Implementación de un Sistema Experto Ginecológico en Prolog

Norma Verónica Ramírez Pérez
Instituto Tecnológico de Celaya
norma.ramirez@itcelaya.edu.mx

Angélica Jiménez Morales
Instituto Tecnológico de Celaya
angie_jimenezm_91@hotmail.com

Martín Laguna Estrada
Instituto Tecnológico de Celaya
martin.laguna@itcelaya.edu.mx

Resumen

En este este artículo se muestra cómo realizar un sistema experto en Prolog, (PROgramming in LOGic). Aprovechando que es un lenguaje de programación declarativo, trabaja con lógica de predicados, hechos y cláusulas que ayudan a modelizar el conocimiento, permitiendo de esta manera, contar con un sistema experto basado en reglas. Para realizar esta propuesta, se trabajó con una serie de reglas extraídas del conocimiento de especialistas en la materia, lo cual permitió describir conceptos importantes de prolog y conjugarlo con sistemas expertos.

***Palabras clave*:** Prolog, inteligencia artificial, sistemas expertos, reglas.

1. Introducción

Frecuentemente en el aula se tratan temas teóricos que más allá de llevarlos a práctica, se quedan en la libreta del alumno. Para salir de este estatus, se pretende dar a conocer lo que se realiza en algunas materias de la carrera de Ingeniería en Sistemas

computacionales, que es aterrizar la teoría en cuestiones prácticas y difundir temas relevantes como son la inteligencia artificial, sistemas expertos (SE), su metodología y herramientas para que los alumnos puedan llevar a cabo prácticas y conozcan las aplicaciones reales del campo de la inteligencia artificial.

Los sistemas expertos surgen de las técnicas de la inteligencia artificial que han sido objeto de amplias investigaciones desde el año de 1950, pero la investigación en este tema comenzó realmente en los 60's donde surgieron los primeros artículos en este campo.

Sánchez y Beltrán [1], en su trabajo sobre sistemas expertos, establecen una metodología de programación y describen la siguiente definición: *"Un sistema experto es un conjunto de programas que son capaces, mediante la aplicación de conocimientos, de resolver problemas en un área determinada del conocimiento o saber que ordinariamente requirieran de la inteligencia humana"*, mientras que Kandel [2], define un sistema experto como *"un sistema informático que simula los procesos de aprendizaje, memorización, razonamiento, comunicación y acción de un experto humano, en una determinada ciencia, suministrando de esta forma, un consultor que puede sustituirle con unas ciertas garantías de éxito"*. Sin embargo, todo esto sugiere que el conocimiento sólo se adquiere con un largo aprendizaje y en base a experiencia, de ahí que un sistema experto deba tener ciertos elementos para su correcta funcionalidad. Se requiere que dicho sistema al ser instalado en un ordenador, sea capaz de resolver, por medio de inferencias, problemas específicos de razonamiento con buenas probabilidades de solución.

Una de las características fundamentales de un sistema experto, es que permite hacer una separación de los conocimientos que se almacenan en el mismo programa y situarlos en una base de datos. Debido a las características propias de los sistemas expertos, es que se puede trabajar con reglas y facilitar su implementación con Prolog, que es un lenguaje de programación declarativo basado en formalismos abstractos por medio de lógica de predicados.

2. Métodos

Prolog

Prolog[3] es el primer lenguaje de programación lógica y fue desarrollado en Marsella, Francia, iniciando con Alain Coulmeauer y Philippe Roussel. La primera implementación de Prolog se completó en Edimburgo en 1972 usando el compilador de ALGOL y los aspectos básicos del lenguaje actual que concluyeron en 1973. Como consecuencia de este lenguaje, surgieron nuevos intérpretes, todos ellos derivados del Prolog original. Entre éstos podemos encontrar a SWI-Prolog[4], CIAO-Prolog[5], GNU-Prolog[6], e incluso algunos intérpretes comerciales, como el Sictus Prolog[7]. La característica principal de estos intérpretes es su sintaxis relativamente simple, caracterizada por su tratamiento generalizado de variables, constantes y de valores como “objetos” opacos, es decir, no se especifica explícitamente el dominio de los objetos, los cuales se pueden manipular de manera intercambiable en la mayor parte de los casos.

Prolog es un lenguaje de programación declarativo que se diferencia de los lenguajes imperativos o procedurales, porque está basado en formalismos abstractos. Un programa Prolog está formado por un conjunto de hechos y reglas, también llamadas cláusulas del programa. Un hecho corresponde a un predicado y una regla corresponde a una implicación cuyo antecedente es una conjunción de predicados y cuyo consecuente es un predicado. Tanto las variables de los hechos como de las reglas, están cuantificadas universalmente, un ejemplo de ello es el programa Prolog con una regla y dos hechos:

$$\textit{Regla:}\quad \forall X, \forall Y, \forall Z, padre(X,Y) \wedge padre\ (Y,Z) \rightarrow abuelo(X,Z)$$

$$\textit{Hechos:}\quad padre(juan, luis).$$

$$padre(luis, pedro).$$

En la sintaxis de Prolog se omiten los cuantificadores y se sobreentiende que todas las variables están cuantificadas universalmente; el signo de implicaciones se sustituye por $:-$, colocando el antecedente a su derecha y el consecuente a su izquierda; la conjunción de predicados del antecedente de una regla se representa por comas,

mientras que las variables se escriben en mayúsculas y las constantes en minúsculas. Por tanto, el ejemplo anterior, se expresa en Prolog de la siguiente manera:

$$Regla: \quad abuelo(X,Z), padre(X,Y), padre(Y,Z)$$

$$Hechos: \quad padre(juan, luis).$$

$$padre(luis, pedro).$$

La introducción de este programa, permite realizar preguntas, que corresponden a una conjunción de predicados cuyas variables están cuantificadas existencialmente. En Prolog, una pregunta es representada por un signo de interrogación seguido de uno o más predicados separados por comas. Si retomamos el ejemplo anterior, al hacer la pregunta de quién es el abuelo de pedro, ésta estaría representada de la siguiente manera:

$?\, abuelo(X, pedro).$

Prolog nos respondería "$X\ es\ Juan$", es decir la única respuesta correcta. Pero si se hace la pregunta $?abuelo(X,Y)$. nos respondería "$X = juan;\ Y = pedro$", recordando que equivale a preguntar si existe X y algún Y, tales que X es abuelo de Y. Si la pregunta fuera $?\, padre(X,Y)$, obtendríamos dos respuestas válidas:

$(X = juan, Y = luis)\ y\ (X = luis, Y = pedro).$

En los ejemplos que se han expuesto, sólo se han considerado variables y constantes, en general. Un programa de Prolog puede contener términos de primer orden $t_1, \ldots, t_n$ en sus predicados, por ejemplo:

$$pariente(nieto(Z,X)) :- hijo(Z,Y), hijo(Y,X)$$

$$pariente(tio(Z,X)) :- hijo(Y,X), hermano(Y,Z)$$

$$pariente(sobrino(X,Z)) :- pariente(tio,(Z,X))$$

De lo anteriormente expuesto, se puede observar que tiene los siguientes términos:

$$nieto(Z,X),\ tio(Z,X),\ sobrino(X,Z)$$

Prolog proporciona dos puntos relevantes importantes para diseñar un interpretador eficiente y así obtener rápidamente las soluciones de un problema si existieran: la ejecución de la inferencia y la estrategia de encadenamiento de inferencias:

1. En la etapa de inferencia (unificación), interviene una operación de unificación que se define brevemente teniendo dos predicados: $P(t_1,\ldots,t_n)$ y $Q(s_1,\ldots,s_n)$, de los cuales se pretende realizar una sustitución $\sigma = \{X_1/r_1,..,X_k/r_k\}$ llamado mgu (unificador más general), que verifica lo siguiente:
   a) Si se sustituye en los predicados $P(t_1,\ldots,t_n)$ y $Q(s_1,\ldots,s_n)$, cada va-riable X_i por el término r_i indicando la sustitución σ, entonces los dos predicados son sintácticamente iguales, es decir, $P(t_1,\ldots,t_n).\sigma$ y $Q(s_1,\ldots,s_n).\sigma$.
   b) Se puede obtener cualquier otro unificador σ' de $P(t_1,\ldots,t_n)$ y $Q(s_1,\ldots,s_n)$, por una operación de composición entre σ y una tercera sustitución p, i.e. $\sigma' = \sigma * p$. Más información en [8].

   La unificación es considerada una operación para reducir el tiempo que el interpretador necesita para encontrar soluciones, ya que interviene en cada inferencia. En la actualidad existen muchos algoritmos de unificación, que han sido diseñados para obtener rápidamente un unificador más general.

   Definida la unificación, se puede indicar el proceso de ejecución de una inferencia y su papel en la resolución del problema. En Prolog, la ejecución de una inferencia recorre los siguientes pasos:

   a) Se elige un átomo Q de la conjunción actual de los predicados y un hecho P, o una regla con el consecuente P si Q y P son unificables.
   b) Obtener σ su unificador más general.
   c) La conjunción de predicados se transforma, extrayendo Q si P es un hecho, o bien, reemplazando el predicado Q por el conjunto de predicados antecedentes de la regla. El último paso consiste en aplicar σ al conjunto de átomos de la conjunción actual.

2. En la estrategia de búsqueda (SLD), es considerado como el encadenamiento de inferencias que realiza el interpretador para obtener las posibles soluciones. El interpretador es el programa que recibe como entrada un problema en Prolog. Su objetivo es el encadenar inferencias para obtener soluciones al problema que le hayan propuesto, considerando que una de sus propiedades del interpretador es que si existen soluciones al problema planteado, el encadenamiento de inferencias deberá ser tal que siempre encuentre todas las soluciones posibles, por lo que recíprocamente, cada respuesta aportada por el interpretador como una solución, debe ser realmente una solución del problema declarado en Prolog.

   Los interpretadores de Prolog están basados en un refinamiento de resoluciones SLD, donde la conjunción de átomos de un estado que modela un subproblema se considera ordenada y se procesa como una pila FIFO (el primer predicado en entrar es el primero en salir, es decir, en ser resuelto).

Sistemas expertos basados en reglas

Los sistemas expertos basados en reglas [9] son una herramienta eficiente para tratar problemas de sistemas de control, sistemas de seguridad, sistemas de transacciones etc., debido a que se ocupan reglas deterministas que constituyen la más sencilla de las metodologías utilizadas en sistemas expertos. La base de conocimientos contiene las variables y el conjunto de reglas que definen el problema y el motor de inferencia obtiene las condiciones aplicando la lógica clásica a estas reglas. Esta regla se entiende como una proposición 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 los operadores lógicos y, o, o no.

Una regla se escribe normalmente como “si premisa, entonces conclusión”. Este tipo de reglas las podemos ejemplificar como cuando un cliente de una sucursal bancaria desea sacar dinero de su cuenta corriente mediante un cajero automático, en cuanto el

cliente introduce la tarjeta en el cajero, la máquina lee y verifica; si la tarjeta no es verificada con éxito, mandará un mensaje de error, esto puede ser también si no tiene dinero disponible, etc. Por lo que existen diversos objetos que la máquina determina cual será el correcto de acuerdo a la transacción que desee el cliente. Así pues, el sistema experto basado en reglas deterministas constituye la metodología más sencilla utilizada en los sistemas expertos.

Base de conocimientos: Contiene el conocimiento especializado que se extrae del experto en el dominio, es decir, contiene el conocimiento general sobre el dominio en el que se trabaja.

Base de hechos: Se considera como una parte de la memoria del ordenador que se utiliza para almacenar los datos que se reciben inicialmente para la resolución de un problema. Contiene conocimiento sobre el caso concreto en que se trabaja.

Motor de inferencia: modela el proceso de razonamiento humano, es decir, controla el proceso de razonamiento que seguirá el sistema experto, utilizando los datos que se le suministran, recorre la base de conocimiento para alcanzar una solución.

Módulo de explicación: está diseñado para aclarar al usuario la línea de razonamiento seguida en el proceso de inferencia.

Interfaz del usuario: permite al usuario pueda describir el problema al sistema experto, por medio de preguntas e información ofrecida.

Por otro lado, los SE pueden ser de tres tipos: los que están basados en reglas, los que están basados en casos y los basados en redes bayesianas. En este trabajo el enfoque estará centrado sólo en la primera, es decir, basado en reglas. Se emplea el lenguaje Prolog, que como ya se mencionó en el apartado 2 trabaja con lógica de predicados para realizar sistemas expertos de una manera más sencilla.

3. Resultados

Para la demostración del uso de Prolog en los sistemas expertos, se realizó un sistema para modelar el conocimiento de un experto ginecológico en cesáreas. Por lo que para extraer el conocimiento, se realizaron tres entrevistas que llevaron a la toma de

decisiones sobre el tipo de cesáreas a realizar, de acuerdo a las características presentadas por la paciente. En dicho conocimiento, se sabe que existen dos tipos de cesáreas: las previstas de antemano, o programadas, y las improvisadas durante el parto (en donde la paciente presente algunos síntomas que pueden poner en riesgo al bebe). Se pueden deducir dos conclusiones con este sistema:

- Cesáreas programas
- Cesáreas no programadas

Las cesáreas programadas se producen cuando se dan alguna de las siguientes circunstancias:

- El bebé está en posición podálica.
- La futura madre padece durante el embarazo alguna de las dos enfermedades: gestosis o diabetes gravídica.
- La placenta está en posición previa-central
- La madre tiene problemas de corazón, renales o graves infecciones en vías genitales.

Las cesáreas no programadas, se producen cuando:

- La cabeza del niño es demasiado grande y no cabe por el canal del parto.
- La cabeza del niño no está encajada correctamente en el canal del parto.
- Existe sufrimiento fetal, lo que significa que el ritmo cardiaco del bebe ya no es regular que empieza a expulsar meconio.
- La placenta se desprende.

También contamos con información para realizar tipos de incisiones, que también se determinan en algunas circunstancias:

Si la cesárea no está programada y la placenta se desprende, se realiza una incisión umbilical púbica; en el resto de los casos se realiza una cesárea del tipo transversal baja o de Joel Coell, de acuerdo a esto contamos con información sobre cada tipo de incisión.

- Umbílico púbica: es vertical, empieza debajo del ombligo y termina en el pubis . la cicatriz es visible y grande.
- Transversal baja: es horizontal, la cicatriz no es visible.
- De Joel Coell, es horizontal.

También, se cuenta que con independencia del tipo se incisión que se realice, las fases de la intervención, siempre suceden de la misma forma y en el siguiente orden: cortar, extraer el niño, extraer la placenta y suturar la herida.

Para la modelización del conocimiento, se realizó la representación basada en reglas, utilizando el lenguaje de programación Prolog, debido a que como ya se había mencionado en el segundo apartado, trabaja con lógica de predicados para facilitar la modelización de este conocimiento. A continuación se presenta un ejemplo con alguna de las condiciones que se tienen para una cesárea programada:

"Si el bebé está en posición podálica, entonces cesárea programada"

En lógica de predicados se presentaría la misma condición de la siguiente manera:

$$\forall x(bebé\ posición\ podálica(x) \rightarrow cesárea\ programada(x)$$

el símbolo $\rightarrow$ significa "si".

En Prolog la condición anterior se presenta como:

cesárea programada :- bebe está en posición podálica

En el caso de Prolog el símbolo :- significa "si".

La condición expuesta nos indica que si el bebé está en posición podálica, entonces se activará la conclusión de cesárea programada.

En la tabla no. 1 se muestran algunas de las diferencias de escritura existentes entre la lógica de predicados y Prolog.

Como se puede apreciar, las reglas son fáciles de construir en Prolog, por lo que se procedió a hacer cada una de las reglas para llegar a una conclusión, como se muestra en la tabla no. 2.

Tabla 1 Comparación de conectivas.

Conectiva	Lógica de predicados	prolog
y	$\wedge$	,
o	$\vee$	;
si	$\rightarrow$	:-

Tabla 2 Algunas reglas del sistema.

Reglas
Si el bebé está en posición podálica entonces cesárea programada
Si la madre padece gestosis entonces cesárea programada…..
Si la cabeza del niño es demasiado grande y no cabe en el canal del parto entonces cesárea no programada
Si cesárea no programa y se desprende la placenta entonces incisión umbilical……

A continuación la figura 1 muestra la estructura del sistema experto ginecológico de cesáreas.

Figura 1 Estructura del sistema experto ginecológico en cesáreas.

Para los datos de entrada, se tuvo que realizar un análisis para saber y determinar cada de una de las conclusiones a las que quería llegar el experto, de acuerdo a la información obtenida.

En la base de hechos, se desarrollaron cada una de las reglas que se generaron con la información que otorgó el experto. Los hechos que se manejaron inicialmente se relacionan con la respuesta que se puede obtener del usuario:

respuesta(si).

respuesta(no).

Estas respuestas nos van a permitir realizar la activación de las reglas, que se muestran en la tabla 2, dichas reglas se estructuraron de la siguiente manera: se colocó un antecedente de regla, que en este caso es la pregunta, seguido de un consecuente que es la respuesta que se dará a la pregunta, almacenando ésta en una variable para poder almacenar el resultado obtenido y compararlo con uno de los hechos declarados, las cuales se muestran en la tabla 3.

Tabla 3 Reglas de la base de hechos.

pregunta('el_bebé_está_en_posición_podálica',X) :- respuesta(X).
pregunta('la_madre_padece_gestosis',X) :- respuesta(X).
pregunta('la_madre_padece_diabetes_gravídica',X) :- respuesta(X).
…

La base de conocimientos es aquella que está compuesta por hechos y reglas, por lo tanto, todo el conocimiento que requiere el sistema se encuentra almacenado en la base de conocimientos, es decir, dicha información nos permite proporcionar los distintos caminos que se tienen para poder generar una solución, que son cesáreas programadas y no programadas, las cuales se describen en una interfaz en la figura 2 y 3 respectivamente, mientras que las incisiones a realizar, también se muestran en la figuras 4 y 5.

Una vez que se ha terminado con la base de hechos y la base de conocimientos, lo que sigue es la elaboración del mecanismo de inferencia que permita realizar un mecanismo de razonamiento y control de una búsqueda para poder arrojar una solución y extraer la información de la base de conocimientos. Se puede confirmar de lo anterior que un

hecho existirá cuando se encuentre almacenado en la base de conocimientos. Para la elaboración del mecanismo de inferencia, se tomaron en cuenta las reglas de la tabla no.3 considerando las respuestas brindadas para poder arrojar un diagnóstico, las cuales se muestran en la tabla 4.

Figura 2 Cesárea programada.

Figura 3 Cesárea no programada.

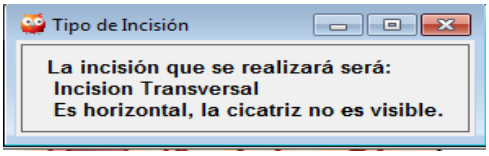

Figura 4 Incisión transversal.

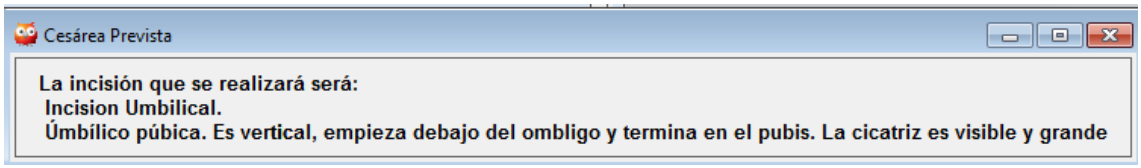

Figura 5 Incisión umbilical.

Tabla 4 Preguntas para el motor de inferencia.

pregunta('el_bebé_está_en_posición_podálica',N),N='si',
pregunta('la_madre_padece_gestosis',O),O='si',
pregunta('la_madre_padece_diabetes_gravídica',P),P='si',
pregunta('la_placenta_está_en_posición_previa_central',Q),Q='si',
…
cesárea_prevista;

De acuerdo a las respuesta proporcionadas por el usuario, se arrojará el tipo de cesárea que se debe de realizar con la incisión respectiva.

Se creó un módulo de interacción con el usuario, por medio de una interfaz que se realizó con el lenguaje swi-prolog, el cual cuenta con el módulo XPCE que puede trabajar con objetos para su mejor visualización. Para poder realizar la interfaz, se necesita tener la importación de las librerías necesarias y un directorio que contenga las imágenes a utilizar, las cuales son:

```
:- use_module(library(pce)).
:- use_module(library(pce_util)).
:- pce_image_directory('./iconos').
```

Después se comienza con la creación de las ventanas que serán utilizadas, tomando en cuenta la siguiente sintaxis:

```
comenzar :-
```

Dentro de la estructura se hace la declaración de los componentes de la ventana, que serán mostrados en la tabla 5.

Como se puede observar en la figura 5, se muestra la ventana principal del sistema experto, mostrando el nombre de la ventana, texto, botones y una imagen dentro de la misma.

Tabla 5 Sintaxis de los elementos de una ventana.

Nombre de la ventana	new(A, dialog('Sistema Experto Ginegología')),
Texto	send(A, append, new(Txt3, text('Sistema Experto para la realización de césareas'))),
	send(Txt3, font, bold),
	send(Txt3,alignment, center),
Imagen	send(A, append, new(P, picture), next_row),
	new(Fig, figure),
	new(Bitmap, bitmap(resource(bebe), @on)),
Botón	send(A, append, new(Btn, button(diagnóstico, and(message(@prolog, diagnostico))))),

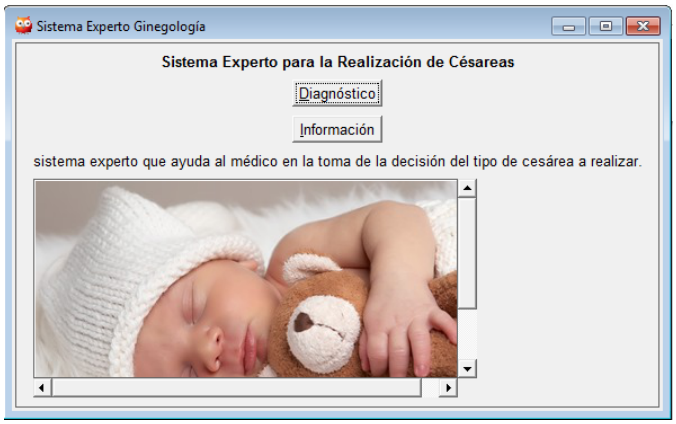

Figura 5 Interfaz principal del sistema experto.

Las preguntas que se le realizan al usuario son presentadas en una ventana, indicando en cada una de las preguntas las opciones de respuesta que se pueden dar, dichas preguntas están relacionados con la base de hechos que se tienen y el resultado arrojado será el que el mecanismo de inferencia deduzca de las reglas que se tienen almacenadas, mandando llamar a la solución que se ha llegado. Para el manejo de las preguntas y respuestas se utiliza la siguiente sintaxis:

send_list(B, append, [new(N, menu(el_bebe_esta_en_posición_podálica)), …]},

send_list(N, append, [si, no]), …

Posteriormente se mandan como parámetros cada una de las respuestas brindadas, de la siguiente manera:

send(B, append, new(Btn, button(cesárea, and(message(@prolog, preguntas, N?selection,O?selection,P?selection,Q?selection,R?selection,S?selection,T?selection,U?selection,V?selection,W?selection,X?selection))))), send(Btn, open).

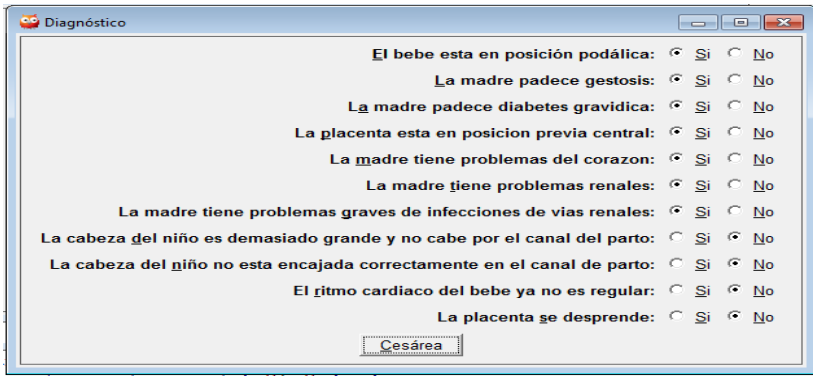

Figura 6 Interfaz de diagnóstico.

4. Conclusiones

La inteligencia artificial es una ciencia multidisciplinaria, donde se pueden involucrar diferentes áreas como la psicología, la tecnología y la medicina entre otras, debido a que modela el funcionamiento de los procesos de razonamiento. En el caso de los sistemas expertos, la inteligencia artificial puede emular el razonamiento humano por medio del conocimiento de un experto.

La demostración del sistema experto expuesto en este trabajo, puede ser de mucha ayuda para iniciarse en las áreas de la inteligencia artificial, en especial los sistemas expertos, además de conocer un poco más del lenguaje de programación Prolog, que ha ganado popularidad en la inteligencia artificial.

Bibliografía

[1] Sánchez y Beltrán, J. P. , Sistemas Expertos una metodología de programación, Editorial Macrobit, 1990.

[2] Kandel, Abrahm , Fuzzy Expert Systems, editorial CRC, 1ª edition, Londres, 1992.

[3] Colmerauer, A. The Birth of Prolog. In the Second ACM-SIGPLAN History of Programming Languages Conference, ACM SIGPLAN Not., pp. 37-52, 1993.

[4] SWI-PROLOG: http://www.swi-PROLOG.org.

[5] Ciao: http://www.ciaohome.org.

[6] GNU-PROLOG: http://www.gPROLOG.org.

[7] SICS AB. Sictus PROLOG: http://www.sictus.se/sictus.

[8] Lloyd, J.W. Foundations of Logic Programming, Springer-Verlag, 1987.

[9] Giarratano, J., Riley, G., Sistemas Expertos, Principios y programación, 3ª edición, Thomson, 2001.