver Figura 5.8



**Figura 5.8. Ventanas que muestran el proceso para añadir y/o modificar una Variable dentro del Sistema Experto.**

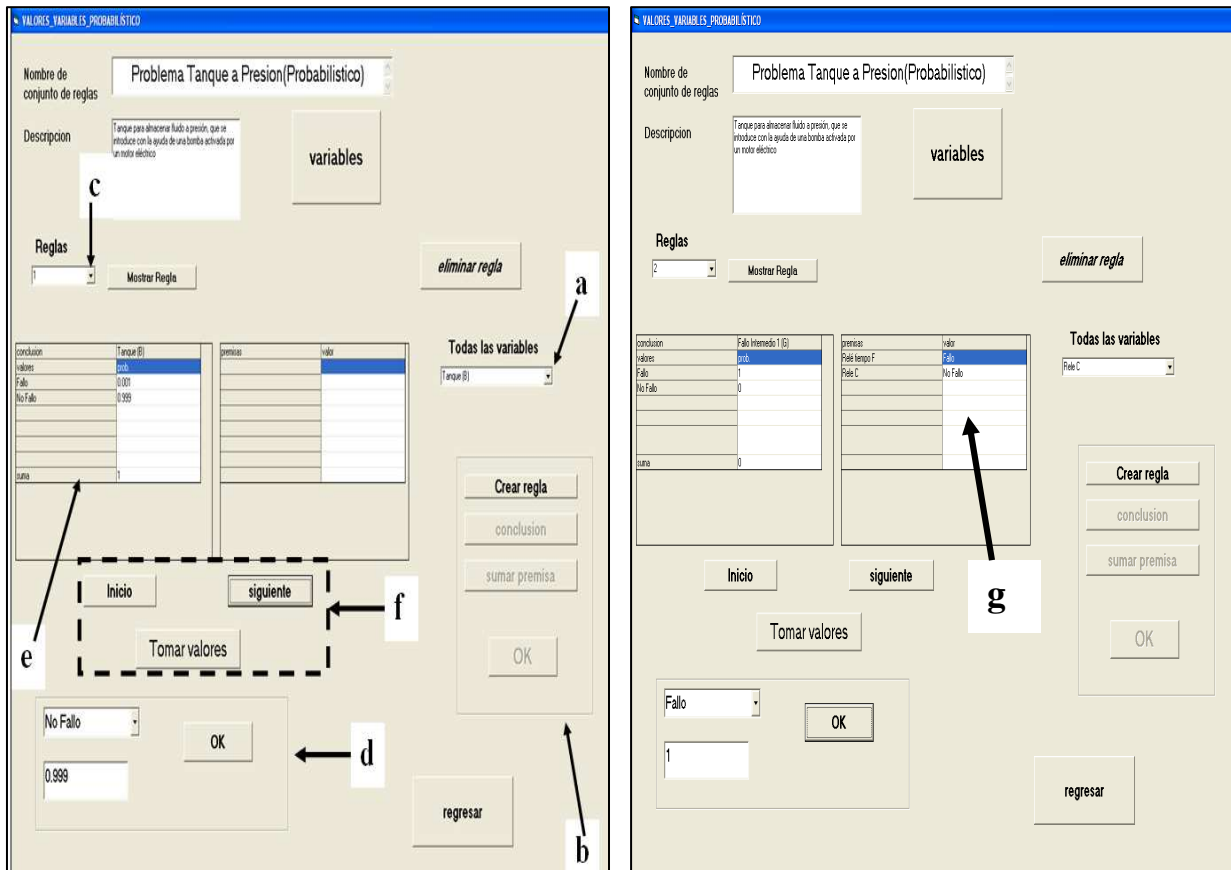
Por otra parte, el usuario debe conocer que si desea eliminar una variable y si dicha variable está incluida en otra regla, el programa no permitirá eliminar a ésta, como se muestra en la Figura 5.9. Aquí, si el usuario da la orden de eliminar la variable G que está dentro de la Regla 1, el programa no dejará realizar dicha acción, explicando que esta variable aparte de estar contenida en la regla 1 también forma parte de la regla 6.



**Figura 5.9. Ventanas que muestran un ejemplo de lo que es el Subsistema de Explicación.**

### 5.3 Subsistema de aprendizaje probabilístico

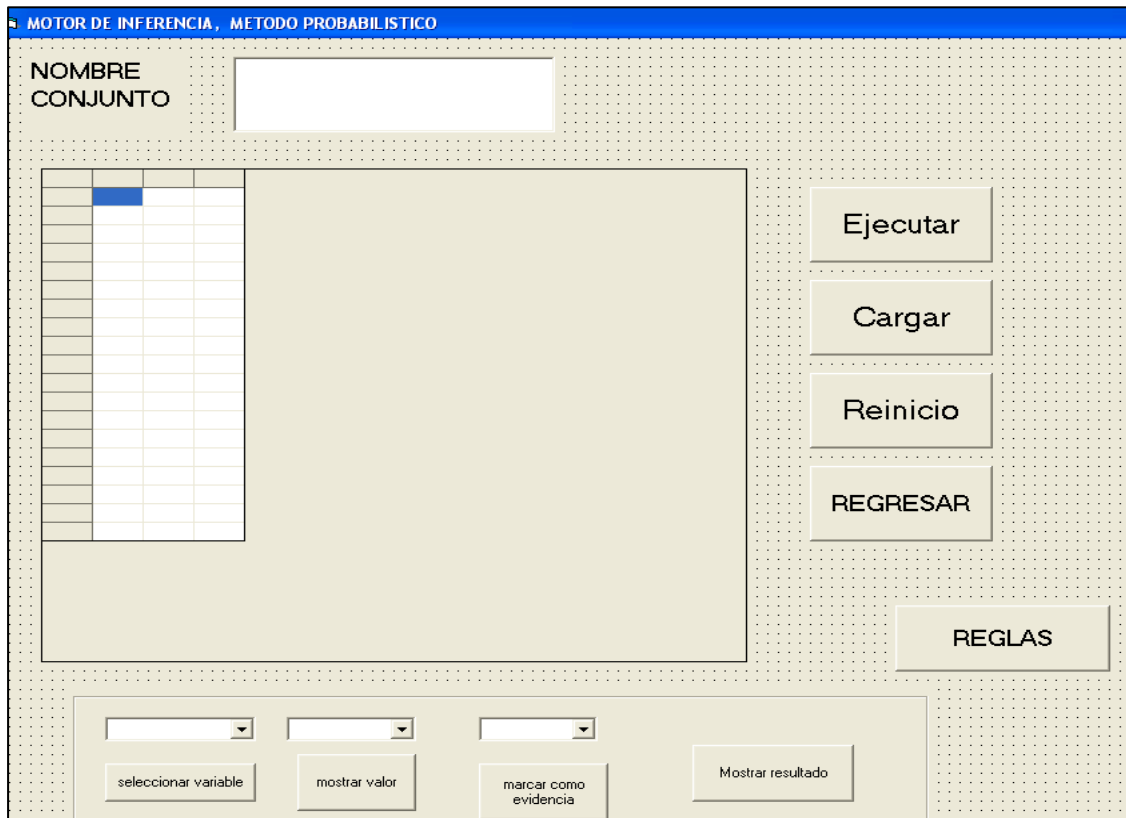
A continuación se muestra el procedimiento a seguir para crear las reglas de un conjunto del tipo probabilístico. Para esto, se da una breve explicación del funcionamiento de cada elemento que se encuentra en la ventana de trabajo (Ver Figura 5.10).



**Figura 5.10. Identificación de los elementos más importantes para crear una regla perteneciente a un conjunto del tipo probabilístico.**

- Muestra cada una de las variables que se definieron en el problema.
- Botones para crear las reglas con o sin premisas; para las que tienen premisas, existe el botón *sumar premisas*.
- Muestra el número de regla creada anteriormente.
- Aquí se asigna el valor a las probabilidades de Fallo y No fallo para las variables.
- Muestra los valores de las probabilidades asignados en d) y la suma de estas (debe ser uno).
- Toma los valores asignados en d) y pasa al siguiente valor.
- Muestra las probabilidades condicionales de fallo de las variables intermedias, debemos crear las reglas y sumar las premisas junto con sus valores correspondientes.

Una vez creadas todas las reglas con sus probabilidades correspondientes, ingresamos al Motor de Inferencia. Dentro de éste, mandamos llamar a todas las variables del problema con “Cargar”, apareciendo en la malla todas las variables de un conjunto. Aquí podemos ver el nombre de la variable, el valor de ésta, en caso de no ser evidencia aparece **NE**, de lo contrario será **Ev** y por último, en la columna llamada “Deducción” aparecerán las probabilidades de Fallo o No fallo de cada variable o elemento una vez que se ejecute el programa. Ver Figura 5.11.



**Figura 5.11. Motor de Inferencia para un conjunto de reglas del tipo probabilístico.**

#### 5.4. Subsistema de aprendizaje por Lógica Difusa

A continuación se describe la función principal de cada elemento que conforma el subsistema de aprendizaje por Lógica Difusa

- I. Al seleccionar una variable muestra el nombre y los valores de esta.
- II. Se introducen los valores numéricos de los límites inferior y superior (conjuntos) permitidos por el valor seleccionado de la variable.
- III. Una vez introducidos los valores a los límites inferior y superior, con el botón “OK” se carga la nueva regla, y en “Número de Reglas” se van contabilizando el total de las reglas.
- IV. El botón de “Inferencia” manda directamente al Motor de Inferencia (Ver Figura 5.13). “Ver Variables” manda a la ventana de VARIABLES similar a la Figura 5.7, donde se muestran las características de cada una de las variables con las que se está trabajando. “Regresar” manda a la ventana anterior de REGLAS.

The screenshot shows a software window titled "reglas de entrada-especificacion". At the top, there is a text field for "Nombre de Conjunto de Reglas" containing the word "DIFUSO". Below this, there are several sections:

- A section for "variables" with a dropdown menu showing "PRESION" and a "Seleccionar" button.
- A section for "valores" with a dropdown menu showing "BAJO" (highlighted) and a list of options: "BAJO", "MEDIO", "ALTO", and "MÁXIMO PERMITIDO".
- A section for numerical limits with four input fields: "Limite inferior" (0.5), "Limite superior" (1), "Limite inferior 1" (0.6), and "Limite superior 1" (1.2).
- A "Valor de la variable" input field containing the number "0".
- A "Numero de reglas" input field containing the number "3".
- A central "OK" button.
- A vertical panel on the right side containing three buttons: "regresar", "ver variables", and "Inferencia".

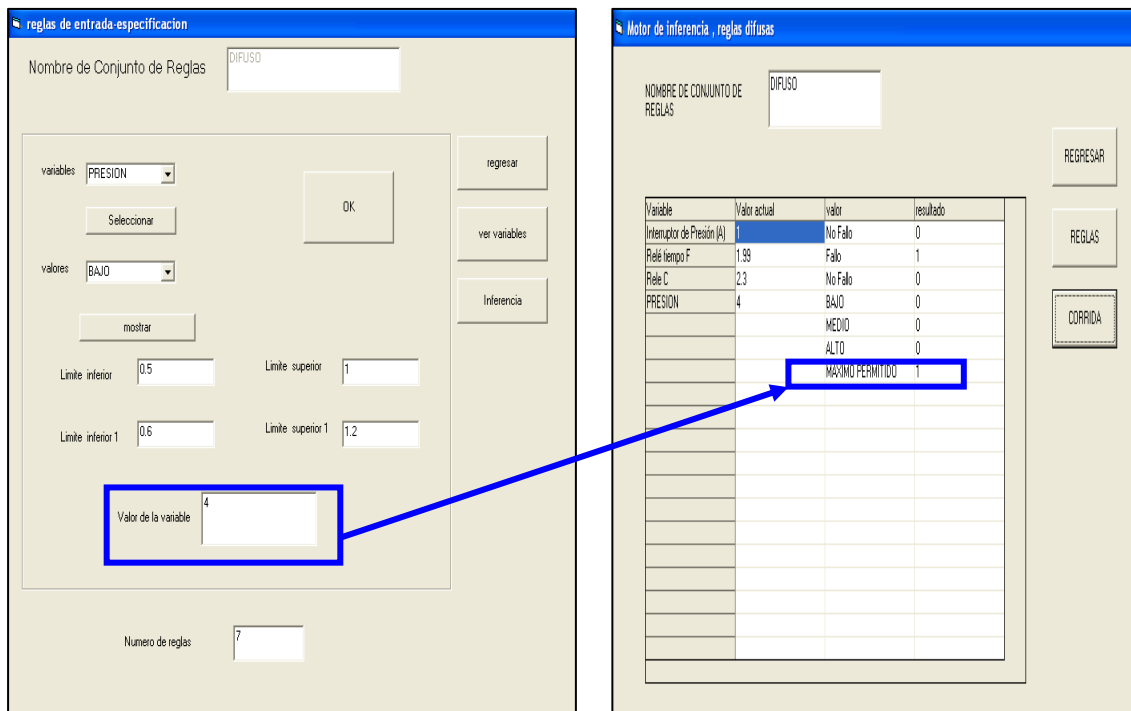
Red boxes and arrows labeled with Roman numerals I, II, III, and IV highlight specific areas: I points to the "valores" dropdown; II points to the limit input fields; III points to the "Numero de reglas" field; and IV points to the "Inferencia" button.

**Figura 5.12. Subsistema de Aprendizaje para Método de Lógica Difusa.**

En la Figura 5.13 se muestra cómo trabaja el Motor de Inferencia por Lógica difusa, aquí podemos ver que si el usuario seleccionó el valor BAJO de la variable PRESIÓN, enseguida aparecen los valores de los límites inferior y superior permitidos para este rango. Si a la variable se le da un valor, por ejemplo de 4; al momento de realizar la inferencia se puede apreciar que el valor de 4 se encuentra dentro del rango



MÁXIMO PERMITIDO. Por lo que se puede notar que al correr el Motor de Inferencia, este mostrará los resultados de las variables, que para Lógica Difusa corresponde al grado de pertenencia de la variable a un conjunto.



**Figura 5.13. Motor de Inferencia para un conjunto de reglas del tipo Lógica Difusa.**

### 5.5 Subsistema de aprendizaje método de explicaciones

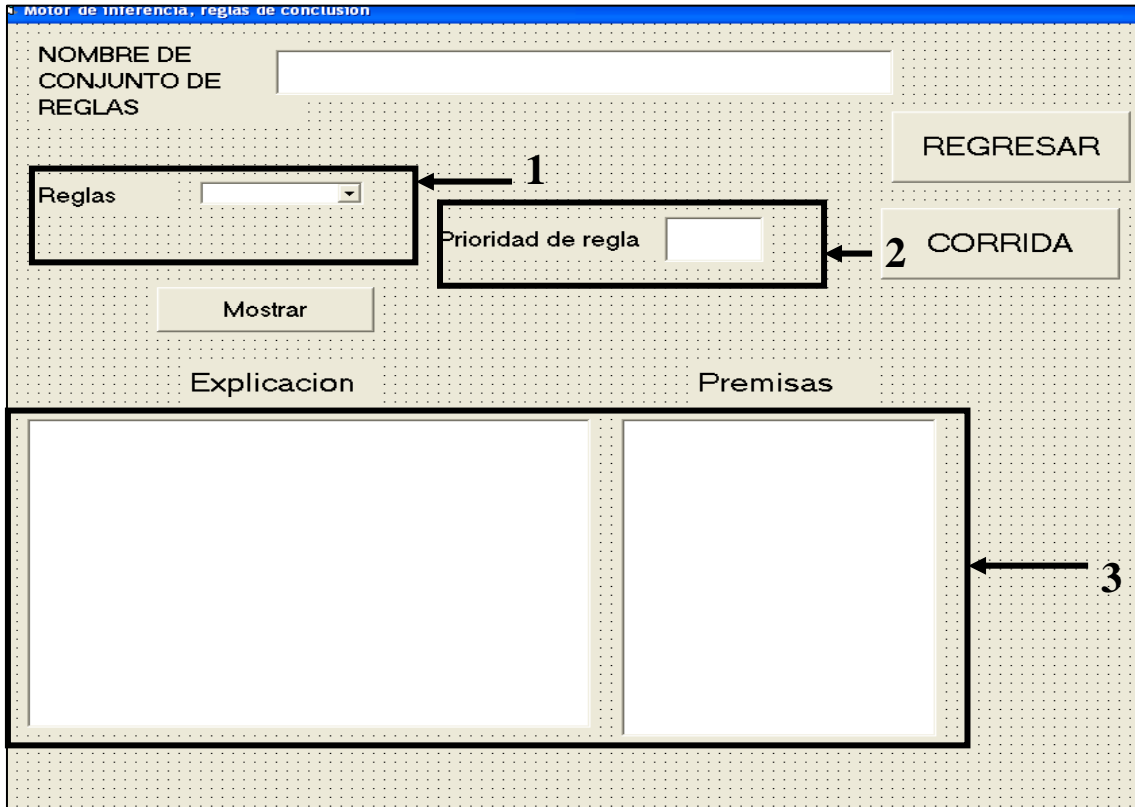
Dentro del Sistema Experto se agregó el método de explicaciones, el cual depende del factor de prioridad de una regla, es decir, muestra una recomendación, comentario o acción a realizar en caso de presentarse un fallo en algún elemento crítico, el cual representa una verificación muy crítica aunque no sea muy probable y que debería realizarse antes.

En la Figura 5.14 se muestra la ventana del Motor de Inferencia para el método de explicaciones. A continuación se describe el funcionamiento principal de cada elemento.

1. Despliega el total de reglas que contiene un conjunto para seleccionar una regla.
2. Muestra el factor de prioridad de la regla seleccionada.

3. Muestra la explicación de la regla y las premisas de ésta.

Por último el botón “*CORRIDA*” ejecutará el motor para dar la explicación o recomendación más probable de acuerdo al factor de prioridad de la regla.



**Figura 5.14. Motor de Inferencia para un conjunto de reglas del tipo Explicaciones.**

## VI. APLICACIÓN DEL SISTEMA EXPERTO

### 6.1. Definición del problema. SISTEMA DEL TANQUE DE PRESIÓN (Caso 1).

La Figura 6.1 muestra un diagrama de un tanque de presión con sus elementos más importantes. Se trata de un tanque para almacenar un fluido a presión, que se introduce con la ayuda de una bomba activada por un motor eléctrico. Se sabe que el tanque no tiene problemas si la bomba funciona durante un periodo inferior a un minuto. Por tanto, se incorpora un mecanismo de seguridad, basado en un relé F, que interrumpe la corriente tras funcionar 60 segundos. Además, un interruptor de presión A, corta también la corriente si la presión del tanque alcanza un cierto valor umbral, que se considera peligroso. El sistema incluye un interruptor, E, que inicia la operación del sistema; un relé D, que suministra corriente tras la etapa de iniciación y la interrumpe tras la activación del relé F; y el C, que activa la operación del circuito eléctrico del motor. El objetivo del estudio consiste en conocer la probabilidad de fallo del tanque a presión.

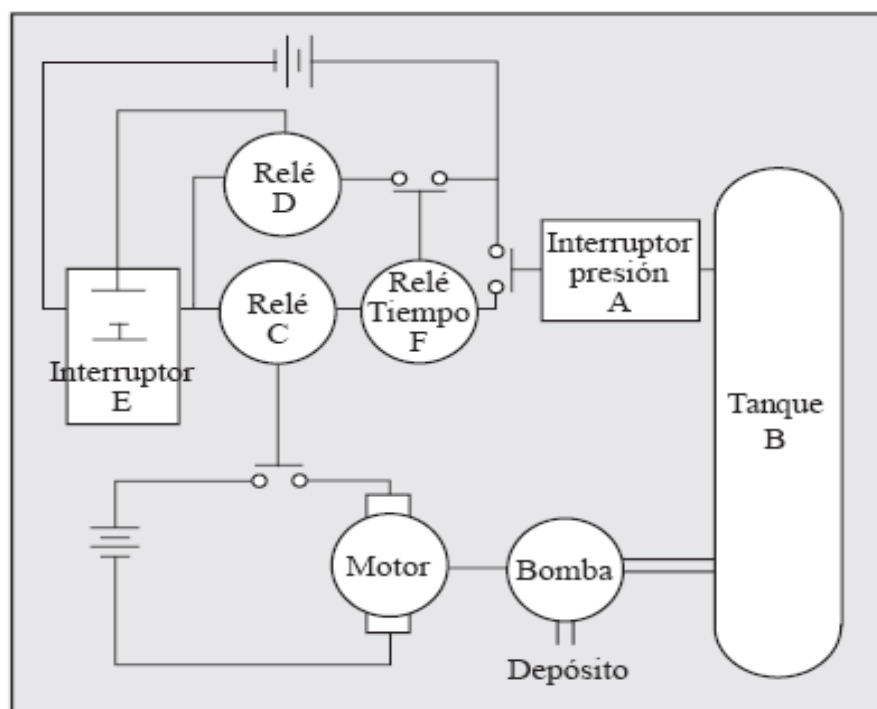


Figura 6.1. Diagrama del Sistema del tanque a presión.

## 6.2 Selección de variables

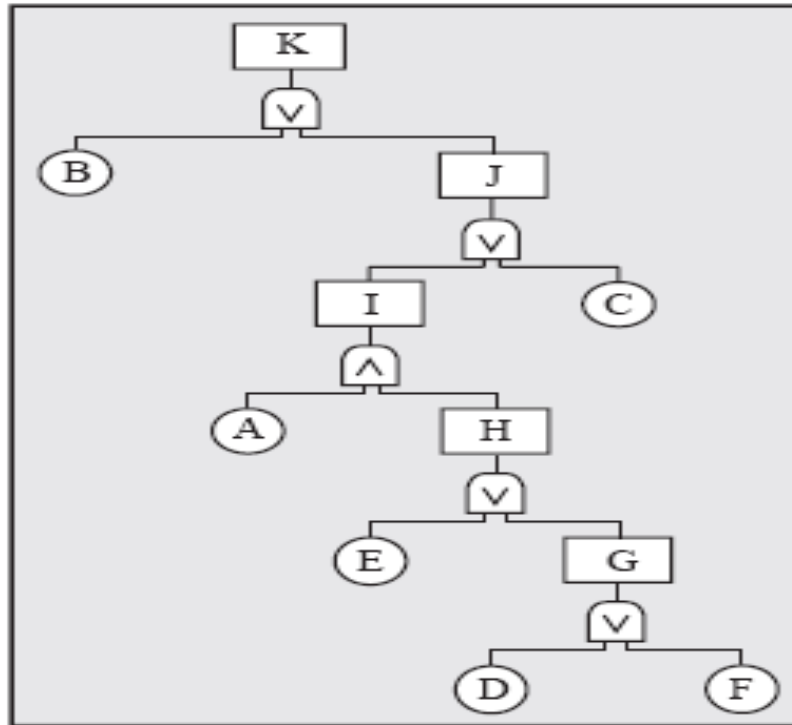
Puesto que se está interesado en el análisis de todas las posibles causas de fallo del tanque B, se introduce una variable K que denota este suceso. Se usará la notación  $K = k$  para indicar la falla del tanque. Similarmente,  $a, \dots, f$  representan los fallos de las respectivas componentes A, ..., F.

Basándose en la descripción previa del problema, se puede escribir la siguiente expresión lógica para el fallo del tanque:

$$\begin{aligned} k &= b \vee c \vee (a \wedge e) \vee (a \wedge d) \vee (a \wedge f) & (6.1) \\ &= b \vee c \vee (a \wedge (e \vee d \vee f)) \end{aligned}$$

donde los símbolos  $\vee$  y  $\wedge$  se usan para operaciones lógicas *or* y *and*, respectivamente. Esta expresión se obtiene combinando todas las posibilidades de fallo de las diferentes componentes que conducen al fallo del tanque.

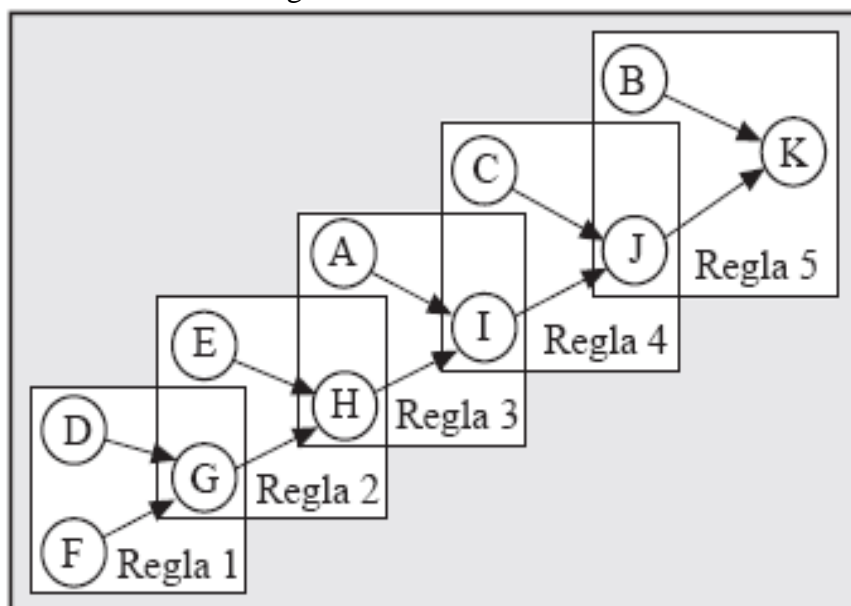
La ecuación (6.1) constituye la base para obtener el conjunto de reglas (para un sistema experto basado en reglas) o la estructura de dependencia (de un sistema de red probabilística). Esta ecuación puede expresarse de una forma mucho más intuitiva usando lo que se llama *árbol de fallos*. La figura 6.2 muestra el árbol de fallos correspondiente a la expresión 6.1. En este árbol, los fallos de los relés D y F se combinan para dar una causa de fallo intermedia, G; seguidamente G se combina con E para definir otra causa intermedia, H, y así sucesivamente. Este árbol incluye las variables iniciales {A, ..., F} así como los fallos intermedios {G, ..., J} que implican el fallo del tanque. Por tanto, el conjunto final de variables usadas en este ejemplo es  $X = \{A, \dots, K\}$ .



**Figura 6.2.** Árbol de fallos del sistema del tanque a presión.

### 6.3 Selección del Modelo de Solución

Este ejemplo puede ser analizado desde un punto de vista determinístico usando los sistemas expertos basado en reglas. La figura 6.3 muestra las reglas que resultan del árbol de fallos de la figura 6.2 así como el encadenamiento entre las premisas y las conclusiones de las diferentes reglas.



### Figura 6.3. Reglas encadenadas para el sistema del tanque a presión.

Las reglas establecen las condiciones bajo las cuales puede derivarse la verdad de los objetos conclusiones a partir de la verdad de los objetos premisas. Sin embargo, los objetos pueden tener asociada una cierta medida de incertidumbre, tal como una probabilidad de ser ciertos. Además, las reglas pueden contener cierta información que permita obtener la incertidumbre de las conclusiones a partir de las incertidumbres de los objetos en las premisas de dichas reglas. De hecho, el problema clave consiste en obtener la incertidumbre asociada a algunos objetos (los objetos que figuran en las conclusiones de las reglas) cuando se conocen la de otros objetos (los objetos que figuran en la premisa de las reglas). Desde un punto de vista estadístico, se puede dar una interpretación mucho más amplia a estas reglas dando las tablas de probabilidades condicionales. Por ejemplo, la incertidumbre de las reglas “Si A y B y C entonces D” y “Si A o B o C entonces D” puede medirse mediante la probabilidad condicional  $p(d/a, b, c)$ . Esto incluye las reglas deterministas con

$$p(d/a, b, c) = \begin{cases} 1, & \text{si } A = B = C = \text{cierto} \\ 0, & \text{en otro caso} \end{cases} \quad (6.2)$$

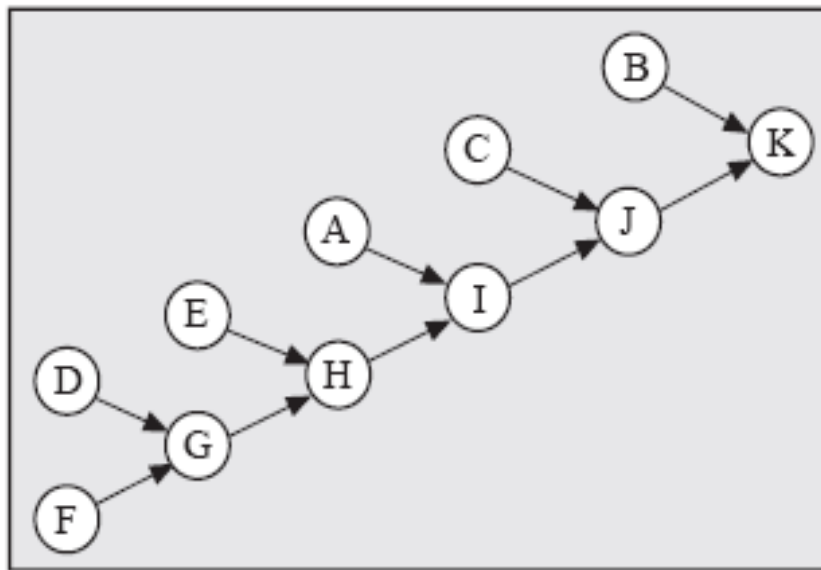
para la primera regla, y

$$p(d/a, b, c) = \begin{cases} 1, & \text{si } A \text{ o } B \text{ o } C \text{ es cierto} \\ 0, & \text{en otro caso} \end{cases} \quad (6.3)$$

para la segunda.

La estructura dada por las reglas encadenadas puede utilizarse para definir la estructura gráfica de un sistema experto probabilístico. Por ejemplo, puesto que los fallos de las diferentes componentes del sistema son las causas de los fallos intermedios y, finalmente, del fallo del tanque, se puede obtener un grafo dirigido que reproduzca estas dependencias entre las variables que intervienen en el modelo (Figura 6.4). De este grafo se deduce que la función de probabilidad conjunta de todos los nodos puede escribirse en la forma:

$$p(x) = p(a)p(b)p(d)p(e)p(f)p(g|d,f)p(h|e,g)p(i|a,h)p(j|c,i)p(k|b,j) \quad (6.4)$$



**Figura 6.4. Grafo dirigido correspondiente al tanque de presión.**

Por otra parte, las distribuciones de probabilidad condicionales asociadas a las causas intermedias en el árbol de fallos, se definen utilizando (6b) y (6c) tal como se muestra en la Tabla 6.1, donde se dan sólo las probabilidades condicionales de los fallos, puesto que  $p(\text{no fallo}) = 1 - p(\text{fallo})$ . Por otra parte, las probabilidades marginales asociadas a las componentes del sistema representan las probabilidades iniciales de fallo de cada una de sus componentes. Supóngase que las probabilidades son

$$\begin{array}{lll} p(a)=0.002 & p(b)= 0.001 & p(c)= 0.003, \\ p(d)= 0.010 & p(e)= 0.001 & p(f)= 0.010 \end{array} \quad (6.5)$$

D	F	$p(g D,F)$
d	f	1
d	$\neg f$	1
$\neg d$	f	1
$\neg d$	$\neg f$	0

E	G	$p(h E,G)$
e	g	1
e	$\neg g$	1
$\neg e$	g	1
$\neg e$	$\neg g$	0

A	H	$p(i A,H)$
a	h	1
a	$\neg h$	0
$\neg a$	h	0
$\neg a$	$\neg h$	0

C	I	$p(j C,I)$
c	i	1
c	$\neg i$	1
$\neg c$	i	1
$\neg c$	$\neg i$	0

B	J	$p(k B,J)$
b	j	1
b	$\neg j$	1
$\neg b$	j	1
$\neg b$	$\neg j$	0

**Tabla 6.1. Probabilidades condicionales de fallo de las variables intermedias en el sistema del tanque a presión.**

El grafo de la Figura 6.4, junto con las tablas de probabilidad marginal que se muestran en (6.5) y en la Tabla 6.1, define una red Bayesiana que corresponde al ejemplo del tanque a presión. La correspondiente función de probabilidad conjunta se da en (6.4).

#### 6.4. Propagación de Evidencia

El grafo de la Figura 6.4 es un poliárbol, lo que significa que se puede utilizar el algoritmo para poliárboles (Apéndice D) para la propagación de la evidencia.

##### Caso 1

Supóngase, en primer lugar que no hay evidencia disponible.

##### Caso 2

Componentes F y D fallan, es decir, se tiene la evidencia  $F = f, D = d$ .

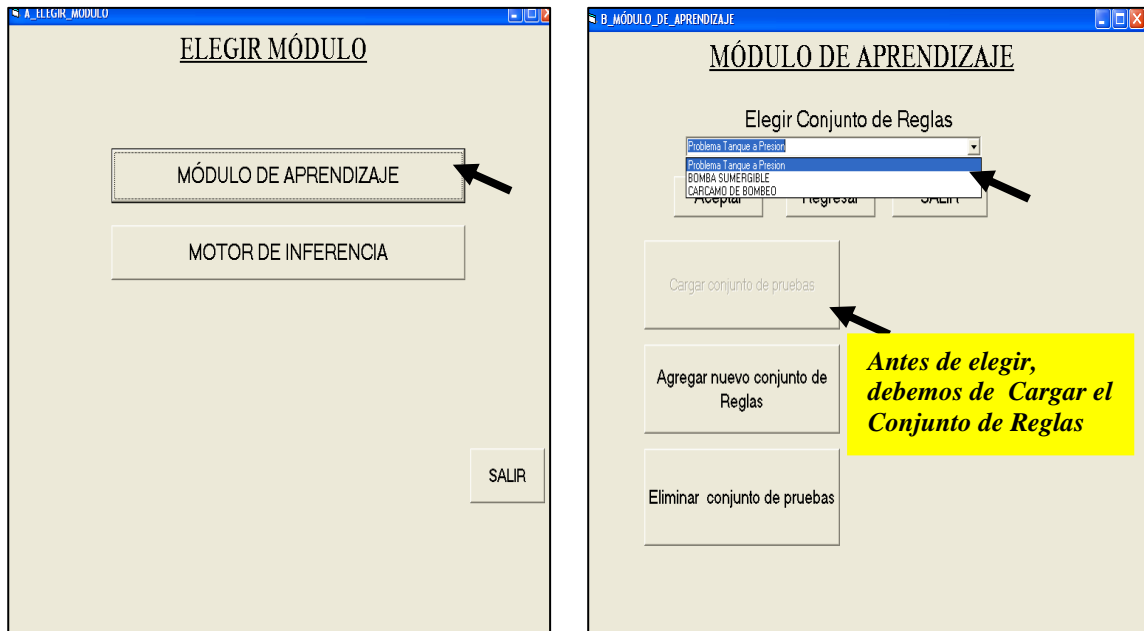
##### Caso 3

El interruptor A también falla ( $A = a$ ); es decir,  $F = f, D = d, A = a$ .



### 6.5 Programación de Reglas y Variables.

Al correr el Sistema Experto entramos a Módulo de Aprendizaje y cargamos el conjunto de reglas para el sistema del Tanque a Presión (Figura 6.5)



**Figura 6.5. Selección del conjunto de Reglas en el Sistema Experto para el sistema del tanque a presión.**

Una vez elegido el conjunto de Reglas, el programa muestra las características más importantes de este conjunto, como son: *Nombre*, una breve *descripción*, *tipo* y el *número* (Ver Figura 6.6). Para ver las Reglas que pertenecen a dicho conjunto, seleccionamos el botón *Ver Reglas*.

**CONJUNTO DE REGLAS**

Nombre: Problema Tanque a Presion

Número de Conjunto de Reglas: 1

Descripción: Tanque para almacenar fluido a presión, que se introduce con la ayuda de una bomba activada por un motor eléctrico

Tipo: exacto

Buttons: REGRESAR, SALIR, Ver Reglas, Modificar, aceptar

Callout: *Para ver las reglas pertenecientes a un conjunto*

**Figura 6.6. Características del conjunto de Reglas para el sistema del tanque a presión.**

La figura 6.7 muestra el número de reglas que pertenecen al conjunto del sistema del tanque a Presión, para este problema el total de reglas son seis. La conclusión se llega de acuerdo al método de encadenamiento de reglas.

**REGLAS**

Regla: 1

Descripción: Si Falla el Relé Tiempo y Rele D

Explicación: Genera un fallo intermedio-1 (G)

Premisas: Fallo Intermedio 1 (G) = Presente  
Rele D = Desactivado

Conclusión: Fallo Intermedio 1 (G) = Presente

Prioridad:  En caso de conjunto de reglas de conclusion

Factor:  En caso de Método de Certeza

Buttons: Eliminar Regla, Añadir Regla, Modificar Regla, Ver Variables, Regresar, borrar

Callout: *Para ver las variables pertenecientes a un conjunto*

**Figura 6.7. Reglas que pertenecen al conjunto del sistema del tanque a presión.**

La Figura 6.8 nos muestra las características de las variables utilizadas para el problema del tanque a presión.

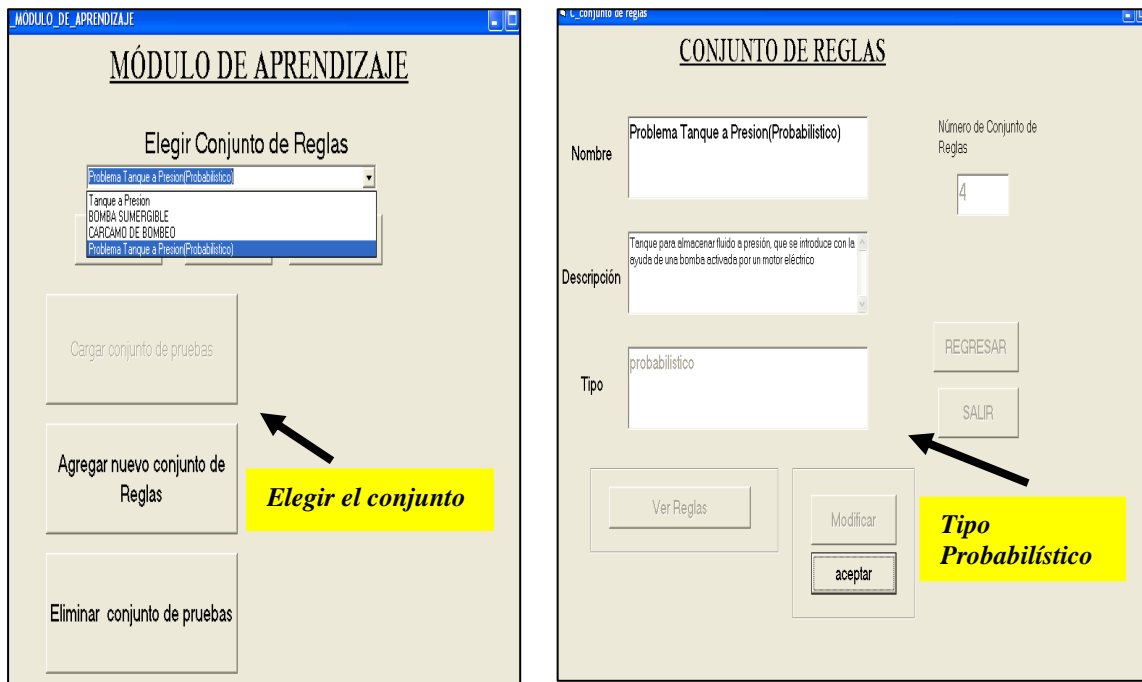
The screenshot shows a window titled "Variables(form9)" with the following components:

- Nombre:** A dropdown menu showing "Interruptor de Presión (A)".
- OK:** A button next to the name field.
- Descripción:** A text area containing "Corta la corriente si la presión en el tanque rebasa el máximo permitido".
- Valores Permitidos:** A list box containing "1) Fallo" and "2) No Fallo".
- Crear variable:** A panel with input fields for "Nombre de variable" and "Valor", and buttons "Crear Variable", "agregar", and "OK".
- Modificar variable:** A panel with input fields for "Nombre de variable" and "Valor", and buttons "eliminar", "agregar", "Modificar Variable", and "OK".
- Eliminar Variable:** A button centered below the "Crear variable" and "Modificar variable" panels.
- Regresar:** A button at the bottom right.
- SALIR:** A button at the bottom right, below "Regresar".

**Figura 6.8. Variables que pertenecen al conjunto del sistema del tanque a presión.**

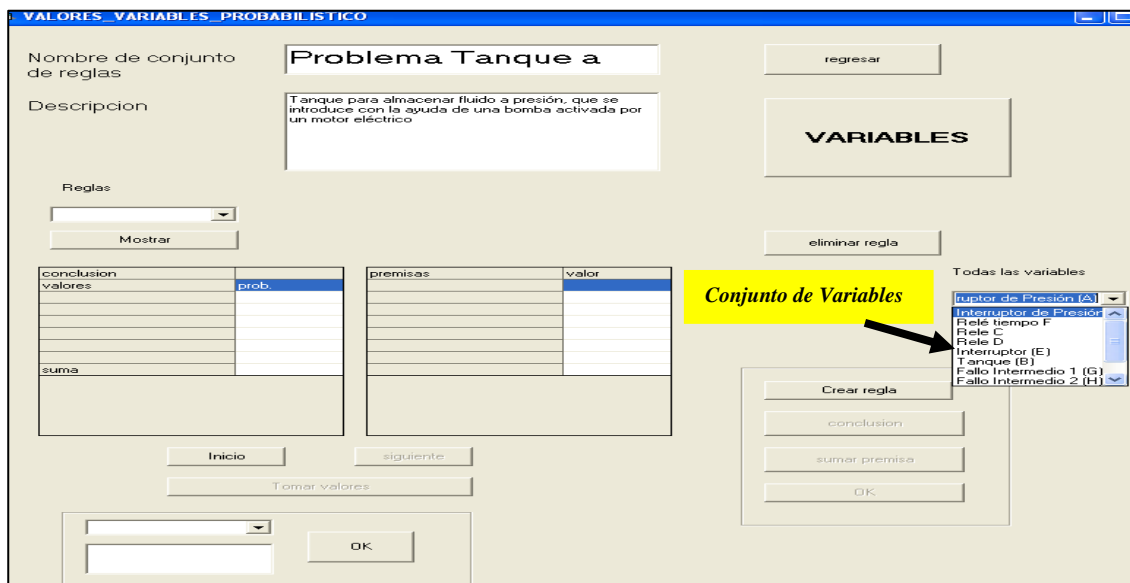
### 6.6 Programación del Método Probabilístico.

Para este método se programó el Algoritmo de propagación de evidencia para Poliárboles (Ver Anexo D). Estando dentro de Sistema Experto (programa) y dentro del módulo de Aprendizaje, escogemos el problema en cuestión (Tanque a presión) como se muestra en la Figura 6.9a y después se puede observar que ahora el tipo de conjunto de reglas es Probabilístico (Figura 6.9b).



**Figura 6.9. a) Elección del conjunto de reglas para el problema del Tanque a Presión, b) El conjunto de reglas ahora es del Tipo Probabilístico.**

Una vez elegido el conjunto de reglas probabilístico, con el botón “Ver Reglas” aparecerá la siguiente ventana (Figura 6.10) en la cual su principal función es mostrar el total de las variables utilizadas para este problema y enseguida crear las reglas y asignarle a cada una sus probabilidades marginales y/o condicionadas según sea el caso.



**Figura 6.10. Ventana para crear reglas para el Método Probabilístico sistema del tanque a presión.**

Ingresando al Motor de Inferencia mandamos llamar a todas las variables del problema con “Cargar”. La Figura 6.11 muestra los datos iniciales de cada variable, el nombre, el valor de esta, en caso de no ser evidencia aparece **NE**, de lo contrario será **Ev** y por último en la columna *Deducción* aparecen valores de cero cuando no se ha ejecutado el motor de inferencia.

**MOTOR DE INFERENCIA, METODO PROBABILISTICO**

NOMBRE CONJUNTO: Problema Tanque a P

VARIABLE	VALOR	EVIDENCIA	DEDUCCION
Interruptor de Presión (A)	No Fallo	NE	0
Tanque (B)	Fallo	NE	0
Relé C	No Fallo	NE	0
Relé D	No fallo	NE	0
Interruptor (E)	No fallo	NE	0
Relé tiempo F	Fallo	NE	0
Fallo Intermedio 1 (G)	Fallo	NE	0
Fallo Intermedio 2 (H)	Fallo	NE	0
Fallo Intermedio 3 (I)	Fallo	NE	0
Fallo Intermedio 4 (J)	Fallo	NE	0
Fallo Total del Sistema (K)	Fallo	NE	0

Buttons: Ejecutar, Cargar, Reinicio, REGRESAR

Bottom controls: [dropdown], [dropdown], [1 dropdown], [Mostrar resultado], [seleccionar variable], [mostrar valor], [marcar como evidencia]

**Figura 6.11. Motor de Inferencia que muestra las variables para el problema del tanque a presión.**

6.7. Definición del problema. SISTEMA DE DISTRIBUCIÓN DE ENERGÍA (Caso 2).

La Figura 6.12 muestra un sistema de distribución con tres motores 1,2 y 3 y tres temporizadores, A, B y C, que están normalmente cerrados. Una pulsación momentánea del pulsador F suministra energía de una batería a los relés G e I. A partir de ese instante G e I se cierran y permanecen activados eléctricamente. Para comprobar si los tres motores están operando correctamente, se envía una señal de prueba de 60 segundos a través de K. Una vez que K se ha cerrado, la energía de la batería 1 llega a los relés R y M. El cierre de R arranca el motor 1. El cierre de T envía energía de la batería 1 a S. El cierre de S arranca el motor 3.

Tras un intervalo de 60 segundos, K debe abrirse, interrumpiendo la operación de los tres motores. Si K dejase de abrirse tras los 60 segundos, los tres temporizadores A, B y C actúan de forma similar para el motor 2 ó el motor 3, por lo que M o S deberían dejar de estar cerrados. A continuación se analiza sólo el efecto sobre el motor 2.

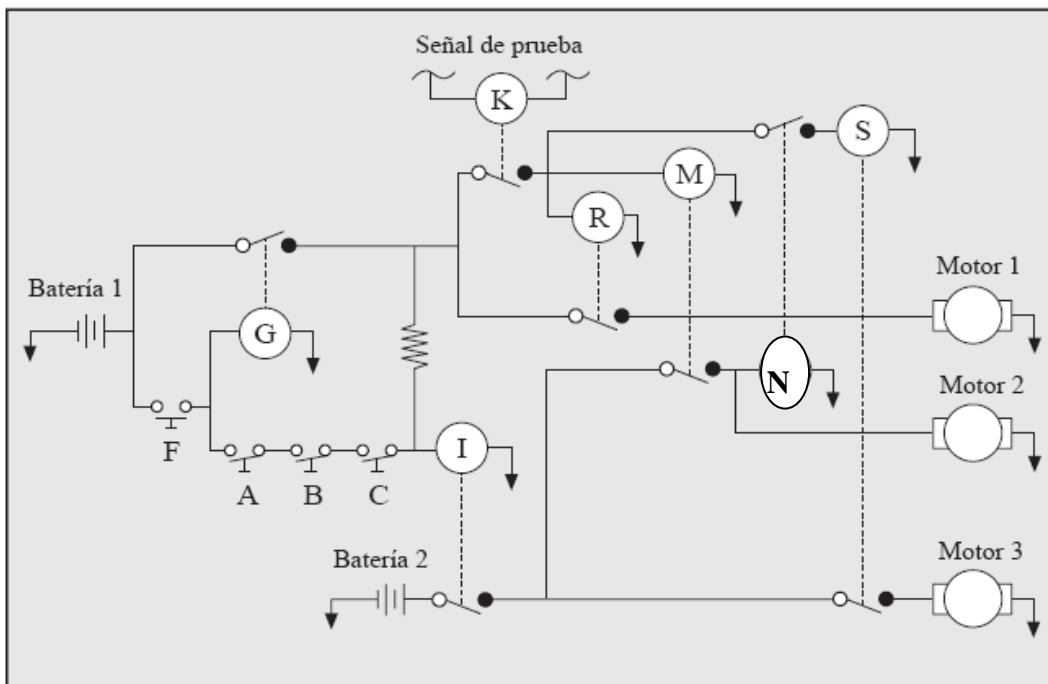


Figura 6.12. Diagrama del Sistema de distribución de Energía.

Se está interesado en conocer la probabilidad de fallo del motor 2. Denotemos a esta variable aleatoria por  $Q$ . Por lo tanto  $q$  significa fallo y  $\neg q$  significa no fallo. La Figura 6.13 muestra el árbol de fallos y los conjuntos que conducen al fallo del sistema. Nótese que el fallo del motor 2 es igual a la expresión lógica

$$q = [(m \vee n \vee (k \wedge g) \vee k \wedge (a \wedge b \wedge c) \vee (k \wedge f))] \wedge (i \vee g \vee b \vee f) \quad (6.6)$$

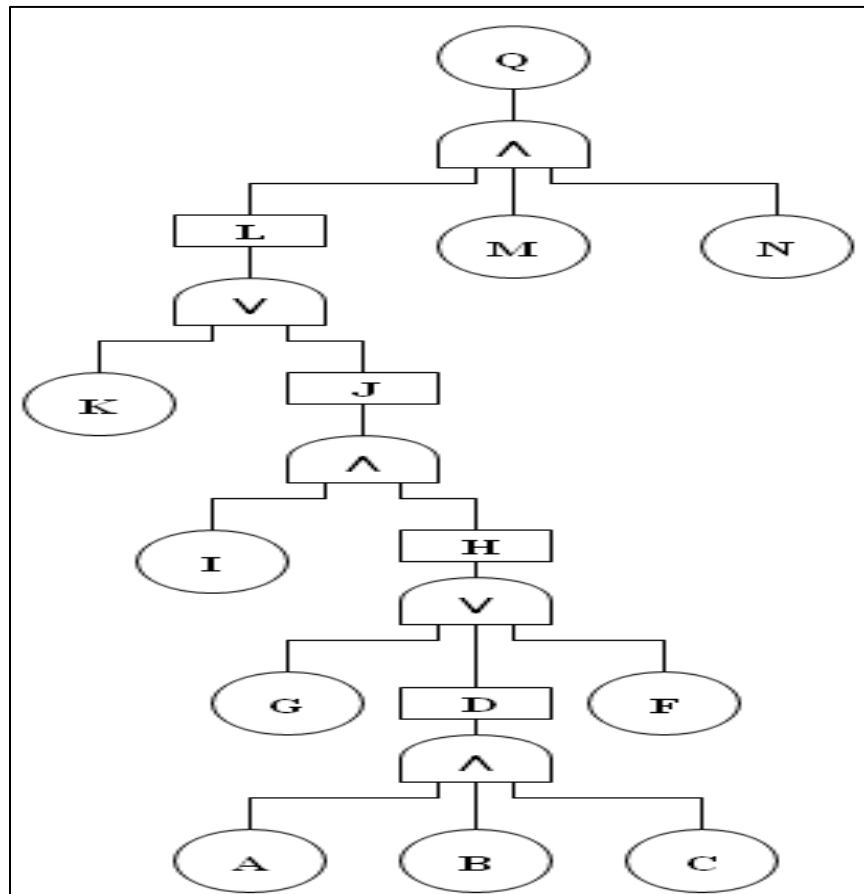


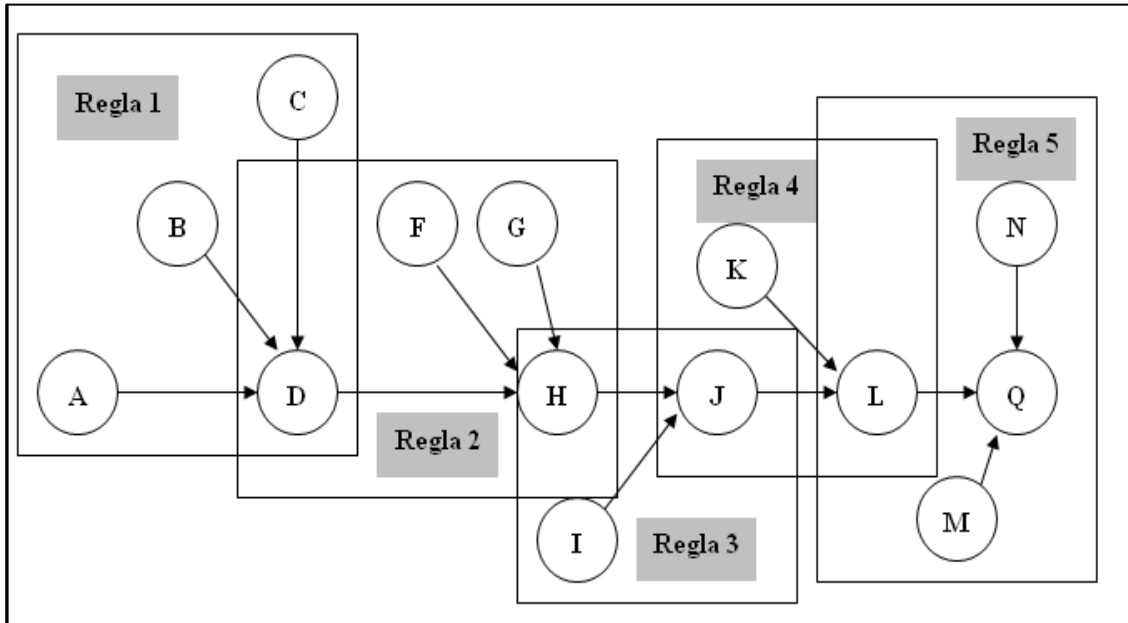
Figura 6.13. Árbol de fallos para el motor 2.

La ecuación (6.2) puede utilizarse para obtener el conjunto de reglas de un sistema experto determinista. El conjunto de variables usadas en este ejemplo es

$$X = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, Q\}$$

donde D, E, L y N son fallos intermedios.

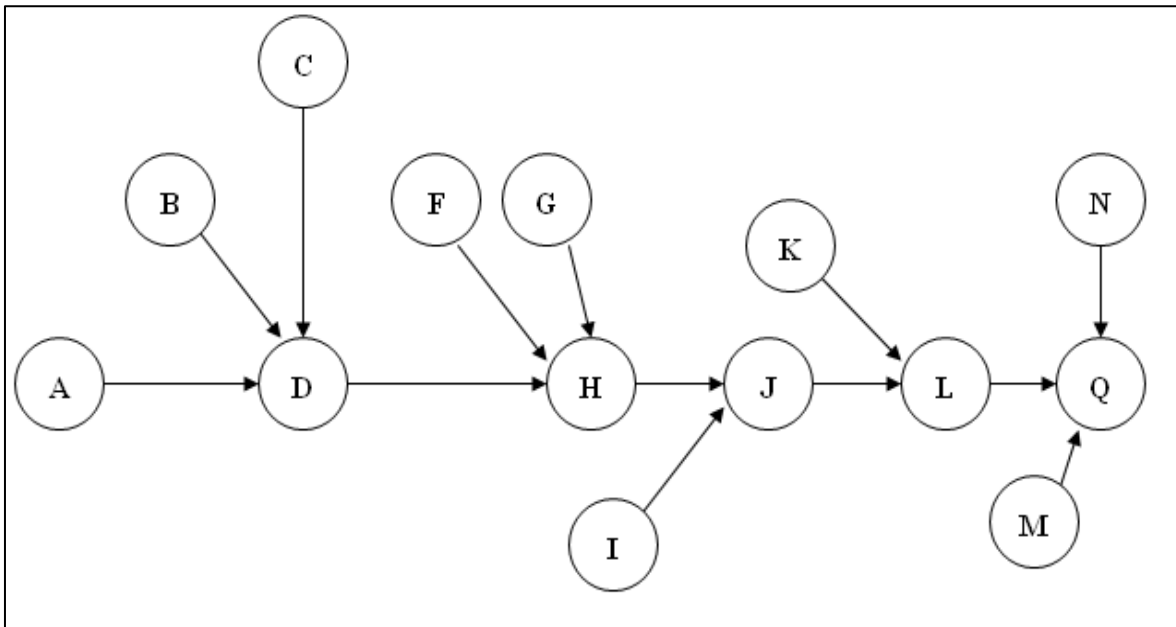
La figura 6.14 muestra el conjunto de reglas obtenidos de (6.12).



**Figura 6.14. Reglas encadenadas para el motor 2.**

Para este ejemplo se utilizará el grafo dirigido de la Figura 6.15 como modelo gráfico para una red Bayesiana cuya función de probabilidad conjunta puede factorizarse de la forma

$$p(x) = p(a)p(b)p(c)p(d|a,b,c)p(f)p(g)p(e|d,f,g)p(k)p(l|e,k)p(m)p(n|l,m)p(i)p(q|i,n)$$



**Figura 6.15. Grafo dirigido para el sistema de distribución de energía (motor 2).**



Las funciones de probabilidad condicionada necesarias para definir la función de probabilidad conjunta se dan en la Tabla 6.2. Las probabilidades marginales de los nodos terminales A, B, C, D, F, G, H, I, J, K, L, M, N y Q son

$$\begin{array}{lll}
 p(a)=0.30 & p(b)=0.25 & p(c)=0.40, \\
 p(f)=0.20 & p(g)=0.25 & p(i)=0.35 \\
 p(k)=0.20 & p(m)=0.30 & p(n)=0.45
 \end{array}$$

A	B	C	$p(d a,b,c)$
a	b	c	1
a	b	$\neg c$	0
a	$\neg b$	c	0
a	$\neg b$	$\neg c$	0
$\neg a$	b	c	0
$\neg a$	b	$\neg c$	0
$\neg a$	$\neg b$	c	0
$\neg a$	$\neg b$	$\neg c$	0

B	F	G	$p(h d,f,g)$
b	f	g	1
b	f	$\neg g$	1
b	$\neg f$	g	1
b	$\neg f$	$\neg g$	1
$\neg b$	f	g	1
$\neg b$	f	$\neg g$	1
$\neg b$	$\neg f$	g	1
$\neg b$	$\neg f$	$\neg g$	0

L	M	N	$p(q l,m,n)$
1	m	n	1
1	m	$\neg n$	1
1	$\neg m$	n	1
1	$\neg m$	$\neg n$	1
$\neg 1$	m	n	1
$\neg 1$	m	$\neg n$	1
$\neg 1$	$\neg m$	n	1
$\neg 1$	$\neg m$	$\neg n$	0

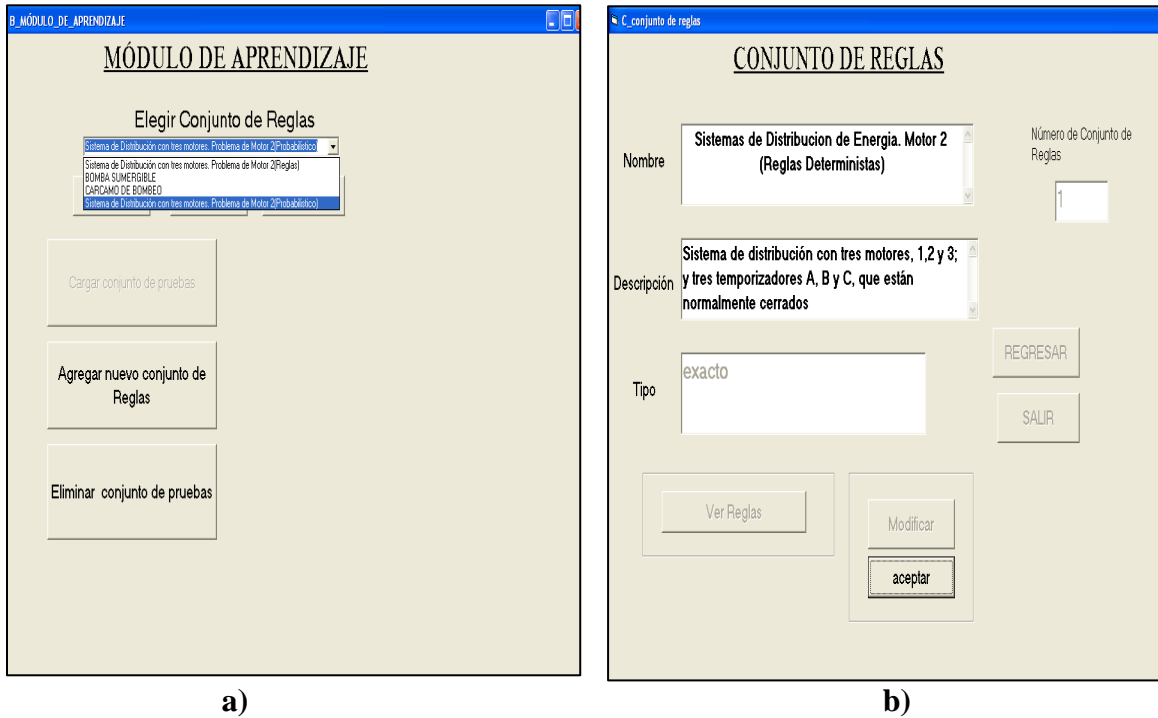
J	K	$p(l j,k)$
j	k	1
j	$\neg k$	0
$\neg j$	k	0
$\neg j$	$\neg k$	0

H	I	$p(j h,i)$
h	i	1
h	$\neg i$	1
$\neg h$	i	1
$\neg h$	$\neg i$	0

**Tabla 6.2. Probabilidades condicionales de fallo de las variables del sistema de distribución de energía (motor 2).**

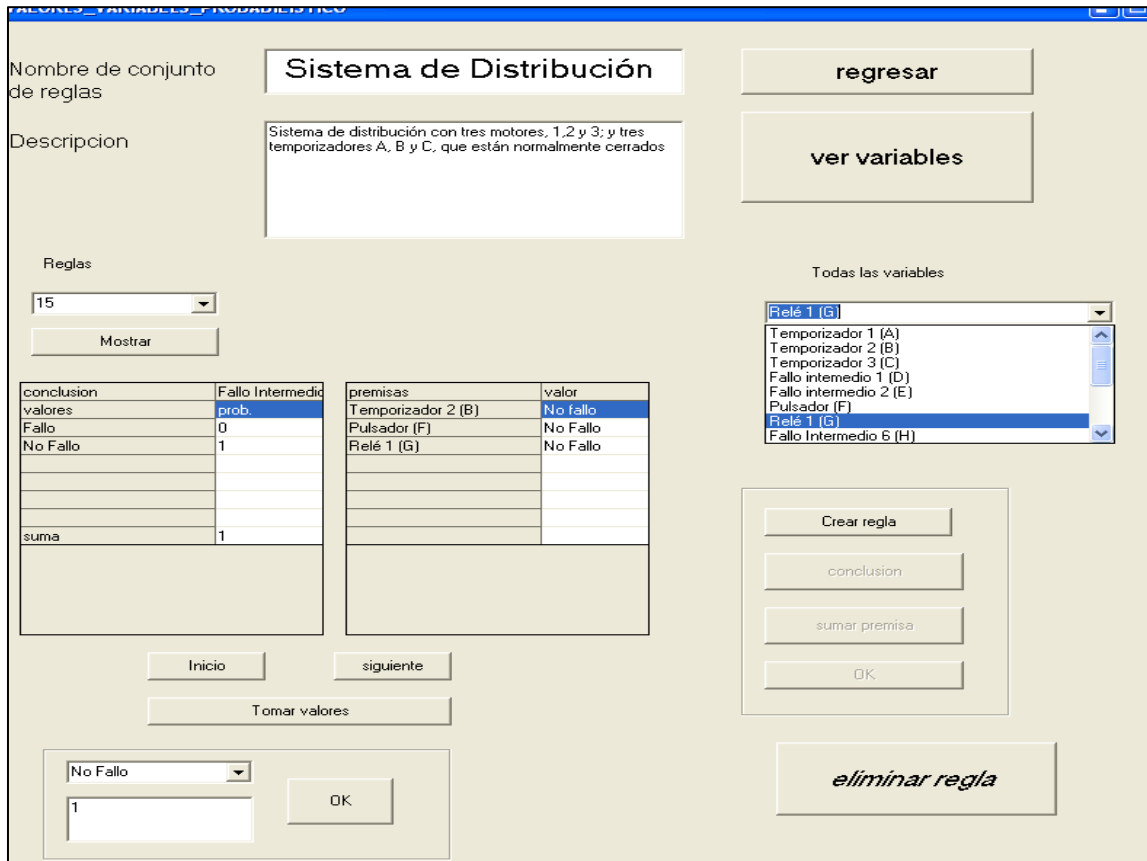
Procedemos de igual forma que el ejemplo anterior. Tenemos un Poliárbol (Figura 6.15), lo que significa que se puede utilizar el algoritmo para poliárboles (Ver Anexo D) para la propagación de evidencia.

Una vez que estamos dentro del Sistema Experto, procedemos de igual forma que el ejemplo anterior, primero debemos de elegir el conjunto de Reglas dentro del módulo de aprendizaje (Figura 6.16a), de aquí el sistema mostrará las características principales de este conjunto (nombre, descripción y tipo) como se muestra en la Figura 6.16b.



**Figura 6.16. Conjunto de Reglas para el sistema de distribución de energía (motor 2).**

El siguiente paso es crear las reglas y asignarle a cada una sus probabilidades marginales y/o condicionadas según sea el caso (Ver Figura 6.17).

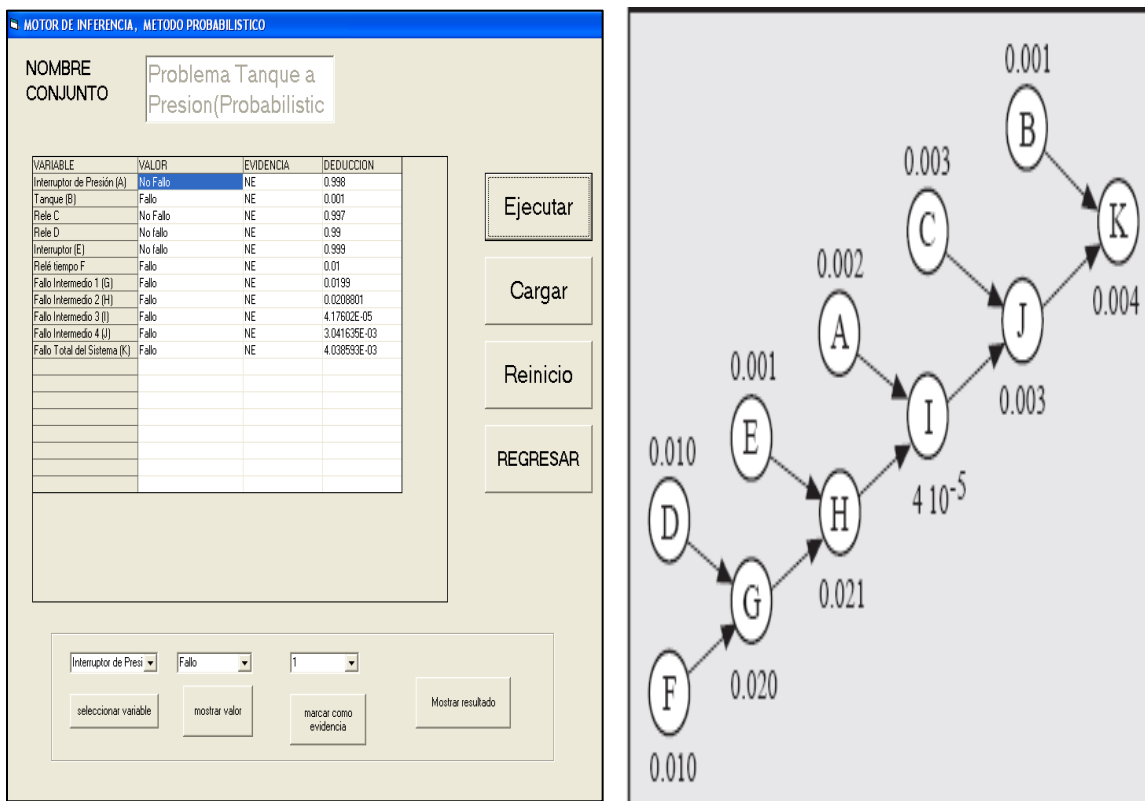


**Figura 6.17. Reglas con sus probabilidades marginales y condicionales pertenecientes al sistema de distribución de energía (motor 2).**

## VII. RESULTADOS

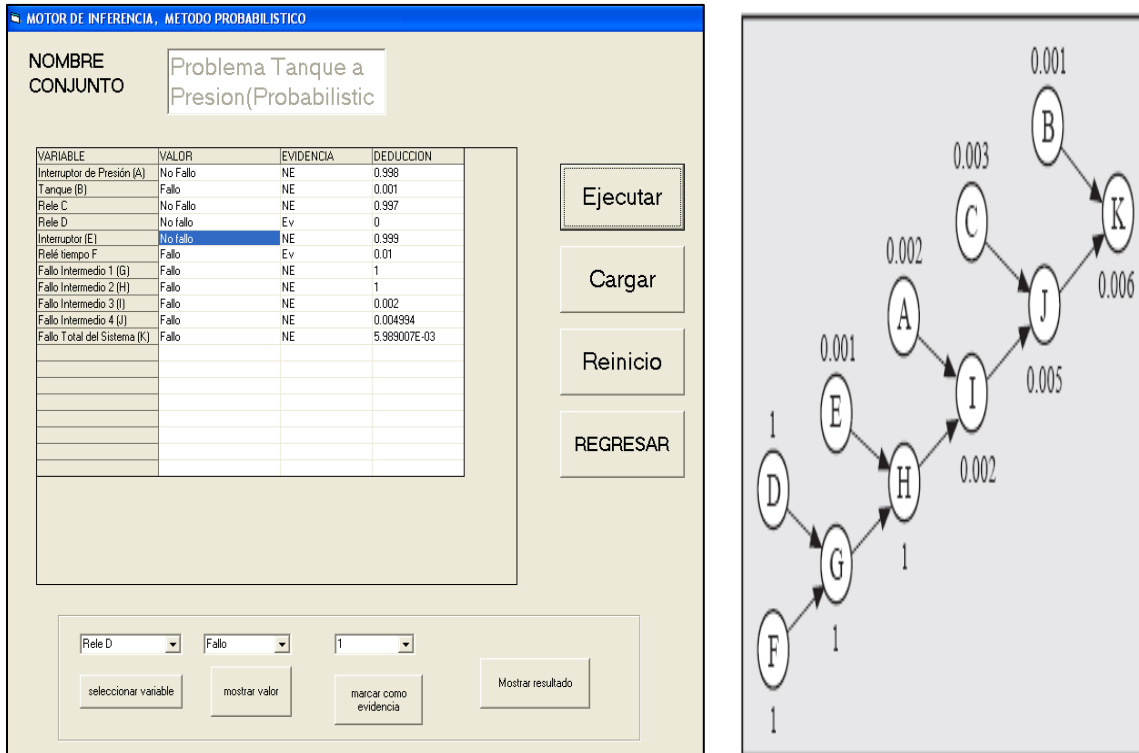
### 7.1 SISTEMA DEL TANQUE A PRESIÓN (CASO1)

El grafo de la Figura 6.4 es un Poliárbol, lo que significa que se puede utilizar el algoritmo para poliárboles (Ver Anexo D) para la propagación de evidencia. Supóngase, en primer lugar, que no hay evidencia disponible. En este caso el Algoritmo da las probabilidades marginales de los nodos y al ejecutarse el motor de inferencia aparecen los resultados que se muestran en la Figura 7.1. Nótese que la probabilidad inicial de fallo del tanque es  $p(k)=0.004$ .



**Figura 7.1. Probabilidades marginales iniciales de los nodos (cuando no hay evidencia) para el tanque a presión.**

Supóngase ahora que las componentes F y D fallan, es decir, se tiene la evidencia  $F=f$  y  $D=d$ . Las probabilidades condicionales de los nodos  $p(x_i|f,d)$  se muestran en la Figura 7.2. Nótese que los fallos de los relés F y D induce el fallo de los nodos intermedios G y H, pero la probabilidad del fallo del tanque es todavía pequeña ( $p(k)=0.006$ ).



**Figura 7.2. Probabilidades condicionales de los nodos dada la evidencia  $F=f$  y  $D=d$  para el sistema del tanque a presión.**

Para continuar ejecutando el Sistema Experto, supóngase ahora que el interruptor de presión A también falla ( $A=a$ ). Si se propaga la evidencia acumulada ( $F=f, D=d, A=a$ ), se obtienen las nuevas probabilidades condicionales de los nodos que se muestran en la Figura 7.3. Ahora, puesto que  $p(k)=1$ , el fallo de estas componentes F, D y A, implican el fallo de todos los nodos intermedios y el fallo del tanque.

MOTOR DE INFERENCIA, METODO PROBABILISTICO

NOMBRE CONJUNTO: Problema Tanque a Presion(Probabilistic)

VARIABLE	VALOR	EVIDENCIA	DEDUCCION
Interruptor de Presión (A)	No Fallo	Ev	0
Tanque (B)	Fallo	NE	0.001
Relé C	No Fallo	NE	0.997
Relé D	No fallo	Ev	0
Interruptor (E)	No fallo	NE	0.999
Relé tiempo F	Fallo	Ev	1
Fallo Intermedio 1 (G)	Fallo	NE	1
Fallo Intermedio 2 (H)	Fallo	NE	1
Fallo Intermedio 3 (I)	Fallo	NE	1
Fallo Intermedio 4 (J)	Fallo	NE	1
Fallo Total del Sistema (K)	Fallo	NE	1

Ejecutar

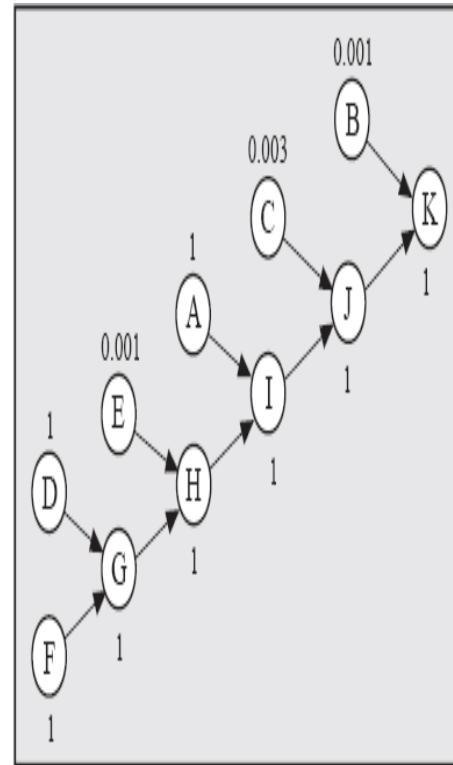
Cargar

Reinicio

REGRESAR

Relé tiempo F:

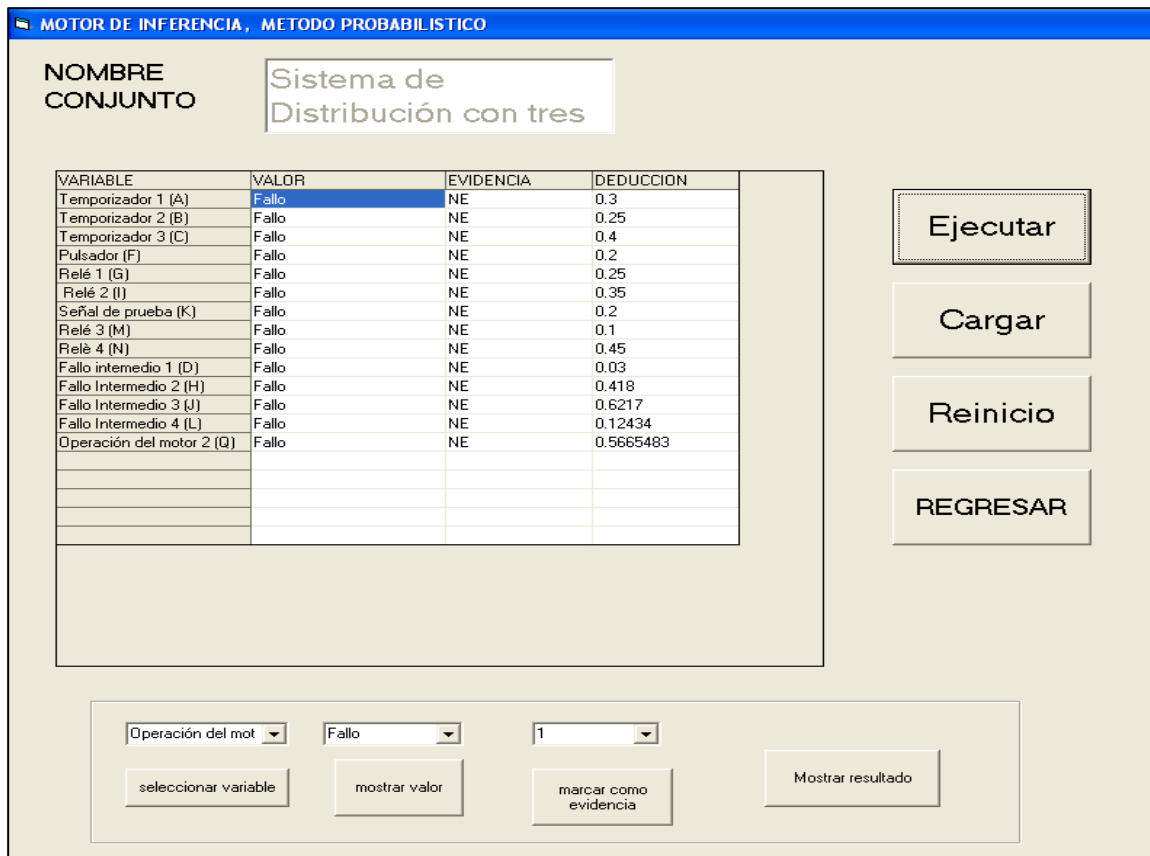
seleccionar variable    mostrar valor    marcar como evidencia    Mostrar resultado



**Figura 7.3. Probabilidades condicionales de los nodos dada la evidencia F=f, D=d y A=a para el sistema del tanque a presión.**

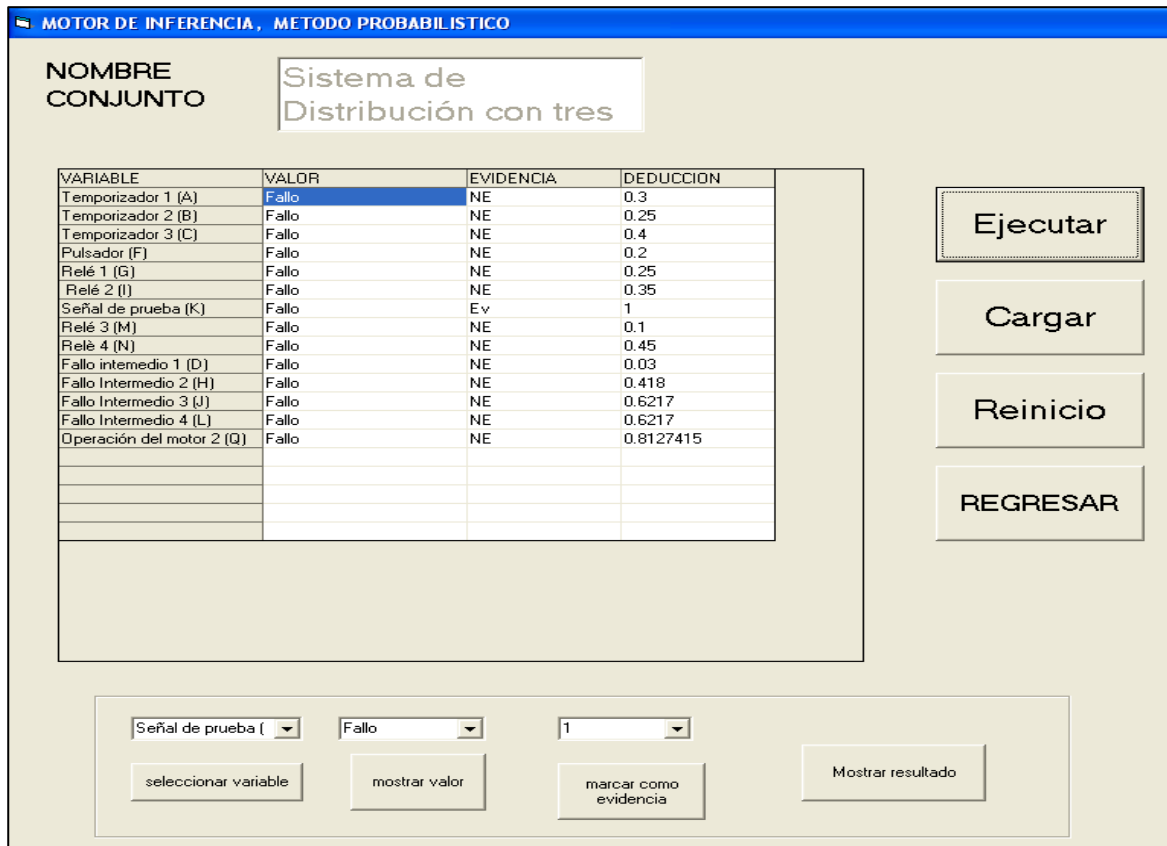
## 7.2 SISTEMA DE DISTRIBUCIÓN DE ENERGÍA (CASO 2)

Supóngase, en primer lugar, que no hay evidencia disponible. En este caso el Algoritmo da las probabilidades marginales de los nodos y al ejecutarse el motor de inferencia aparecen los resultados que se muestran en la Figura 7.4. Nótese que la probabilidad inicial de fallo del motor 2 es  $p(q) = 0.566$ .



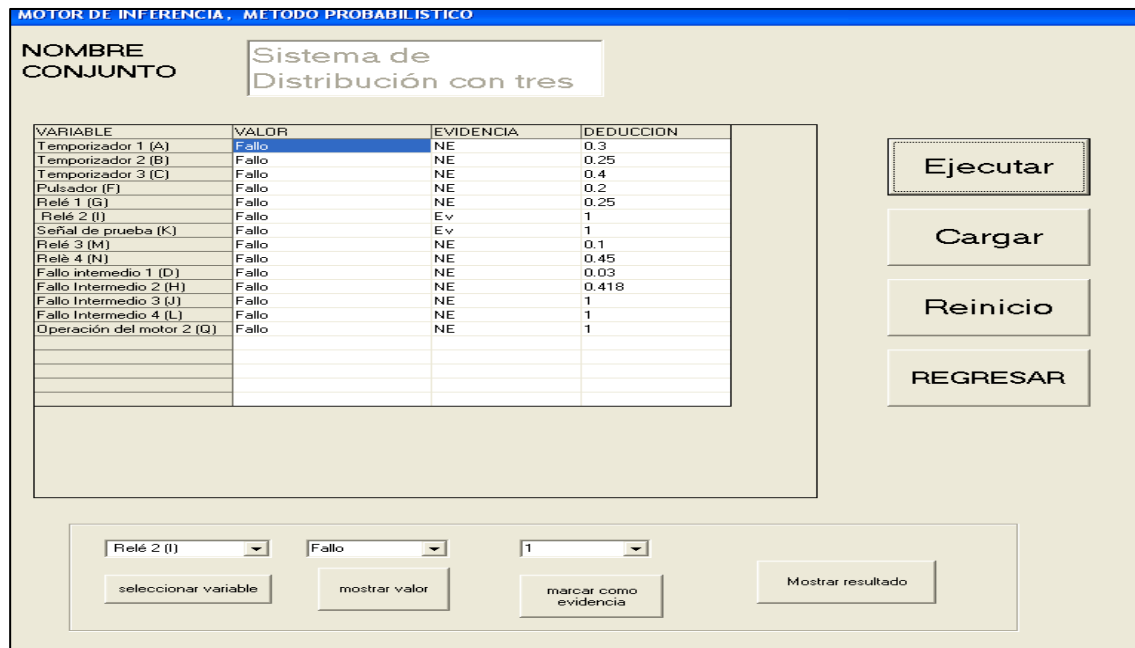
**Figura 7.4. Probabilidades marginales de los nodos cuando no hay evidencia para el sistema de distribución de energía (motor 2).**

Supóngase ahora que se tiene la evidencia  $K=k$ . las probabilidades condicionales de los nodos dada esta evidencia se muestran en la Figura 7.5. En este caso la probabilidad de fallo aumenta pasando del valor inicial  $p(q) = 0.566$  al valor  $p(q|K=k) = 0.8127$ .



**Figura 7.5. Probabilidades condicionales de los nodos dada la evidencia  $K=k$ , para el sistema de distribución de energía (motor 2).**

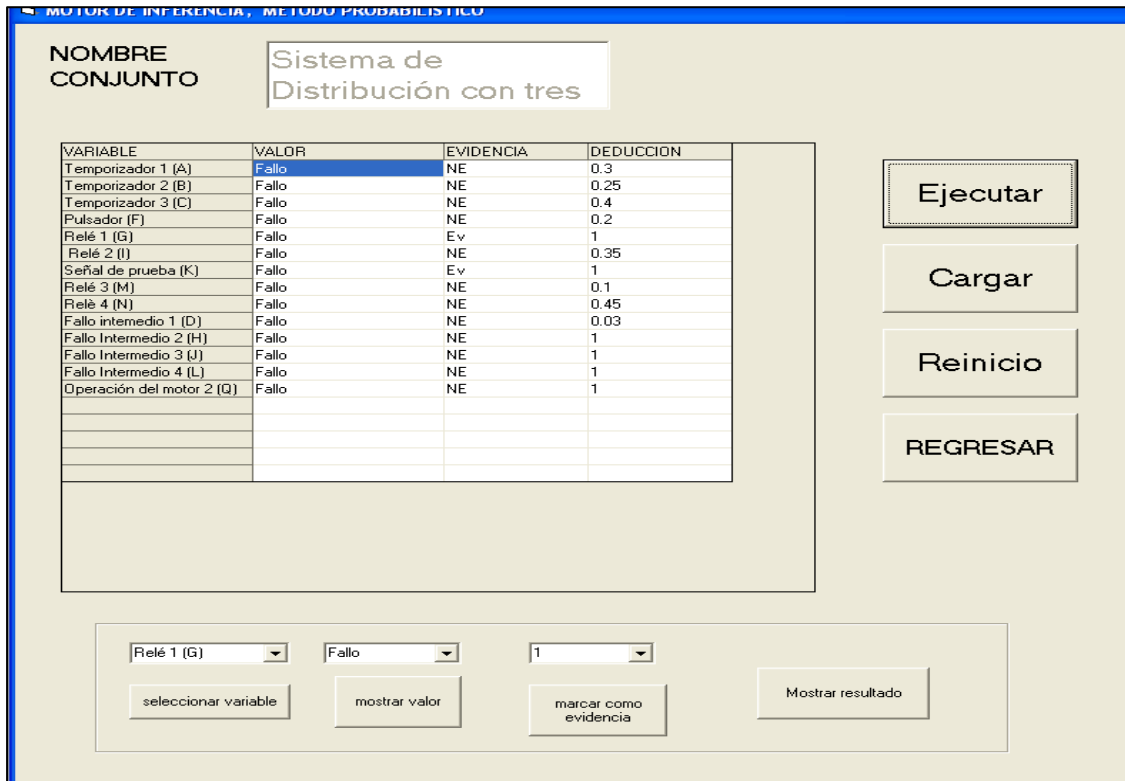
Quando se introduce la evidencia adicional  $I=i$ , entonces J y L se presentan. Consecuentemente, el sistema falla;  $p(q|I=i, K=k) = 1$  (Ver Figura 7.6).



**Figura 7.6. Probabilidades condicionales de los nodos dada la evidencia  $K=k$  e  $I=i$ , para el sistema de distribución de energía (motor 2).**

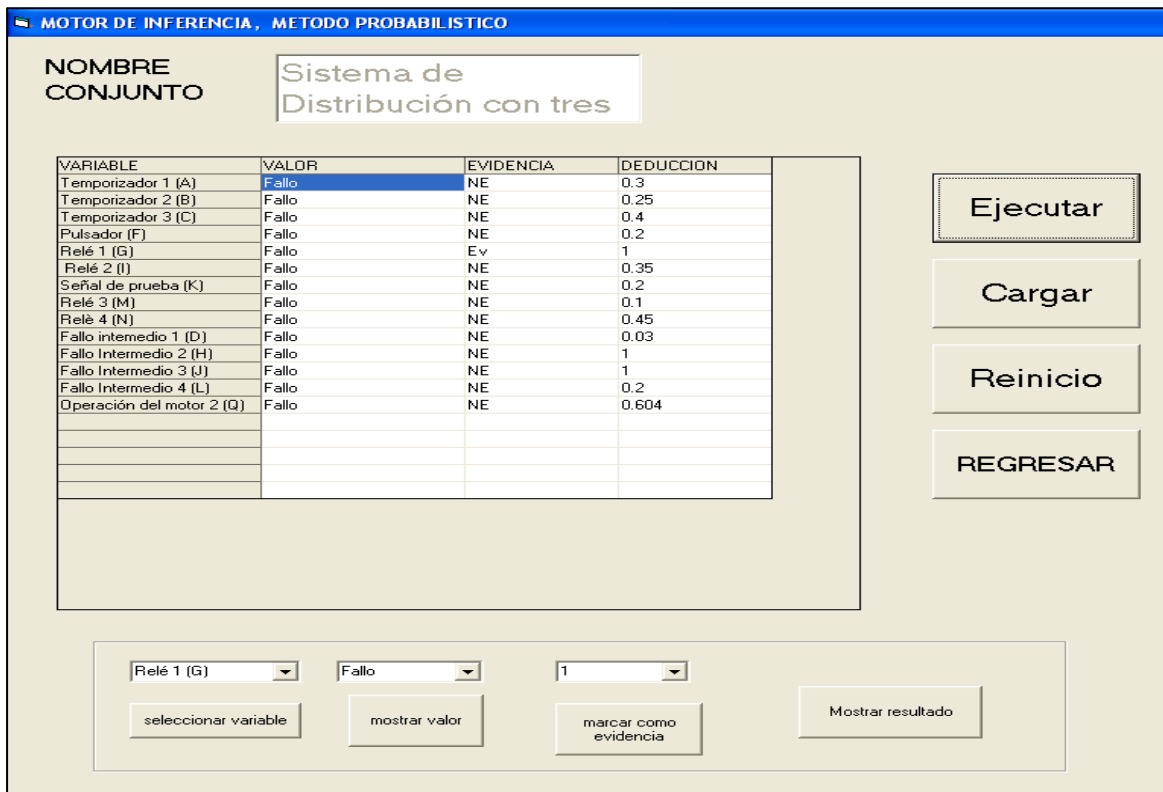


Otros ejemplos de fallos dados distintas evidencias se muestran en las siguientes figuras: Figura 7.4 evidencia  $K=k$  y  $G=g$ , Figura 7.7 evidencia de que Relé G falla ( $G=g$ ) y por último; la Figura 7.6 evidencia de que Relé I falla ( $I=i$ ).

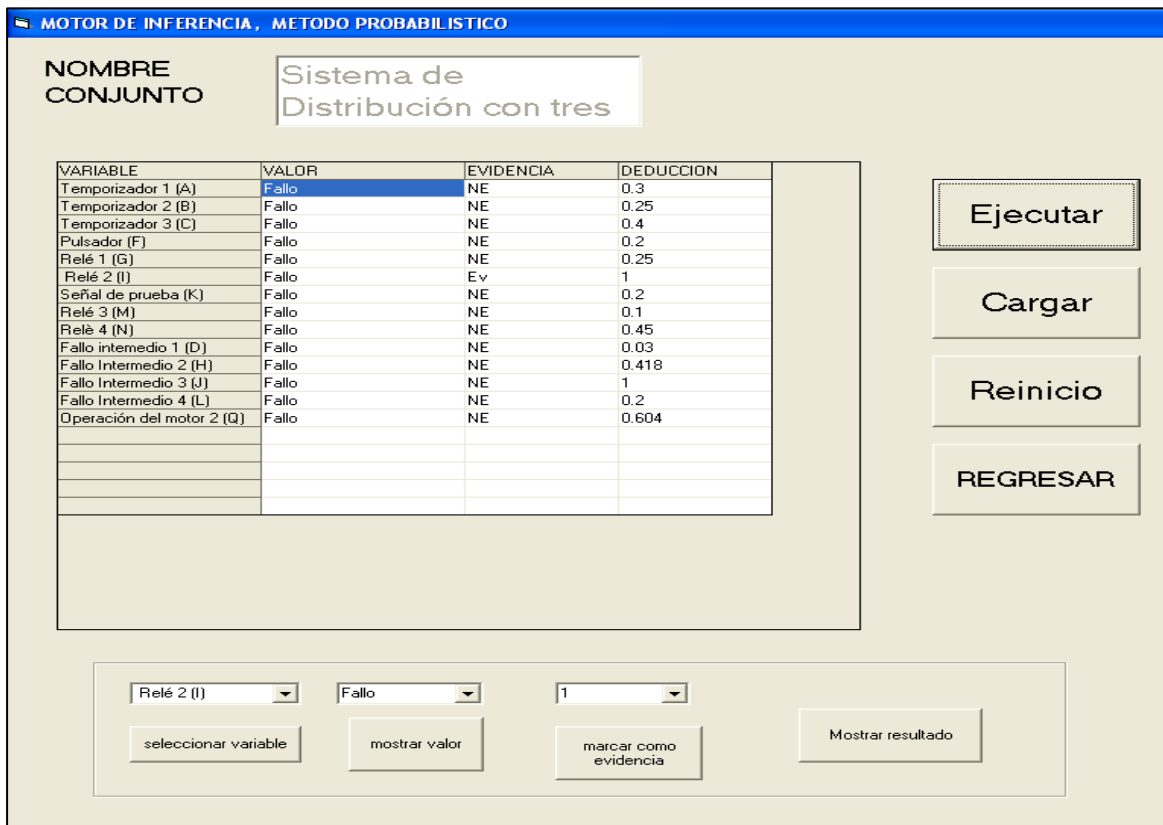


**Figura 7.7. Probabilidades condicionales de los nodos dada la evidencia  $K=k$  e  $G=g$ , para el sistema de distribución de energía (motor 2).**

La probabilidad de fallo del motor 2 ( $q=0.604$ ) es igual para el caso de que falle el Relé G ó el Relé I. Véase Figuras 7.8 y 7.9.



**Figura 7.8. Probabilidades condicionales de los nodos dada la evidencia  $G=g$ , para el sistema de distribución de energía (motor 2).**



**Figura 7.9. Probabilidades condicionales de los nodos dada la evidencia  $I=i$ , para el sistema de distribución de energía (motor 2).**

# CONCLUSIONES

En base al resultado de la presente investigación, y considerando la experiencia obtenida en el trabajo, se puede concluir primeramente que; para el caso de la programación del Sistema Experto por Método Determinístico (Basado en Reglas), éste representó un apoyo importante para entender de una manera más sencilla el proceso de Encadenamiento de Reglas. Es decir que, se puede ver como un método de programación didáctico, ya que se realizó con el fin de probar el Sistema Experto, encontrar y solucionar fallas dentro de éste. El Método Basado en Reglas forma parte dentro de la programación del Método de Explicaciones.

Por otra parte, dentro de la aportación más importante de presente trabajo, se encuentra la programación del Sistema Experto por el Método Probabilístico; ya que éste método muestra un algoritmo más exacto y aplicable a la solución de problemas comunes que se pueden presentar en un área de mantenimiento específica. De acuerdo a éste método de programación, en base a las probabilidades que introduce y actualiza el usuario, le permitirá darse una idea más clara de cuando un elemento en particular dentro del sistema se encuentra próximo a presentar una falla, gracias a esto el usuario podrá realizar una acción de mantenimiento adecuada para así evitar un fallo general en todo el sistema o proceso.

Dentro del trabajo futuro para la presente investigación se encuentra la tarea de programación del algoritmo encargado del proceso de explicación para el Método Probabilístico; dicho algoritmo deberá de cumplir con la tarea de proporcionar al usuario la explicación correcta de por qué un elemento presentó fallo y de acuerdo a esto último, deberá de presentar una o más recomendaciones así como las acciones a seguir para evitar un fallo o paro total en el sistema o proceso. Todo esto deberá realizarse en base a las probabilidades condicionales obtenidas por el Sistema Experto.

Parte del trabajo futuro importante, se encuentra la programación de problemas que requieran una solución mediante el Método de Lógica Difusa y revisar los resultados obtenidos por el SE; el algoritmo para dicho método ya se encuentra

programado en el Sistema Experto realizado. Al igual que para el Método Probabilístico, faltará programar el algoritmo de explicación y recomendación para Lógica Difusa.

Los sistemas expertos son de mucha utilidad en la vida real, y apoyan en gran manera a los sistemas de soporte a la decisión, ya que nos permiten realizar decisiones basadas en la experiencia humana de algún especialista en determinada área, esto es con el fin de retener el conocimiento y de esa manera lograr convertirlo en un activo importante de una organización y que se traduce en un valor importante para la misma, pues con ese tipo de sistemas, nos permiten contar con la experiencia primordial, aunque sea de manera virtual.

El periodo de realización de un Sistema Experto es largo, no por el desarrollo de la aplicación, sino del proceso de adquisición de conocimientos, ya que el conocimiento a adquirir es un conocimiento especializado, con el cual el ingeniero del conocimiento no se encuentra familiarizado.

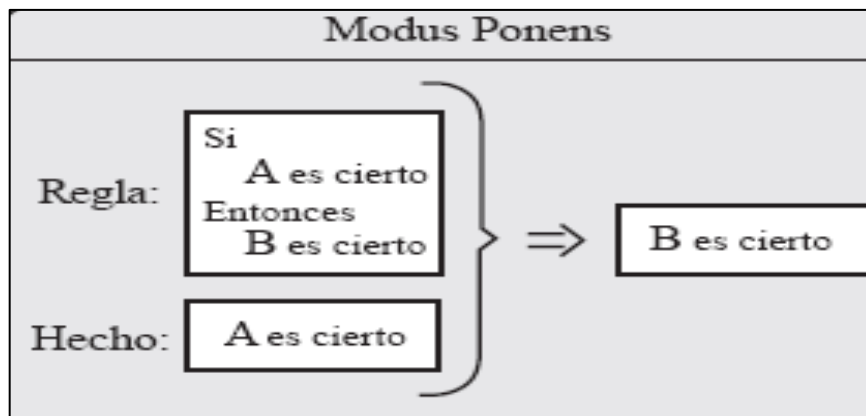
No se puede considerar este trabajo como una investigación completa y absoluta de los Sistemas Expertos. Hace falta mucho más por investigar y profundizar, como ejemplo están los lenguajes de programación de IA para construcción de Sistemas Expertos.

Para finalizar, cada día aparecen nuevas aplicaciones para los Sistemas Expertos, por lo cual deben haber futuras investigaciones acerca de las aplicaciones de los SE, con el fin de actualizar la información que aquí se ha presentado. Ojala que en un futuro se realice una investigación que muestre una aceptación mayor hacia los Sistemas Expertos en el área de mantenimiento dentro de las Empresas Mexicanas.

## Apéndice A Motor de Inferencia para Sistema Experto Determinístico.

### *Modus Ponens y Modus Tollens*

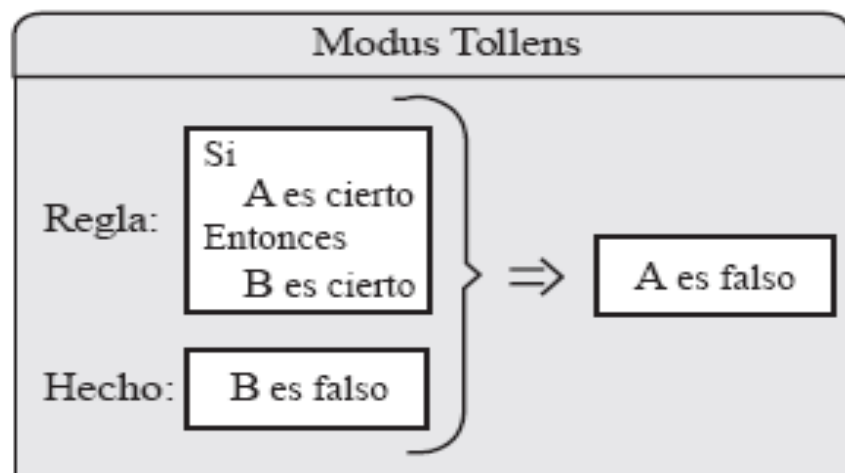
El *Modus Ponens* es quizás la regla de inferencia mas comúnmente utilizada. En ella, se examina la premisa de la regla, y si es cierta, la conclusión pasa a formar parte del conocimiento. Como ilustración, supóngase que se tiene la regla, “Si *A* es cierto, entonces *B* es cierto” y que se sabe además que “*A* es cierto.” Entonces, tal como muestra la Figura 1A, 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.



**Figura 1A. Una ilustración de la regla de inferencia Modus Ponens.**

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, tal como se muestra en la Figura 2A, la regla Modus Tollens concluye que “*A* es falso.” Aunque muy simple y con muchas aplicaciones útiles, la regla Modus Tollens es menos utilizada que la Modus Ponens.

Por ello, la regla Modus Ponens se mueve hacia adelante, es decir, de la premisa a la conclusión de una regla, mientras que la regla Modus Tollens se mueve hacia atrás, es decir, de la conclusión a la premisa. Las dos reglas de inferencia no deben ser vistas como alternativas sino como complementarias. La regla Modus Ponens necesita información de los objetos de la premisa para concluir, mientras que la regla Modus Tollens necesita información sobre los objetos de la conclusión. De hecho, para un motor de inferencia que solamente utiliza Modus Ponens, la incorporación de la regla de inferencia Modus Tollens puede ser considerada como una expansión de la base de conocimiento mediante la adición de reglas, tal como ilustra el ejemplo que sigue.



**Figura 2A. Una ilustración de la regla Modus Tollens.**

### *Encadenamiento de Reglas*

Una de las estrategias de inferencia más utilizadas para obtener conclusiones es llamada *encadenamiento de reglas*. Esta estrategia puede utilizarse cuando las premisas de ciertas reglas coinciden con las conclusiones de otras. Cuando se encadenan las reglas, los hechos pueden utilizarse para dar lugar a nuevos hechos. Esto se repite sucesivamente hasta que no pueden obtenerse más conclusiones. El tiempo que consume este proceso hasta su terminación depende, por una parte, de los hechos conocidos, y por otra, de las reglas que se activan. La estrategia de encadenamiento de reglas se da en el algoritmo siguiente:

### **Algoritmo de encadenamiento de reglas.**

- **Datos:** Una base de conocimiento (objetos y reglas) y algunos hechos iniciales.
- **Resultado:** El conjunto de hechos derivados lógicamente de ellos.
  1. Asignar a los objetos sus valores conocidos tales como los dan los hechos conocidos o la evidencia.
  2. Ejecutar cada regla de la base de conocimientos y concluir nuevos hechos si es posible.
  3. Repetir el punto 2 hasta que no puedan ser obtenidos nuevos hechos.

Este algoritmo puede ser implementado de muchas formas. Una de ellas comienza con las reglas cuyas premisas tienen valores conocidos. Estas reglas deben concluir y sus conclusiones dan lugar a nuevos hechos. Estos nuevos hechos se añaden al conjunto de hechos conocidos, y el proceso continúa hasta que no pueden obtenerse nuevos hechos.

## APENDICE B MOTOR DE INFERENCIA POR LÒGICA DIFUSA

La lógica difusa o borrosa (Fuzzy logic) descansa en la idea que en un instante dado, no es posible precisar el valor de una variable  $X$ , sino tan solo conocer el grado de pertenencia a cada uno de los conjuntos en que se ha participado el rango de variación de la variable.

El grado de pertenencia se cuantifica mediante la función de pertenencia  $f$ , que normalmente se escoge de una forma trapezoide.

TB: Temperatura.

TM: Temperatura media.

TA: Temperatura alta.

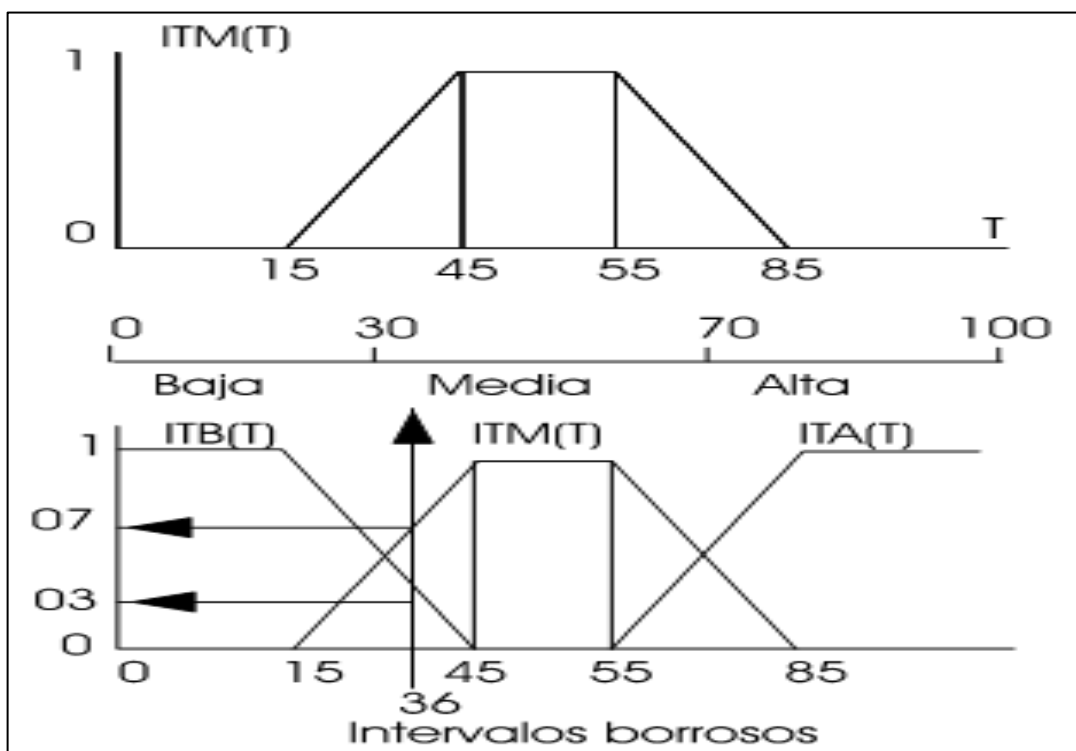


Figura 1B. Ejemplo de funciones de pertenencia

Si  $f_A(x)$  indica la función de pertenencia de  $x$  al conjunto  $A$ , entonces



$f_A(x)$  esta entre 0 y 1

si  $f_A(x)=1$ ,  $x$  pertenece totalmente a  $A$

si  $f_A(x)=0$ ,  $x$  no pertenece a  $A$

A partir de esta definición es posible comprobar que se cumplen las siguientes propiedades:

$f_{A \text{ or } B}(x) = \max(f_A(x), f_B(x))$

$f_{A \text{ and } B}(x) = \min(f_A(x), f_B(x))$

$f_{\text{not } A}(x) = 1 - f_A(x)$

### **Fuentes de incertidumbre**

- Confiabilidad de la información
- Difusividad
- Aleatoriedad
- Imprecisión del lenguaje de representación mediante reglas lingüísticas
- Información incompleta
- Información agregada
- Precisión de la representación
- Declaración en conflicto
- Reglas de combinación evidentes

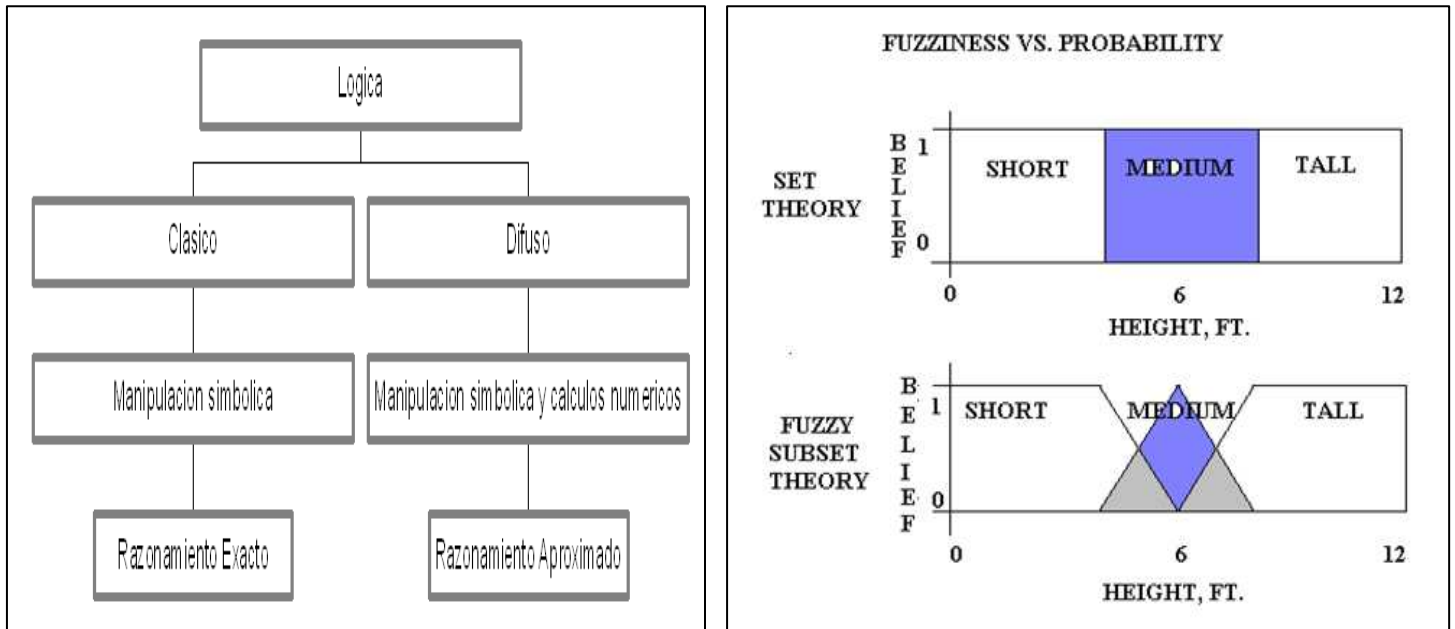
Difusividad esta relacionada al grado con el cual los eventos ocurren sin importar la probabilidad de su ocurrencia.

Por ejemplo, el grado de juventud de una persona es un evento difuso sin importar que sea un elemento aleatorio.

### **Difusividad contra Probabilidad.**

- Difusividad es una incertidumbre determinística, la probabilidad es no determinística.

- La incertidumbre probabilística se disipa con el incremento del número de ocurrencias y la difusividad no.
- La difusividad describe eventos ambiguos, La probabilidad describe los eventos que ocurren. Si un evento ocurre es aleatorio. El grado con el cual ocurre es difuso.



**Figura 2B. Comparación entre Lógica Difusa y Lógica Clásica.**

**Para qué es usada la lógica difusa.**

Comúnmente se usa para toma de decisiones en presencia de datos o conocimientos inciertos.

Reconocimiento de patrones ambiguos.

Como un componente de sistemas expertos difusos

**Para que se usan los sistemas expertos difusos.**

- En control de sistemas lineales y no lineales.
- Para modelar sistemas lineales y no lineales.

- En diagnóstico y aislamiento de fallas en sistemas en tiempo real y análisis de datos fuera de línea.
- Para tomar una decisión completa.

### **Ventajas de los sistemas expertos difusos (SED)**

- Son una forma fácil de codificar un sistema no lineal.
- Tiene una buena correspondencia a la forma del pensamiento humano sobre una gran clase de problemas matemáticos.
- Los sistemas expertos difusos se ejecutan rápidamente sobre computadoras convencionales.
- Los sistemas expertos difusos los ejecutan a velocidades extremadamente altas sobre hardware especializado.

## APENDICE C MOTOR DE INFERENCIA POR MÈTODO DE CERTEZA

Los sistemas expertos implementan el formalismo de Factor de Certeza (FC) por medio de reglas:

SI evidencia

ENTONCES hipótesis (FC)

donde evidencia es uno o más hechos conocidos para soportar la derivación de la hipótesis. El valor de FC denota la creencia en la hipótesis, dada que la evidencia es observada. El formalismo FC usa tres valores distintos:

- 1) La medida de credibilidad  $MC(h,e)$ , la cual es un valor entre 0 y 1 que representa el grado de creencia en que la hipótesis  $h$  es soportada por la evidencia  $e$ .
- 2) La medida de incredibilidad  $MI(h,e)$ , la cual es un valor entre 0 y 1 que representa el grado de no creencia de que la hipótesis  $h$  es soportada por la evidencia  $e$ .
- 3) El factor de certeza FC, el cual es una combinación de MC y MI, su rango de variación fluctúa entre -1 (negación de  $h$ ) y +1 (confirmación de  $h$ ). Se define así:

$$FC = \frac{MC - MI}{1 - \min(MC - MI)}$$

Durante el proceso de inferencia, muchas reglas pueden derivar la misma hipótesis o conclusión. Como resultado debe existir un mecanismo para combinar los FC de cada regla y obtener un solo FC para esa hipótesis. Se usan las siguientes ecuaciones:

$$\begin{aligned}
FC_{rev}(FC_{old}, FC_{new}) &= FC_{old} + FC_{new} (1 - FC_{old}) \text{ si } FC_{old} \text{ y } FC_{new} > 0 \\
&= -FC_{rev} (-FC_{old}, -FC_{new}) \text{ si } FC_{old} \text{ y } FC_{new} < 0 \\
&= (FC_{old} + FC_{new}) / (1 - \min(|FC_{old}|, |FC_{new}|)) \\
&\quad \text{si uno de } FC_{old} \text{ y } FC_{new} < 0
\end{aligned}$$

## APENDICE D: MOTOR DE INFERENCIA PARA SISTEMA EXPERTO PROBABILISTICO, METODO EXACTO

El mecanismo para obtener conclusiones a partir de la evidencia se conoce como *propagación de evidencia*. Esta tarea consiste en actualizar las probabilidades de las variables en función de la evidencia.

### *Propagación de Evidencia (Poliárboles)*

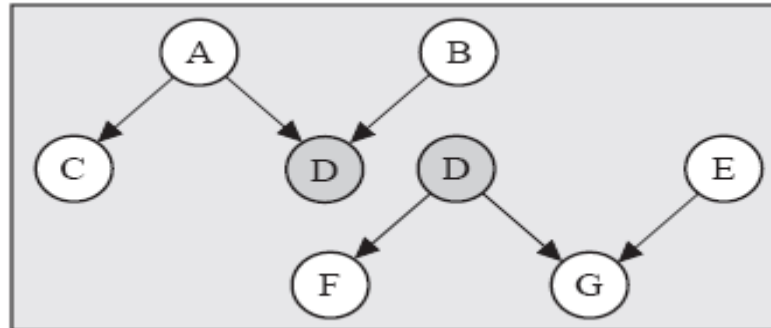
La propagación de evidencia es una de las tareas más importantes de un sistema experto, pues permite obtener conclusiones cuando se dispone de nueva información. Supóngase un conjunto de variables discretas  $X = \{X_1, \dots, X_n\}$  y una función de probabilidad  $p(x)$ , en  $X$ . Cuando no se dispone de ninguna información, es decir, cuando no existe evidencia, el proceso de propagación consiste en calcular las probabilidades marginales  $p(X_i = x_i)$ , también denotadas por  $p(x_i)$ , para cada  $X_i$ . Estas probabilidades proporcionan información “a priori” sobre los distintos valores que pueden tomar las variables. Cuando se dispone de cierta evidencia, es decir, cuando se conoce un conjunto de variables que tienen asociadas los valores  $X_i = e_i$ , el proceso de propagación debe tener en cuenta estos valores para calcular las nuevas probabilidades de los nodos.

Un algoritmo de propagación se denomina exacto si calcula las probabilidades de los nodos sin otro error que el resultante del redondeo producido por las limitaciones de cálculo de la computadora.

En el siguiente ejemplo se explica como se trabaja el motor de inferencia para un grafo en un sistema probabilístico.

En un poliárbol dos nodos cualesquiera están unidos por un único camino, lo cual implica que cada nodo divide al poliárbol en dos poliárboles inconexos: uno que contiene a sus padres y a los nodos a los que está conectado a pasando por sus padres, y otro que incluye sus hijos y a los nodos a los que está conectado pasando por sus hijos. Por ejemplo, el nodo  $D$  divide al poliárbol de la Figura 1D en dos poliárboles

inconexos, el primero de los cuales,  $\{A,B,C\}$ , incluye a sus padres y a los nodos que son accesibles desde  $D$  a través de sus padres, y el segundo,  $\{E,F,G\}$ , que incluye a sus hijos y a los nodos que son accesibles desde  $D$  a través de sus hijos.



**Figura 1D. Ejemplo de grafo dirigido o Poliárbol.**

El proceso de propagación puede realizarse en este tipo de grafos de un modo eficiente combinando la información procedente de los distintos poliárboles mediante el envío de mensajes (cálculos locales) de un poliárbol a otro.

La formula para obtener la probabilidad de cierto nodo  $X_i$  dada la evidencia es:

$$p(\mathbf{x}_i|\mathbf{e}) = p(\mathbf{x}_i|\mathbf{e}_i^-, \mathbf{e}_i^+) = \frac{1}{p(\mathbf{e}_i^-, \mathbf{e}_i^+)} p(\mathbf{e}_i^-, \mathbf{e}_i^+ | \mathbf{x}_i) p(\mathbf{x}_i)$$

Tratando la ecuación se llega a:

$$\begin{aligned}
 p(\mathbf{x}_i|\mathbf{e}) &= \frac{1}{p(\mathbf{e}_i^-, \mathbf{e}_i^+)} p(\mathbf{e}_i^-|\mathbf{x}_i) p(\mathbf{e}_i^+|\mathbf{x}_i) p(\mathbf{x}_i) \\
 &= \frac{1}{p(\mathbf{e}_i^-, \mathbf{e}_i^+)} p(\mathbf{e}_i^-|\mathbf{x}_i) p(\mathbf{x}_i, \mathbf{e}_i^+) \\
 &= k p(\mathbf{e}_i^-|\mathbf{x}_i) p(\mathbf{x}_i, \mathbf{e}_i^+) \\
 &= k\lambda_i(\mathbf{x}_i)\rho_i(\mathbf{x}_i)
 \end{aligned}$$

Luego el algoritmo para calcular la evidencia es:

- ✦ Datos: Una red Bayesiana (D,P) definida sobre un conjunto de variables X a partir de un polígrafo D y un conjunto de nodos evidenciales E que toman los valores  $E = e$ .
- ✦ Resultados: Las funciones de probabilidad condicionada  $p(x_i|e)$  para cada nodo  $X_i$ .

### Etapa de Iniciación:

1. Asignar a todos los nodos evidenciales, las funciones:

$$\text{✦ } \rho_i(x_i) = 1 \text{ si } x_i = e_i, \text{ o } \rho_i(x_i) = 0 \text{ si } x_i \neq e_i.$$

$$\text{✦ } \lambda_i(x_i) = 1 \text{ si } x_i = e_i, \text{ o } \lambda_i(x_i) = 0 \text{ si } x_i \neq e_i.$$

2. Asignar a todos los nodos que no pertenezcan a la evidencia que no tengan padres la función:

$$\rho_i(x_i) = p(x_i).$$

3. Asignar a todos los nodos que no pertenezcan a la evidencia que no tengan hijos la función:

$$\lambda_i(x_i) = 1, \text{ para todo } x_i.$$

### Etapa Iterativa:

4. Para cada nodo  $X_i$  que no forma parte de los nodos evidenciales:

(a) Si  $X_i$  ha recibido los mensajes  $\rho$  de todos sus padres, calcular  $\rho_i(x_i)$ .

(b) Si  $X_i$  ha recibido los mensajes  $\lambda$  de todos sus hijos, calcular  $\lambda_i(x_i)$ .

(c) Si ya se ha calculado  $\rho_i(x_i)$ , entonces, para cada hijo  $Y_j$  de  $X_i$  tal que  $X_i$  haya recibido los mensajes  $\lambda$  del resto de sus hijos, calcular y enviar el mensaje  $\rho_{X_i Y_j}(x_i)$ . Por tanto, si  $X_i$  ha recibido los mensajes  $\lambda$  de todos sus hijos, entonces, puede ya enviar todos los mensajes  $\rho$  correspondientes.

(d) Si ya se ha calculado  $\lambda_i(x_i)$ , entonces, para cada padre  $U_j$  de  $X_i$  tal que  $X_i$  haya recibido los mensajes  $\rho$  del resto de sus padres, calcular y enviar el mensaje  $\lambda_{U_j X_i}(u_j)$ . Análogamente al caso anterior, si  $X_i$  ha recibido los mensajes  $\rho$  de todos sus padres, entonces, ya puede enviar todos los mensajes  $\lambda$  correspondientes.



5. Repetir el Paso 4 tantas veces como sea necesario hasta que se calculen las funciones  $\rho$  y  $\lambda$  de todos los nodos, es decir, hasta que no se produzca ningún nuevo mensaje en una iteración completa.
6. Para cada nodo, calcular  $\beta_i(x_i)$  .Estas son las probabilidades no normalizadas correspondientes a  $p(x_i|e)$ .
7. Para cada nodo calcular  $p(x_i|e)$  normalizando la función  $\beta_i(x_i)$ .

Este método de propagación en poliárboles es válido solamente para redes de estructura simple, en las cuales existe un único camino entre cada par de nodos. Para redes múltiplemente conexas se pueden utilizar dos métodos de propagación derivados del anterior: método de condicionamiento y método de agrupamiento. La idea fundamental del método de propagación por condicionamiento es cortar los múltiples caminos entre los nodos mediante la asignación de valores a un conjunto reducido de variables contenidas en los bucles. De esta forma se tendría un poliárbol en el cual se podrá aplicar el algoritmo de propagación descrito. Por otra parte, el método de agrupamiento construye representaciones auxiliares, de estructura más simple, uniendo conjuntos de nodos del grafo original.

**APENDICE E: MOTOR DE INFERENCIA PARA SISTEMA EXPERTO  
PROBABILISTICO, METODO ESTOCASTICO**

Para este motor de inferencia se sugiere un método de simulación que genera las variables una a una, con un muestreo hacia delante, es decir, se muestrea una variable sólo cuando ya han sido muestreados todos sus padres. Según este método, se simulan todas las variables, incluyendo las evidenciales, en caso de que las haya. La distribución con la que se simula  $X_i$  es su función de probabilidad condicionada, es decir:

$$h(x_i|\pi_i) = h(x_i|\pi_i), i \in \{1, \dots, n\}$$

Por ello, las variables deben ordenarse de tal forma que los padres de una variable deben preceder a ésta en la simulación. Una ordenación de los nodos que satisface tal propiedad se llama *ordenación ancestral*. Esta estrategia de simulación se llama muestreo *hacia delante* por que va de padres a hijos. Una vez simulado los padres de  $X_i$  y asignados sus valores, se simula un valor de  $X_i$  usando la distribución de simulación  $h(x_i|\pi_i)$ , que en este caso es  $p(x_i|\pi_i)$ . Por lo tanto, los pesos se obtienen mediante

$$s(\mathbf{x}) = \frac{p_e(\mathbf{x})}{h(\mathbf{x})} = \frac{\prod_{X_i \notin E} p_e(x_i | \pi_i) \prod_{X_i \in E} p_e(x_i | \pi_i)}{\prod_{X_i \notin E} p(x_i | \pi_i) \prod_{X_i \in E} p(x_i | \pi_i)}$$

$$s(\mathbf{x}) = \begin{cases} 1, & \text{si } x_i = e_i, \text{ para todo } \mathbf{X}_i \in E \\ 0, & \text{en otro caso.} \end{cases}$$

Nótese que si  $x_i \neq e_i$  para algún  $X_i \in E$ , entonces el peso es cero; por tanto, tan pronto como el valor simulado para los nodos evidenciales no coincida con el valor observado, se rechaza la muestra (peso cero). El proceso de simulación procede secuencialmente, variable a variable.

Consecuentemente, no es posible tener en cuenta la evidencia de que se dispone hasta que las variables correspondientes han sido muestreadas. Una vez obtenidos los valores simulados de las variables evidenciales, si dichos valores coinciden con la evidencia, la muestra se da por válida; en otro caso, se rechaza. Por tanto, las probabilidades condicionales se aproximan calculando el cociente entre los casos que están en concordancia con la evidencia y el número de casos totales.

En algunos casos, este método conduce a un porcentaje de rechazo muy alto y puede requerir un altísimo número de simulaciones, especialmente en los casos en los que las probabilidades de la evidencia son pequeñas, es decir, en redes con probabilidades extremas.

A continuación se muestra el Pseudocódigo para el método del muestreo de aceptación-rechazo.

```
Iniciar
  Ordenar los nodos ancestralmente
Ciclo Principal
  para  $j \leftarrow 1$  a  $N$  hacer
    para  $i \leftarrow 1$  a  $n$  hacer
       $x_i \leftarrow$  generar un valor a partir de  $p(x_i|\pi_i)$ 
      si  $X_i \in E$  y  $x_i \neq e_i$  entonces, repetir el ciclo  $i$ 
```

**Figura 1E. Pseudocódigo método aceptación-rechazo.**

## EJEMPLO DE MUESTREO POR EL MÉTODO DE ACEPTACIÓN-RECHAZO

Considérese la red con seis nodos de la Figura 2E y las correspondientes funciones de probabilidad condicional de la Tabla E1. La función de probabilidad conjunta de las seis variables puede factorizarse como:

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1) p(x_3|x_1) p(x_4|x_2) p(x_5|x_2, x_3) p(x_6|x_3)$$

Dada la evidencia  $X_3=1$ ,  $X_4=1$  (véase la figura 1), se desea calcular la función de probabilidad “a posteriori” de cada una de las otras cuatro variables aplicando el método de aceptación-rechazo.

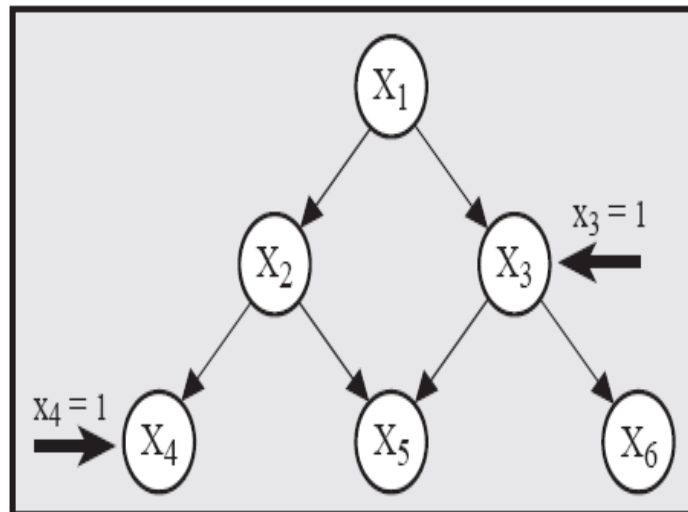


Figura 2E. Grafo dirigido.

$X_1$	$p(X_1)$
0	0.3
1	0.7

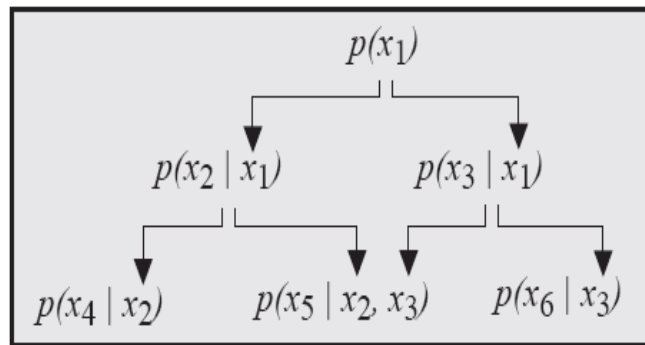
$X_1$	$X_2$	$p(X_2 X_1)$	$X_1$	$X_3$	$p(X_1 X_3)$	$X_2$	$X_4$	$p(X_4 X_2)$
0	0	0.4	0	0	0.2	0	0	0.3
0	1	0.6	0	1	0.8	0	1	0.7
1	0	0.1	1	0	0.5	1	0	0.2
1	1	0.9	1	1	0.5	1	1	0.8

$X_3$	$X_6$	$p(X_6 X_3)$
0	0	0.1
0	1	0.9
1	0	0.4
1	1	0.6

$X_2$	$X_3$	$X_5$	$p(X_5 X_2, X_3)$
0	0	0	0.4
0	0	1	0.6
0	1	0	0.5
0	1	1	0.5
1	0	0	0.7
1	0	1	0.3
1	1	0	0.2
1	1	1	0.8

**Tabla 1E. Probabilidades condicionales requeridas para definir la función de probabilidad conjunta correspondiente a la Red Bayesiana de la Figura E2.**

En primer lugar debe elegirse una numeración ancestral de los nodos. La Figura 3E muestra la estructura ancestral asociada a este ejemplo. En consecuencia, las variables tienen que ser numeradas de arriba a abajo, eligiendo los números de las variables en la misma fila arbitrariamente. Por ejemplo, se elige la ordenación  $(X_1, X_2, X_3, X_4, X_5, X_6)$ .



**Figura 3E. Estructura ancestral de la distribución de probabilidad conjunta asociada a la red de la Figura E2**

Entonces, las variables se muestrean a partir de las distribuciones de probabilidad condicionada de la Tabla E1 como sigue:

1. Muestreo de la variable  $X_1$ : Basándose en  $p(x_1)$ , se consulta un generador de números aleatorios que de un cero con probabilidad 0.3 un uno con probabilidad 0.7. Con objeto de ilustrar el método, supóngase que el valor obtenido para  $X_1$  es  $x_1^1=1$ . Seguidamente se utiliza este valor  $x_1^1$  para calcular las probabilidades de los restantes nodos de esta extracción.
2. Muestreo de la variable  $X_2$  : dado  $X_1=1$ , de la Tabla E1 las probabilidades de que  $X_2$  tome los valores de 0 y 1 son  $p(X_2=0 | x_1^1) = p(X_2=0 | X_1=1) = 0.1$  y  $p(X_2= 1 | X_1=1) = 0.9$ , respectivamente. Por ello, se utiliza un generador de números aleatorios con estas características para obtener el valor simulado de  $X_2$ . Supóngase que dicho valor para  $X_2$  resulta ser  $x_2^1 = 0$ .
3. Muestreo de la variable  $X_3$ : dado  $X_1=1$ , de la Tabla E1  $X_3$  toma los valores 0 y 1 con igual probabilidad. Se genera un valor de  $X_3$  con esta distribución de probabilidad. Si el valor resultante es 0, entonces se rechaza esta extracción por que no coincide con la evidencia  $X_3 = 1$  y se comienza con la etapa 1 de nuevo. En otro caso,  $X_3$  toma el valor de 1 y la simulación continua. Por tanto, se siguen simulando valores para  $X_1, X_2$  y  $X_3$  hasta que  $x_3^3 = 1$ .
4. Muestreo de la variable  $X_4, X_5, X_6$ : la situación es similar a la de las etapas previas. Si el valor simulado del nodo  $X_4$  no coincide con la evidencia  $X_4 = 1$ , se

rechaza la muestra completa y se comienza de nuevo; en otro caso, se simulan los nodos  $X_5$  y  $X_6$ . Supóngase que se obtiene  $x^1_5 = 1$  y  $x^1_6 = 0$ .

La extracción concluye con la primera realización

$$x^1 = (x^1_1, x^1_2, x^1_3, x^1_4, x^1_5, x^1_6) = (1, 0, 1, 1, 1, 0)$$

Se repite el proceso hasta que se obtienen N realizaciones. Entonces, la distribución de probabilidad de cualquier variable se puede estimar por el porcentaje de muestras en las que ocurre el suceso de interés.

## APÉNDICE F SISTEMAS DIFUSOS Y OPERADORES

La lógica tradicional es Bivaluada, es decir; solo toma dos valores: Falso y Verdadero. Regularmente estos valores se toman como 0 (Falso) y 1 (Verdadero).

Existen muchos casos en donde no se conoce con certidumbre el valor de una variable, no se sabe con exactitud cuando será Falso o cuando Verdadero. En este caso se postula que la variable en cuestión tome valores intermedios dependiendo de que tan cerca se juzgue que la variable sea verdadera o falsa. Por ejemplo, la variable “A” puede tomar valores entre 0 y 1 ( $0 \leq A \leq 1$ ).

Para este tipo de lógica, se ve a definir la operación unaria (afecta a un solo elemento) de negación y las operaciones binarias de suma y producto lógico.

### *Negación*

Esta operación esta definida como:

$$\neg A = 1 - A. \quad (1f)$$

esta expresión nos da el valor de certeza de la negación de la variable lógica “A” y tiene como propiedad que:  $\neg \neg A = A$  (doble negación implica afirmación).

### *Suma Lógica*

Definimos el operador de suma lógica como “s”. De tal forma que la suma lógica de las variables “A” y “B” queda expresada como:

$$A s B \quad (2f)$$

este operador aumenta el grado de veracidad de una variable. Por ejemplo, si originalmente  $C=A$  y posteriormente a esta última se le adiciona una variable lógica más, entonces  $C = A s B > A$ .



Este operador debe cumplir ciertas propiedades:

1)  $\text{Máx}(A, B) \leq A \text{ s } B \leq 1$ .

La suma de dos argumentos siempre es mayor o igual al efecto de un solo argumento en la veracidad de una variable.

2)  $A \text{ s } B = B \text{ s } A$  .Propiedad Conmutativa.

El orden en que se presentan los argumentos, no debe alterar el valor de la veracidad de la variable.

3)  $(A \text{ s } B) \text{ s } C = A \text{ s } (B \text{ s } C)$ . Propiedad distributiva.

4)  $A \text{ s } 0 = A$ .

El hecho de que no se presenten más argumentos, no debe aumentar la veracidad de la variable.

### ***Producto Lógico***

Esta operación binaria debe estar definida con “t”. Por ejemplo, el producto lógico de “A” y “B” se representa como:

$$A \text{ t } B \tag{3f}$$

Esta operación también se define como:

$$A \text{ t } B = \neg (\neg A \text{ s } \neg B) \tag{4f}$$

esta expresión es conocida como: *Ley de Morgan*<sup>1F</sup>.

---

<sup>1F</sup> Augustus de Morgan (27-Junio-1806 a 18 – marzo - 1871), matemático y lógico inglés, llevan el nombre de De Morgan las siguientes leyes fundamentales del álgebra de la lógica.  $\neg(A * B) = \neg A + \neg B$ , la negación de un producto entre dos variables lógicas es igual a la suma de las negaciones de las variables.  $\neg(A + B) = \neg A * \neg B$ , El negado de una suma de dos variables lógicas, es igual al producto de los sumandos negados.

El producto lógico decrementa el valor de una variable lógica. Por ejemplo, si  $C=A$  y después  $C=AtB$ , entonces “C” disminuye su valor.

Utilizando la ley de Morgan y las propiedades de la suma lógica, se pueden deducir las siguientes propiedades del producto lógico.

- 1)  $0 \leq AtB \leq \text{Min}(A,B)$
- 2)  $AtB = BtA$
- 3)  $(AtB)tC = At(BtC)$
- 4)  $At1 = A$

En algunos tipos de lógica difusa se cumple la propiedad distributiva. Ejemplo:

$$At(BsC) = (AtB)s(AtC)$$

Si se cumple esta propiedad y utilizando la ley de Morgan, se puede deducir que:

$$As(BtC) = (AsB)t(AsC)$$

En otros casos de lógica difusa si se tiene que:

$$At(BsC) \leq (AtB)s(AtC)$$

aplicando la ley de Morgan se obtiene:

$$As(BtC) \geq (AsB)t(AsC)$$

Existen diferentes tipos de lógica difusa, dependiendo de cómo se defina la suma lógica, a continuación se presentan algunos ejemplos:

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>• 1. <b>Suma-Producto:</b> <math>x + y - xy</math>;</li> <li>• 2. <b>Suma Drástica:</b> <math>\longrightarrow</math></li> <li>• 3. <b>Suma Acotada:</b> <math>\min(1, x + y + pxy)</math>, <math>p \geq 0</math>;</li> <li>• 4. <b>Familia Sugeno:</b> <math>\min(1, x + y + p - xy)</math>, <math>p \geq 0</math>;</li> <li>• 5. <b>Familia Yager:</b> <math>\min(1, \sqrt[p]{x^p + y^p})</math>, <math>p &gt; 0</math>;</li> <li>• 6. <b>Familia Dubois-Prade:</b> <math>1 - \frac{(1-x)(1-y)}{\max(1-x, 1-y, p)}</math>, <math>p \in [0, 1]</math>;</li> <li>• 7. <b>Familia Frank:</b> <math>\log_p \left( 1 + \frac{(p^{1-x} - 1)(p^{1-y} - 1)}{p - 1} \right)</math>, <math>p &gt; 0, p \neq 1</math>;</li> </ul> | $\left\{ \begin{array}{l} x, \text{ si } y = 0; \\ y, \text{ si } x = 0; \\ 1, \text{ en otro caso;} \end{array} \right.$ |
|--|---|

**Figura 1F. Tipos d operadores para Lógica Difusa.**

*Método Probabilístico*

El método probabilístico también funciona como un sistema de lógica difusa. En este se define la certidumbre como probabilidad, por ejemplo; el valor de variable lógica “A” esta definida como la probabilidad de que “A” sea cierto queda definido como:

$$A = P(A) = (\text{número de casos en que A es cierto}) / \text{número total de casos} \quad (5f)$$

*Negación*

La negación de “A” es la probabilidad de que “A” sea falso, y esta definida por:

$$\neg A = P(\neg A) = 1 - P(A) \quad (6f)$$

*Suma Lógica*

La suma lógica para el método probabilístico esta definida como:

$$A \cup B = P(A \cup B) \quad (7f)$$

y expresa la probabilidad de que A=cierto ó B=cierto.

### *Producto Lógico*

Para el método probabilístico queda definido como:

$$A \cap B = P(A \cap B) \quad (8f)$$

y expresa la probabilidad de que A=cierto y B=cierto.

### *Probabilidad Condicional*

La probabilidad condicional de “B” dado “A”, es la probabilidad de que B=cierto dado que A=cierto, y se obtiene de acuerdo a la siguiente ecuación:

$$P(B|A) = (\text{número de casos en que A y B son ciertos}) / (\text{número de casos en que A es cierto}) \quad (9f)$$

de aquí se puede deducir que:

$$P(A \cap B) = P(B|A)P(A) \quad (10f)$$

### *Producto Lógico*

$$A \cap B = P(A \cap B) = P(B|A)P(A) = P(B|A)P(A) \quad (11f)$$

### *Suma Lógica*

$$\begin{aligned} A \cup B = P(A \cup B) &= P(A) + P(B) - P(A \cap B) \\ &= P(A) + P(B) - P(A)P(B|A) \end{aligned} \quad (12f)$$

Los valores de 0 y 1 para este tipo lógica quedan expresados como:

$P(U) = 1$ , probabilidad de conjunto universo.

$P(\emptyset) = 0$ , probabilidad del conjunto vacío.

Los operadores binarios de suma y producto lógico cumplen 4 propiedades de los operadores de lógica difusa mencionados anteriormente.

La demostración puede partir de comprobar las propiedades en la suma lógica y utilizando las leyes de Morgan extenderlas al producto lógico; también puede hacerse la demostración a partir del producto lógico y extender a la suma lógica.

*Propiedad conmutativa*

$$A \text{t} B = P(A)P(B|A)$$

$$B \text{t} A = P(B)P(A|B)$$

*Propiedad asociativa*

$$(A \text{t} B) \text{t} C = P(A \cap B)P(C|A, B), \quad \text{como, } A \text{t} B = P(A)P(B|A) \text{ entonces queda,}$$
$$= P(A)P(B|A) P(C|A, B)$$

$$A \text{t} (B \text{t} C) = P(A)P(B \cap C|A)$$
$$= P(A)P(B|A)P(C|A, B)$$

*Propiedad distributiva*

$$A \text{t} (B \text{s} C) = (A \text{t} B) \text{s} (A \text{t} C)$$

y por extensión mediante la ley de Morgan.

$$A \text{s} (B \text{t} C) = (A \text{s} B) \text{t} (A \text{s} C)$$

Las probabilidades utilizadas en esta lógica difusa, están acotadas como sigue:

$$\begin{aligned} \text{Máx}[P(A),P(B)] &\leq P(A \cup B) \leq \text{Min} [P(A) + P(B), 1] \\ \text{Máx}[0, P(A) + P(B) - 1] &\leq P(A \cap B) \leq \text{Min} [P(A), P(B)] \end{aligned} \quad (13f)$$

## LÓGICA MYCIN

Considera de entrada las propiedades independientes, es decir:

$$P(B|A) = P(B) \quad (14f)$$

Esto quiere decir que, la probabilidad de que B=cierto NO depende del valor que haya tomado la variable "A". Esto nos afecta las operaciones de suma y producto lógico, quedando como sigue:

$$A \text{ s } B = A + B - A * B$$

$$A \text{ t } B = A * B$$

Esta lógica tiene ciertos añadidos por las siguientes razones:

- 1) El argumento se apoya en que A=cierto no debería tomarse como razón para apoyar a que A=falso.
- 2) Los autores decidieron crear una lógica difusa extendida, es decir; la variable lógica puede tomar valores negativos.

$$-1 < A < 1$$

$$A=1, \quad A= \text{cierto}$$

$$A=-1, \quad A= \text{falso}$$

$$A=0, \quad A \text{ es incierto}$$

$$0 < A < 1, \quad \text{Hay cierta certeza de que } A=\text{cierto}$$

$$0 > A > -1, \quad \text{Hay cierta certeza de que } A=\text{falso}$$

Para manejar este tipo de lógica se modifica los operadores de Negación, Producto y Suma lógica.

*Negación*

$$\neg A = -A$$

*Producto Lógico*

$$A \wedge B = \text{Máx}(A,0) * \text{Máx}(B,0)$$

*Suma Lógica*

Ordinariamente el sumar una variable “B” a una variable “A” aumenta el grado de veracidad (tomando como analogía que sería como agregar un argumento más), pero en este tipo de lógica podemos tener lo contrario, es decir; si  $B < 0$  esto puede decrementar la veracidad (como agregar un contraargumento). Por ejemplo:

Si  $A, B > 0$

$$A \vee B = A + B - AB > A$$

y

$$A \vee \neg A < A$$

Para definir la suma lógica en este caso, tenemos que:

$$C = A \vee B = A + B - AB \quad (15f)$$

y necesitamos crear una variable  $\wedge B$ , de tal forma que:

$$C \vee \wedge B = C + (\wedge B) - C(\wedge B) = A \quad (16f)$$

Despejando  $\hat{B}$  y sustituyendo C de (15f) tenemos:

$$\hat{B} = (A-C) / (1-C) = -[B/(1-B)] \quad (17f)$$

sustituyendo  $\hat{B}$  en (16f), nos queda:

$$Cs\bar{B} = (C - B) / (1-B) > 0 \text{ para } (C > B) \quad (18f)$$

Analizando ahora el caso cuando  $C < B$ , creamos una variable  $\hat{D}$  y debe cumplir con  $\hat{D} < 0$ . Analizando de igual forma para cuando  $C > B$ , llegamos a una expresión similar a (17f), por lo tanto tenemos:

$$\hat{D} = -[D/(1-D)] = (C - B) / (1-B), \text{ para } D > 0$$

despejando D, nos queda:

$$D = (B-C) / (1-C)$$

De esto podemos deducir que:

$$Cs\bar{B} = \bar{D}, \text{ ver (18f)}$$

Con esto, podemos llegar a una forma general que incluye los dos casos  $C > B$  y  $C < B$ ,

$$Cs\bar{B} = [C+(\bar{B})] / [1-\text{Min}(B,C)] \quad (19f)$$

Analizamos un ultimo caso, ahora tenemos  $\bar{C}s\bar{B}$  cuando  $C, B > 0$ , para calcular esta suma lógica se tiene que:

$$\hat{D} = \hat{C} + (\hat{B}) - (\hat{B})(\hat{C}) \quad (20f)$$



tomando (17f) y de igual forma encontramos una expresión para  $\hat{C}$  obtenemos que:

$$\hat{B} = -[B/(1-B)] \quad \text{y} \quad \hat{C} = -[C/(1-C)]$$

Sustituyendo  $\hat{B}$  y  $\hat{C}$  en (20f), reduciendo y agrupando términos llegamos a la siguiente expresión:

$$[(-B -C + BC) / (1-B-C+BC)] = -[D/(1-D)]$$

despejando D tenemos:

$$D = B+C-BC \quad (21f)$$

Por lo tanto llegamos a la conclusión de que:

$$\neg C \neg B = \neg D \quad (22f)$$

Esta suma lógica retiene las propiedades conmutativa y asociativa. Puede extenderse este procedimiento a otros tipos de lógica (no todos).

Nótese también, que son las mismas ecuaciones que utilizaron los diseñadores de la lógica MYCIN en su motor de inferencia. Aquí se plantea una forma diferente de llegar a estas.

# BIBLIOGRAFÍA

- ✚ ANGULO Usategui José María y Anselmo del Moral Bueno.  
Guía fácil de la Inteligencia Artificial. Editorial Paraninfo.  
Segunda Edición. Madrid, 1994.
  
- ✚ ARTEAGA René y Juan Carlos Armijos.  
**Tutorial de Programación Heurística.**  
[URL\(www.uc3m.es/cgi-bin/nph-count\)](http://www.uc3m.es/cgi-bin/nph-count). 2-Sep-1998.
  
- ✚ ASOCIACIÓN ARGENTINA DE INTELIGENCIA  
ARTIFICIAL. **Glosario.**  
[URL\(http://www.lafacu.com/apuntes/ingenieria/asociacion\\_argentina\\_de\\_inteligencia\\_artificial/asociacion\\_argentina\\_de\\_inteligencia\\_artificial.htm\)](http://www.lafacu.com/apuntes/ingenieria/asociacion_argentina_de_inteligencia_artificial/asociacion_argentina_de_inteligencia_artificial.htm).
  
- ✚ BAÑARES, José Ángel. **Herramientas para las Asignaturas del IAIC.** [URL\(http://diana.cps.unizar.es/banares/IA/noticias.html\)](http://diana.cps.unizar.es/banares/IA/noticias.html), 8-Oct-1998.
  
- ✚ BENCHIMOL Guy, Pierre Levine y Jean Charles Promerol.  
**Los sistemas expertos en la empresa.** Macrobit. Primera edición, México, 1990.
  
- ✚ BONSÓN Enrique. **Tecnologías Inteligentes para la Gestión Empresarial.** Alfaomega – Rama. Primera edición. México, 1999.

- ✚ BRAIN Keith y Steven Brain. **Inteligencia Artificial en el Dragón**. Traductor: Jordi Abadal Berini. Editorial Gustavo Gili. Primera Edición. México, 1984.
  
- ✚ BRATKO Ivan. **Prolog programming for Artificial Intelligence**. Addison Wesley. Segunda Edición. New York, 1990.
  
- ✚ BRUCE G. Buchanan, Edward H. Shortliffe  
Rule-Based Expert System  
Addison-Wesley Publishing Company
  
- ✚ CASTILLO Enrique, José Manuel Gutierrez y Adi S. Hadi  
Sistemas Expertos y Modelos de Redes Probabilísticas  
Universidad de Cantabria 39005, Santander, España  
E-mail: [castie@ccaix3.unican.es](mailto:castie@ccaix3.unican.es)
  
- ✚ CONTRERAS Carlos. **INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL**. [URL\(www.gdl.uag.mx/66/0ia.htm\)](http://www.gdl.uag.mx/66/0ia.htm).
  
- ✚ CRIADO Briz José Mario. **Introducción a los Sistemas Expertos**.  
[URL\(www.ingenieroseninformatica.org/recursos/tutoriales/sist\\_exp/index.php\)](http://www.ingenieroseninformatica.org/recursos/tutoriales/sist_exp/index.php). 2000.
  
- ✚ CRIADO Briz José Mario. **Sistemas Expertos**.  
[URL\(http://home.worldonline.es/jmariocr/\)](http://home.worldonline.es/jmariocr/). 2002 .
  
- ✚ CRUZ Roberto. **Área de Bases de Datos e Inteligencia Artificial**.  
[URL\(http://dcc.ing.puc.cl/investigacion/areas/bases\\_dat.html\)](http://dcc.ing.puc.cl/investigacion/areas/bases_dat.html).
  
- ✚ DE ALBORNOZ Bueno Álvaro. **Laboratorio de Procesamiento de Imágenes**.  
[URL\(http://www.cic.ipn.mx/organización/lab\\_de\\_int\\_art.htm\)](http://www.cic.ipn.mx/organización/lab_de_int_art.htm).

- ✚ DE ÁVILA Ramos Jorge. **Sistemas Expertos.**  
URL([http://www.lafacu.com/apuntes/informatica/sist\\_expe/](http://www.lafacu.com/apuntes/informatica/sist_expe/)).
  
- ✚ DE MIGUEL González Luis Javier. **Técnicas de Mantenimiento Predictivo Industrial basadas en Sistemas Expertos.**  
URL(<http://www.cartif.es/mantenimiento/expertos.html>)
  
- ✚ DÍEZ Vegas Francisco Javier. **Sistema Experto Bayesiano para Ecocardiografía.** URL(<http://www.ia.uned.es/~fdiez/tesis/tesis.html>),  
22-Feb-1999.
  
- ✚ ELGUEA Javier. **Inteligencia artificial y psicología: la concepción contemporánea de la mente humana.**  
URL([http://www.hemerodigital.unam.mx/ANUIES/itam/estudio/estudio10/sec\\_16.html](http://www.hemerodigital.unam.mx/ANUIES/itam/estudio/estudio10/sec_16.html)). 1987.
  
- ✚ GONZÁLEZ Ayala Luis Enrique. **Sistemas Expertos.**  
URL(<http://www.prodigyweb.net.mx/enrayala/Sistemas.htm>)
  
- ✚ HARMON Paul y Curtis Hall. **Intelligent Software Systems.** Editorial John Wiley & Sons Inc. Primera edición. New York, 1993.
  
- ✚ HASEMER Tony y John Domingue. **Common LISP Programming for Artificial Intelligence.** Addison Wesley. Primera edición. New York, 1989.
  
- ✚ HOPGOOD Adrian. **Knowledge – Based Systems for Engineers and Scientists.** Editorial CRC. Primera edición. Londres, 1993.

- ✚ HURTADO Vega José de Jesús. **Inteligencia Artificial**.  
URL(<http://www.itlp.edu.mx/publica/boletines/actual/inteligencia.html>).
  
- ✚ JOSEPH Giatarrano, Gary Riley  
Sistemas Expertos, Principios y programación 3ª Ed.  
International Thomson Editores
  
- ✚ KANDEL Abraham. **Fuzzy Expert Systems**. Editorial CRC. Primera edición, Londres, 1992.
  
- ✚ MARTÍNEZ De Ibarreta León, Francisco Javier.  
**Sistemas Expertos: Áreas de Aplicación**.  
URL([www.geocities.com/SiliconValley/Way/7788/SISEXP.HTM](http://www.geocities.com/SiliconValley/Way/7788/SISEXP.HTM)). 1-Abr-1994.
  
- ✚ PARSAYE Kamran, Mark Chignell, Setrag Khoshafian y Harry Wong.  
**Intelligent Databases**. Editorial Wiley. Primera Edición. New York, 1989.
  
- ✚ RICH Elaine y Kevin Knight. **Artificial Intelligence**. Mc GrawHill. Segunda edición. México, 1991.
  
- ✚ ROLSTON, David W. **Principios de Inteligencia Artificial y Sistemas Expertos**. Traductor: Alfonso Pérez Gama. McGraw Hill. Primera edición. México, 1992.
  
- ✚ ROSS Rita. **Historia de los Sistemas Expertos**.  
URL([www.monografias.com/trabajos10/exper/exper.shtml](http://www.monografias.com/trabajos10/exper/exper.shtml)).

- ✚ RUSSELL Stuart y Peter Norving. **Inteligencia Artificial: Un enfoque moderno.** Traductor: Raúl Bautista Gutiérrez. Prentice Hall. Primera edición. México, 1996.
  
- ✚ SAMPER Márquez Juan José. **SISTEMAS EXPERTOS. EL CONOCIMIENTO AL PODER.**  
[URL\(\[www.psycologia.com/articulos/ar-jsamper01.htm\]\(http://www.psycologia.com/articulos/ar-jsamper01.htm\)\)](http://www.psycologia.com/articulos/ar-jsamper01.htm). 2002.
  
- ✚ SÁNCHEZ Tomás Antonio. **Aplicación de los Sistemas Expertos en Contabilidad.**  
[URL\(<http://ciberconta.unizar.es/Biblioteca/0002/Sanchez95.html#CONTABILIDAD>\)](http://ciberconta.unizar.es/Biblioteca/0002/Sanchez95.html#CONTABILIDAD). 1990.
  
- ✚ SÁNCHEZ y Beltrán Juan Pablo. **Sistemas Expertos: Una metodología de programación.** Macrobit. Primera edición. México, 1990.
  
- ✚ SCARABINO Juan Carlos. **Sistemas Expertos: Aspectos técnicos.**  
[URL\(<http://ciberconta.unizar.es/LECCION/sistexpat/INICIO.HTML>\)](http://ciberconta.unizar.es/LECCION/sistexpat/INICIO.HTML).
  
- ✚ SCHILDT Herbert. **Utilización de C en Inteligencia Artificial.**  
Traductor: José Andrés Moreno Ruiz. Mc GrawHill. Primera edición. México, 1989.
  
- ✚ SELL Peter. **Sistemas Expertos para principiantes.** Traductor: Hugo Villagómez Velásquez. Noriega Editores. Primera edición. México, 1989.
  
- ✚ TELLO Ernest. **Object Oriented Programming for Artificial Intelligence.** Addison Wesley. Primera edición. New York, 1989.