



**Universidad Nacional Autónoma de México**

**FACULTAD DE INGENIERIA**

**USO DE SISTEMAS EXPERTOS EN  
LA EDUCACION**

**T E S I S**

Que para obtener el título de:

**INGENIERO EN COMPUTACION**

P r e s e n t a :

**GERARDO AYALA SAN MARTIN**



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## TEMARIO.

=====

INTRODUCCION .....	1
--------------------	---

### CAPITULO I

-----

#### LAS COMPUTADORAS EN LA EDUCACION.

1.1.- La inteligencia y las operaciones concretas y formales .....	3
1.2.- Métodos pasivos y métodos activos .....	5
1.3.- CAI y LOGO .....	7
1.4.- Capacidad educativa de las máquinas .....	9
1.5.- El uso de las computadoras en la escuela .....	11
1.6.- Problemas para la introducción de las computadoras en la educación .....	12
1.7.- La publicidad del uso de la computadora .....	16
1.8.- Software y equipo .....	17
1.9.- El uso de las computadoras en la educación en México y otros países .....	20
1.10.- El problema del profesorado .....	21

### CAPITULO II

-----

#### ELEMENTOS Y CARACTERISTICAS DE LOS SISTEMAS EXPERTOS.

2.1.- La inteligencia artificial .....	25
2.2.- Definición de sistema experto .....	27
2.3.- Diferencias entre los sistemas expertos y otros sistemas .....	29
2.4.- Funciones de los sistemas expertos .....	31
2.5.- Aplicaciones .....	32
2.6.- Configuración general .....	34
2.7.- La base de conocimientos .....	36
2.7.1.- Los conceptos .....	36
2.7.2.- Las reglas .....	40
2.7.3.- Otra alternativa en la representación de conocimiento .....	43
2.8.- El espacio de trabajo .....	49

2.8.1.- El espacio de estados .....	49
2.8.2.- El uso de una pizarra en algunos sistemas expertos .....	49
2.9.- La máquina de inferencia .....	51
2.9.1.- Búsqueda de soluciones .....	53
2.9.2.- Búsqueda hacia adelante .....	54
2.9.3.- Búsqueda hacia atrás .....	54
2.9.4.- Búsqueda bidireccional .....	56
2.9.5.- Reducción del problema .....	57
2.9.6.- Selección de reglas aplicables.....	58
2.9.7.- Evaluación de reglas .....	58
2.10.- La interfaz con el usuario .....	60
2.10.1.- Interfaz a lenguaje natural .....	60
2.10.2.- Interfaz basada en menús .....	62
2.10.3.- Gramáticas semánticas .....	64
2.10.4.- Gramáticas sintácticas .....	64
2.10.5.- Comunicación entre expertos .....	65
2.11.- Subsistema de adquisición de conocimiento .....	66
2.11.1.- La labor del ingeniero en conocimiento ....	66
2.11.2.- La etapa de adquisición de conocimiento ...	66
2.11.3.- Métodos .....	67
2.11.4.- Problemas en la adquisición de conocimien- to .....	71
2.12.- Subsistema de explicación .....	72
2.12.1.- Reconstrucción simple .....	72
2.12.2.- Razonamiento acerca del razonamiento .....	77
2.13.- Construcción de sistemas expertos .....	79
2.13.1.- Desarrollo de un sistema experto .....	79
2.13.2.- Herramientas para la construcción de siste- mas expertos .....	80
2.14.- Algunos sistemas expertos .....	81

### CAPITULO III

#### LOS SISTEMAS INTELIGENTES DE EDUCACION ASISTIDA POR COMPUTADORA.

3.1.- Características de los sistemas de ICAI .....	86
3.2.- Elementos de un sistema de ICAI .....	86
3.2.1.- El módulo de experiencia .....	87
3.2.2.- El módulo tutorial .....	88
3.2.3.- El módulo de representación del estudiante ..	89
3.2.4.- El módulo de comunicación .....	90
3.3.- Ventajas y desventajas de los sistemas de ICAI ....	90
3.4.- Experiencias con sistemas tutoriales .....	92
3.5.- Algunos sistemas .....	92

## CAPITULO IV

### PEQUEÑO SISTEMA EXPERTO DE PROPOSITO GENERAL.

4.1.- Características generales .....	98
4.2.- Configuración de los archivos .....	100
4.3.- Configuración de los datos en memoria .....	101
4.4.- Configuración de los programas del sistema .....	104
4.4.1.- Programa de adquisición de conocimiento .....	105
4.4.2.- Programa de inferencia, explicación y aprendizaje .....	106
CONCLUSIONES .....	118
ANEXO 1	
Listado del programa de adquisición de conocimiento .....	120
ANEXO 2	
Listado del programa de inferencia, explicación y aprendizaje .....	128
BIBLIOGRAFIA .....	146
REFERENCIAS .....	158

## INTRODUCCION.

=====

Actualmente en nuestro país, específicamente desde mediados de 1984, la idea de utilizar a la computadora, como auxiliar en el proceso de enseñanza-aprendizaje, ha captado la atención tanto de instituciones educativas y de investigación como de particulares quienes vislumbran, cada uno desde su particular punto de vista, el uso de la computadora como un elemento de gran apoyo y utilidad en la educación.

De manera similar la aparición de los Sistema Expertos, como una realidad práctica y no sólo como sistemas experimentales, ha provocado gran expectación e interés. La idea de elaborar sistemas de computación basados en conocimiento y capaces de aprender resulta fascinante, siendo cada día más los trabajos desarrollados y las investigaciones referentes a este campo de la Inteligencia Artificial.

Tanto los sistemas educativos de computación como los sistemas de Inteligencia Artificial han convergido en un punto, generándose lo que se denomina Instrucción Inteligente Asistida por Computadora, también llamada ICAI (Intelligent Computer Assisted Instruction), en la cual se plantea, entre otras cosas, el uso de Sistemas Expertos para la educación. En nuestro país poco se ha escrito y no son muchas las referencias serias respecto a estos temas de actualidad he importancia.

El presente trabajo de investigación que presentamos pretende dar una idea clara respecto al uso de computadoras en la educación, así como de lo que es un Sistema Experto, cuáles son sus componentes y funciones y las características de los sistemas inteligentes de instrucción asistida por computadora. Podemos mencionar, como objetivos de este trabajo, los siguientes:

- a) Presentar un panorama general correspondiente al uso de la computadora como instrumento para el proceso de enseñanza-aprendizaje.
- b) Determinar lo que es un Sistema Experto, describiendo cada uno de sus componentes, sus funciones y la manera de desarrollarlos.
- c) Presentar las características, elementos, ventajas y desventajas de los sistemas de Inteligencia Artificial diseñados para la educación, entre los cuales se encuentran algunos Sistemas Expertos.

- d) Presentar un pequeño Sistema Experto de propósito general elaborado por el autor, planteando su utilización como elemento educativo.

## CAPITULO I

### LAS COMPUTADORAS EN LA EDUCACION.

=====  
Antes que nada, debemos de tener en cuenta los aspectos fundamentales del campo de la pedagogía, para así poder emitir un juicio coherente respecto al uso de las computadoras como auxiliares en la enseñanza. Comenzaremos diciendo que la pedagogía es una ciencia que intenta aplicar datos tanto de la sociología como de la psicología. Recibe sólo ayudas modestas de sus ciencias precursoras, pero le falta un adelanto suficiente respecto a estas disciplinas, dificultándose con eso una formación de un cuerpo de conocimientos específicos.

Educar, según Piaget, es "adaptar al niño al medio social adulto, es decir, transformar la constitución psicobiológica del individuo en función del conjunto de aquellas realidades colectivas a las que la conciencia común atribuye un cierto valor" (Piaget 1969).

Podría decirse, simplemente, que educar al niño es adaptarlo a su medio social ambiente. Según Piaget, la pregunta que resume el problema central de la educación es :

"Tiene el niño la capacidad para una actividad que es característica de las más altas conductas del adulto, como lo es la investigación continuada seguida de una necesidad espontánea ?"

Aparentemente el uso de las computadoras en la educación, sobre todo si nos referimos a las experiencias de Seymour Papert con LOGO, nos da una respuesta afirmativa a la pregunta anterior.

#### 1.1- LA INTELIGENCIA Y LAS OPERACIONES CONCRETAS Y FORMALES.

-----

Siguiendo con el punto de vista de Piaget, explicaremos

brevemente sus ideas referentes al conocimiento y al aprendizaje.

Conocer un objeto es operar sobre él, transformarlo, para así captar los mecanismos de estas transformaciones en relación con las acciones que se ejercieron para dichos cambios. "Conocer es asimilar lo real a estructuras de transformaciones, siendo éstas elaboradas por la inteligencia" (Piaget 1969).

Luego entonces, la inteligencia deriva de la acción y consiste en ejecutar y coordinar acciones transformadoras en forma interiorizada y reflexiva, o sea, en la mente del sujeto.

Estas "acciones interiorizadas" son los motores de todo razonamiento; son las operaciones lógico-matemáticas. Cuando el niño llega a los siete u ocho años, ya se presentan en él dichas acciones interiorizadas, pudiendo ser capaz de operaciones como las siguientes:

- Separaciones espaciales y desplazamientos ordenados, lo cual le da el concepto de medida.
- Origen de la seriación, lo cual le da el concepto de número.
- Encadenamiento de relaciones. (ej:  $A < B < C$ )
- Clasificaciones.
- Correspondencias.
- Reuniones y disociaciones de clase.

A estas operaciones se les denomina CONCRETAS, ya que se refieren simplemente a objetos.

Al llegar a los once y doce años, el niño conquista un nuevo modo de razonamiento referente ya no sólo a objetos, sino a hipótesis enunciadas verbalmente como proposiciones. A estas operaciones se les llama FORMALES y tienen, de manera general, dos características:

- 1.- Cada operación formal tiene una inversa y una recíproca.
- 2.- Implican una combinatoria la cual es aplicada a ideas y objetos.

Se tienen operaciones como:

- Si A entonces B.

- Disyuntivas ( A o B ).
- Conjunciones ( A y B ).
- Negaciones ( NO ( A ) ).
- Desigualdades ( A <> B ).

Posteriormente, entre los doce y los quince años aparece, en el ahora adolescente, un espíritu experimental, tratando de agrupar los factores de un fenómeno complejo e ir variando éstos, uno por uno, de manera sistemática. Esto es debido a que, al pasar de las operaciones CONCRETAS a las FORMALES, el niño maneja operaciones hipotético-deductivas, siendo capaz de combinarlas y verificarlas a través de la experiencia. Gracias a esto el conocimiento se abstrae de los objetos operando sobre ellos, variando factores que los transforman, haciendo combinaciones, etc. Conocer no se trata entonces de obtener únicamente una copia figurativa del objeto, sino de trabajar con él.

Vemos entonces que la inteligencia empieza por ser práctica y, poco a poco, se va interiorizando a pensamiento. Lo importante es que su actividad es una continua CONSTRUCCION.

Comprender e inventar son las funciones primordiales de la inteligencia, construir estructuras, estructurar lo real. Para comprender un fenómeno hay que reconstruir toda una serie de transformaciones de las cuales éste ha sido resultado. Para ello hay que elaborar toda una estructura de transformaciones, lo cual implica una invención.

Piaget define a la inteligencia como "la adaptación por excelencia, el equilibrio entre la asimilación continua de las cosas a la propia actividad y el acomodo de esos esquemas asimiladores a los objetos". Es por todo esto que se ha dado tanto impulso a los METODOS ACTIVOS de enseñanza en los cuales, con respecto a el uso de la computadora, se encuentra el lenguaje LOGO y algunos programas CONSTRUCTORES, de los que hablaremos posteriormente.

## 1.2.- METODOS PASIVOS Y METODOS ACTIVOS.

---

En lo que a métodos de enseñanza se refiere puede decirse que hay, de manera general, dos corrientes (Pérez y Pérez 1984):

a.- CONDUCTISTA.

Basada en el concepto de Estímulo-Respuesta y cuyos personajes principales han sido Pavlov y Skinner.

A esta corriente pertenecen los métodos pasivos.

b.- COGNOSCIVISTA.

Basada en la participación del niño en un medio ambiente de libertad. De este punto de vista fueron Montessori y Piaget.

A esta corriente pertenecen los métodos activos.

El problema en la elección de uno de estos dos métodos de enseñanza debe plantearse de acuerdo a la disciplina que se quiera impartir:

- 1.- Materias cuya verdad no depende de acontecimientos determinados, sino de investigaciones y descubrimientos. (ej: Matemáticas).
- 2.- Materias cuyo contenido lo ha inventado el adulto y cuyos conceptos sólo se transmiten. (ej: Historia).

Ultimamente los métodos de enseñanza de disciplinas lógico-matemáticas han sufrido varias modificaciones, al contrario de lo que ha sucedido con las disciplinas literarias y las humanidades.

Si el objetivo de la educación es la formación de alumnos inteligentes, entonces existen carencias en los métodos pasivos de enseñanza tradicional, los cuales tienden mas a poblar la memoria del alumno y formar eruditos, no investigadores. Esto fue determinado, de una manera más formal, en una Conferencia Internacional de Instrucción Pública, donde el artículo 64 de la recomendación #49, a los ministerios, propone: " a fin de aumentar en los alumnos el interés en los estudios técnicos y científicos es conveniente, desde la escuela primaria, utilizar métodos activos apropiados para desarrollar en el alumno un espíritu experimental " (Piaget 1969).

### 1.3.- CAI Y LOGO.

-----

En 1924 el Dr. Sidney Pressey inventó una máquina para la clasificación de exámenes de opción múltiple, la cual fue probada y resultó efectiva en la enseñanza. Más tarde, el aprendizaje por medio de una máquina fue implementado y expandido por Skinner en Harvard, en la década de los cincuenta (Frenzel 1980).

Primeramente se tuvo la idea de Instrucción Programada, en donde el material didáctico es dividido en módulos, los cuales son presentados en un orden específico. Después de la presentación de un módulo, se hace una pregunta al alumno. Si la respuesta que se proporciona es correcta entonces se pasa a presentar el siguiente módulo, de lo contrario se presenta el módulo nuevamente, a manera de repaso.

Los primeros sistemas de instrucción programada presentaban únicamente textos. Posteriormente se aplicaron diapositivas y películas. Desgraciadamente, las máquinas para la enseñanza, en los años cincuenta y sesenta, no llegaron a ser populares, dadas las carencias de equipo y material (Frenzel 1980).

Al llegar las computadoras se tuvo un dispositivo con mayor flexibilidad para la Instrucción programada. Fue entonces que nació la CAI.

CAI es el término utilizado para la descripción del proceso de educación por computadora. Las siglas vienen de Computer Aided Instruction. En español se le denomina EAC ( Educación Asistida por Computadora). CAI es, en general, un sistema de instrucción individualizada que utiliza un programa de computadora como medio esencial para el aprendizaje. La idea central es usar a la máquina como una herramienta auxiliar, presentando textos y gráficas en una secuencia determinada e interactuando con el alumno a través de preguntas y respuestas.

Sin embargo, actualmente la CAI no ha sido utilizada de una manera más amplia. Esto se debe a razones como las siguientes:

a).- Utilizar a la computadora únicamente como un dispositivo de instrucción programada ha resultado relativamente caro, si pensamos que el mismo contenido puede presentarse en un libro, lo cual resulta mucho más económico.

b).- Implica una política de instrucción individualizada y no grupal.

c).- No ha demostrado, hasta ahora, ser mejor o peor a otros métodos de enseñanza programada como lo son audiovisuales, televideos, lecturas, etc.

d).- El tiempo dedicado a la programación de la máquina ha representado un costo adicional.

e).- La producción de este tipo de courseware implica una especialización en la materia por parte del programador.

f).- Dada la cantidad de computadoras diferentes, no se había tenido material estándar altamente comercial.

Por otro lado, los métodos basados en imágenes, tanto estáticas como dinámicas, no son del todo adecuados para la formación de constructivismo operatorio definido por Piaget. Esto es por que la inteligencia es comparable más bien "a un motor que asegura el desarrollo de las imágenes" o a "un mecanismo cibernético que asegura tal desarrollo gracias a una lógica interna y a procesos autorreguladores y autocorrectores" y no se reduce tan sólo a las imágenes de un filme.

A pesar de todo lo anterior, actualmente la situación de CAI puede tornarse diferente, dado el bajo costo en el que puede adquirirse un equipo, en relación a los años anteriores, así como la enorme producción de courseware y hasta de lenguajes especiales para producirlo, además de la compatibilidad de paquetes, elaborados generalmente para APPLE II, COMMODORE e IBM Pc.

Lo que es un hecho es que la computadora hace más atractiva la enseñanza. Aprender con la máquina resulta más divertido.

Podemos decir que CAI existe por ser una aplicación interesante y válida, y habrá interés en ella mientras las computadoras existan. Sin embargo, en la práctica, CAI permanece basada en el modelo educativo TRADICIONAL ( métodos pasivos ) ; se identifica un contenido didáctico específico y se presenta. La mayoría de los paquetes de software educativo se manejan de esta manera.

Segun Papert, " CAI es un método con el cual se programa al niño, no es el niño quien programa a la computadora ". En el medio ambiente activo de LOGO el niño conoce el manejo de la computadora y se mantiene en íntimo contacto con las ideas científicas y lógico-matemáticas.

Con LOGO se da un "aprendizaje por descubrimiento" pues el niño aprende mejor cuando es libre de descubrir relaciones por el mismo, más que cuando se le indica lo que debe aprender. Este tipo de aprendizaje produce cambios en el sujeto, tanto en la manera de pensar como de aprender (Papert 1980).

Los factores de recepción, la obediencia pasiva al adulto y la memoria son naturales en el niño, tanto como también su actividad espontánea. Por ello no puede decirse que los métodos pasivos hayan despreciado la observación del niño en este aspecto.

El niño va desarrollándose conquistando, mediante su esfuerzo y experiencia personal, la razón y las reglas de comportamiento. La sociedad espera, de las nuevas generaciones, un enriquecimiento, no una imitación.

Por último, mencionaremos la recomendación # 43 de la Conferencia Internacional de Instrucción Pública ( Oficina Internacional de Educación y UNESCO ), presentada en su sesión de 1956, donde en el artículo # 20 se propone (Piaget) :

a.- No confiar al automatismo más que las operaciones que ya se hayan asimilado.

b.- Antes de introducir el formalismo hay que asegurar la adquisición de las nociones y de los procesos operatorios.

c.- Más que proponerle al alumno un pensamiento adulto, ya elaborado, hay que conducirlo a formar las nociones y a descubrir por sí mismo las relaciones y propiedades matemáticas.

#### 1.4.- CAPACIDAD EDUCATIVA DE LAS MAQUINAS.

Basándose en el esquema Estímulo-Respuesta, los psicólogos norteamericanos han planteado varias teorías de aprendizaje, al igual que los psicólogos soviéticos. Ambas escuelas se han basado en este esquema propuesto por Pavlov.

Skinner, uno de los teóricos norteamericanos sobre aprendizaje más reconocidos, conciente del nivel rudimentario de los conocimientos neurológicos de su época y sintiendo inaccesibles los procesos intermedios en el aprendizaje, decidió considerar únicamente los estímulos y las respuestas. Esto dado que los estímulos pueden ser fácilmente manipulables a voluntad y a que las respuestas son posibles de analizar.

Liberando los obstáculos teóricos con sus generalizaciones y aplicaciones prácticas, Skinner observó que sus experimentos tenían mejores resultados en cuanto más era reemplazada la

intervención humana por dispositivos de tipo mecánico, bajo cierto control. De aquí se propuso que las máquinas para enseñar darían un rendimiento mejor al proporcionado por la enseñanza oral. Obviamente tales máquinas deberían de estar adecuadamente programadas.

Con las máquinas de Skinner todo castigo o sanción es eliminado, pues sólo se utilizan refuerzos positivos. Las experiencias fueron exitosas.

Sin embargo, y como ya se mencionó anteriormente, al ser éste un método pasivo de enseñanza, tuvo sus cuestionamientos y críticas. En Física y Matemáticas las ideas de Skinner no fueron aceptadas con mucho entusiasmo. Si el objetivo es que el alumno reinvente una serie de razonamientos, bien es que la máquina de Skinner no llega a excluir totalmente la comprensión ni el razonamiento mismo, sino que lo canaliza de una forma fastidiosa, excluyendo la iniciativa del sujeto.

En los casos en los que sólo se trate de adquirir conocimientos (ej: la enseñanza de idiomas, Geografía, Anatomía) es indudable que la máquina de Skinner puede prestar un buen servicio. Puede también pensarse en una máquina cuyo comportamiento sea Activo o Pasivo, variando de acuerdo a los conceptos de la rama del saber que se está enseñando. Las computadoras tienen esta flexibilidad.

El empleo de máquinas para el aprendizaje, de disciplinas que impliquen métodos pasivos, economiza tiempo lo cual permite al maestro aumentar las horas disponibles a trabajar activamente con los niños y en grupo. Al utilizar a las computadoras como máquinas para la enseñanza se ha visto que el modo tutorial, usando una instrucción individualizada, es más eficiente que el salón de clases. Podemos enunciar algunas de las más importantes posibilidades que brinda el uso de la computadora en la educación (Pérez y Pérez 1984):

- Interactividad.
- Ejercitación.
- Visualización.
- Evaluación.
- Preparación Intensiva.
- Apertura al futuro tecnológico.
- Flexibilidad de horario.

Actualmente se cuenta con una gran variedad de equipo periférico al computador, el cual ayuda considerablemente a su fácil manejo y a su uso para la instrucción. Se tienen elementos

como :

- Teclados.
- Pantallas "sensibles" al tacto.
- Joysticks.
- Mouses.
- Lápices digitalizadores.
- Alta resolución en gráficas a color.
- Graficadores en varios colores.
- Videodiscos.

También se han diseñado paquetes de courseware, los cuales constan de elementos específicos tanto en software como en hardware. Más tarde hablaremos sobre el software y los equipos disponibles. Todo esto es importante como argumento para responder a una pregunta que Piaget planteara: "El éxito en la transmisión de conocimientos depende tan sólo de la mejor o peor presentación, por parte del adulto, de lo que se desea inculcar en el niño?" (Piaget 1969).

### 1.5.- EL USO DE COMPUTADORAS EN LA ESCUELA.

---

Los beneficios mayores, al introducir a la computadora en la enseñanza, se dan cuando la tecnología es usada por el estudiante cuando éste desea llevar a cabo un aprendizaje de alto nivel, en el cual el alumno puede "conversar" con la computadora y desarrollarse una mejor manera de pensar a través de este proceso de interacción.

Aprender con las computadoras, ya sea utilizando métodos pasivos o activos implica, para el alumno, una mejoría en sus habilidades tanto en la escritura como en la lectura, así como un mejor desarrollo de su razonamiento lógico-matemático. Es por ello que los maestros deberán de conjuntar ambos métodos de enseñanza, pasivos y activos, tomando de cada uno lo mejor y utilizando a la máquina como una herramienta para implementar el método respectivo a la disciplina que se imparte.

La idea de usar a la computadora sólo como una herramienta y no como objeto de estudio es sumamente importante. Respecto a esto, Joseph Weizenbaum del MIT (y yo personalmente me uno a este punto de vista) dice que "enseñar BASIC a los niños es, desde el punto de vista pedagógico, una monstruosidad intelectual, la cual debe

empezar a erradicarse y no intentar utilizar dicho lenguaje como base introductoria para nada" (Weinzenbaum 1984).

Por otro lado, respecto al papel del maestro frente a la computadora incorporada en las escuelas, mucha gente cuestiona el hecho de la sustitución del profesor por la máquina. Este es un aspecto interesante, pues las computadoras están demostrando el carácter mecánico de la función del maestro de enseñanza tradicional. Simplemente, si no hubiera un cambio en los métodos de enseñanza, entonces podríamos decir que las computadoras podrían cumplir, satisfactoriamente, con el papel del maestro.

Según se ha visto, los niños se vuelven más entusiastas al utilizar a la computadora en el salón de clases, ayudándose unos a otros y trabajando bien sin la necesidad de una supervisión directa. Sin embargo dicha supervisión por parte del maestro, aunque indirecta, es muy necesaria e importante. Debemos tomar en cuenta que el profesor no sólo aporta los conocimientos de las materias impartidas, sino que también provee toda una serie de normas de conducta, valores éticos, disciplina, etcétera. Esta labor del maestro es importante pues, aunque no se ha demostrado con exactitud, se dice que el uso de la computadora suprime los valores afectivos. Lo importante es entonces establecer que la afectividad del maestro juega un papel favorable.

Por otro lado, es muy importante mencionar que el maestro debe dejar que el cambio a un ambiente computarizado ocurra paulatinamente, sin presionar al escolar. Para ello es importante conocer la idea que los niños tienen de la computadora, pues no debemos olvidar que son ellos quienes van a ser afectados por la introducción de esta tecnología en el ámbito educativo. Se ha sabido que la mayoría de los niños le adjudican a la máquina utilidades ilimitadas y que, mientras los más pequeños desean utilizarla para jugar, los mayores quieren usarla para conocer más del mundo que les rodea, como elemento de investigación (Chorover 1984).

#### 1.6.- PROBLEMAS PARA LA INTRODUCCION DE LAS COMPUTADORAS EN LA ----- EDUCACION. -----

Con todo lo anteriormente presentado, vemos que el problema de enseñar con la computadora no es trivial. En los Estados Unidos, según algunos educadores y publicistas que fueron entrevistados en 1980 al comenzar a entrar las microcomputadoras de lleno al

ámbito educativo, indicaron que muchos de los errores de la educación computarizada, en la etapa anterior al uso de micros, han sido reinventados y perpetuados. Entre estas ideas erróneas se encuentran principalmente dos:

- Pensar que la simple instrucción es la mejor manera de aprendizaje.
- Pensar que el papel de la tecnología es el de automatizar esa instrucción para que sea efectiva en costo.

Los educadores, en los Estados Unidos, que han visto en las tres décadas pasadas la introducción de la tecnología en la escuela, dicen " ahí vamos otra vez...", pensando que con la computadora sucederá lo mismo que con los elementos educativos anteriores que no tuvieron éxito y dejaron de ser utilizados ( Boyer 1984). Sin embargo no están en lo correcto, pues este tipo de tecnología es nueva y no existen precedentes reales de su uso. Con la experiencia hasta ahora obtenida, se ha visto que los mayores obstáculos para hacer un uso óptimo de la computadora en la escuela son los siguientes (Booner 1984):

- a) Casi no existe software educacional de alta calidad, tanto a nivel primaria como a nivel secundaria.
- b) Los maestros no han sido adecuadamente capacitados en el uso de la computadora y su integración en el plan de estudios.
- c) No hay hardware plenamente disponible.
- d) El hardware disponible no está distribuido equitativamente.

Para 1980 se postulaba que, para la existencia de una verdadera potencialidad de las microcomputadoras en el campo educativo, era necesario un esfuerzo masivo en la producción tanto de software como de hardware. Hasta la fecha, en el desarrollo de software se ha dado énfasis en los programas para el aprendizaje de lenguaje y de matemáticas básicas, dado que tales aplicaciones son las más solicitadas y las más vendidas, además de no implicar conocimientos profundos en la materia de parte del programador. Los autores de software educativo de calidad son contados, pues se trata de gente especializada tanto en la materia que se va a enseñar como en técnicas de programación.

En el Primer Simposio Internacional "La Computación en la Educación Infantil", en la ponencia " El transplante de software

educacional y el diseño de productos nacionales" se propusieron algunas bases para la producción de software educativo en México (Pérez y Pérez 1984). Son las siguientes:

- 1.- Debe ser tutorial.
- 2.- Debe ser abierto.  
(Usarse para ejemplificar en cualquier momento).
- 3.- Ameno.
- 4.- Esencialmente comprensivo.
- 5.- Creativo.
- 6.- Apegado al Plan Nacional de Educación.
- 7.- Operativo. (fácil de manejar).
- 8.- Dinámico.
- 9.- Evaluativo y estimulativo.

Yo agregaría que se debe suprimir las simulaciones de violencia, típicas de los juegos computarizados actuales.

De acuerdo a investigaciones hechas en norteamérica, en High-School la mayoría de los equipos de cómputo son utilizados para enseñar programación más que como herramienta de apoyo a las materias impartidas (Hasset 1984). En las escuelas elementales las máquinas son usadas, principalmente, para juegos didácticos. Teóricamente con el uso de la computadora se incrementan la motivación, la atención y el aprendizaje, pero prácticamente la mayoría del software educacional es pobremente diseñado.

En la Unión americana, la EPIE (Educational Products Information Exchange) ha hecho las evaluaciones de software más sistemáticas, abarcando a un 20% de las escuelas de ese país. Más de 7000 programas educativos, referentes a diferentes áreas, fueron analizados. Los resultados fueron desalentadores: sólo un 25% del software es elaborado con los mínimos estándares técnicos y de instrucción y tan sólo un 4% son considerados como productos excelentes (Hasset 1984). En muchos de ellos, el juego se convierte en el centro de atención y no los conceptos que se pretenden enseñar. Desgraciadamente la gente usa a las computadoras más para jugar que para otra cosa, pues puede así manejarlas de una manera sencilla.

Aparentemente, en varias escuelas de la Unión Americana, se ha decidido que el mejor uso educacional de las computadoras es el de enseñar computación a los alumnos. En otras escuelas, los maestros pretenden que se tenga todo un conjunto de material didáctico para proporcionar práctica y maestría al estudiante en el manejo del computador. La mayoría de estos maestros desean utilizar el courseware para darse más tiempo de conocer a los alumnos.

Casi todos los maestros de los Estados Unidos se quejan de que los paquetes de software educativo son pobres. Esto está dando pie a la proposición de una Comisión a Nivel Nacional, compuesta por maestros y gente de computación, para evaluar la calidad del software que se quiera implementar. Las carencias en los paquetes educativos de programación han decepcionado a los maestros. Afortunadamente la situación tiende a cambiar pues, al aumentar el número de computadoras personales en el mercado, ha aumentado la disponibilidad de courseware. Hemos planteado, como primer problema, la falta de software educativo de alta calidad. Hablemos ahora del problema de la capacitación del profesorado.

Afortunadamente existen esfuerzos para rectificar esta carencia. En México ya se han formado grupos para la capacitación del maestro en computación, y en Norteamérica han aparecido sistemas de software diseñados para familiarizar al maestro con la computadora. También las escuelas de Educación han comenzado a instituir, dentro de sus planes de estudio, la capacitación en cómputo. Es obvio que la mayoría de los maestros no pueden elaborar buen software, por ello es que éste escasea. A algunos otros se les dificulta enseñar utilizando a la computadora como herramienta. Por éstas razones es importante plantearnos la formación del profesorado así como el contenido de los cursos que se impartirán con el auxilio de la máquina. Para determinar la formación del maestro que utilice a la computadora, se propone el curriculum siguiente (Suzuky 1984):

- a.- Algorítmica.
- b.- Estructuras de Información.
- c.- Lenguajes de Programación (LOGO, PILOT, PLATO).
- d.- Sistemas Operativos, a nivel de usuario y a nivel de operación.
- e.- Operación de un equipo de cómputo.
- f.- Pedagogía.
- g.- Técnicas Didácticas.
- h.- La materia que se va a impartir.

Tal curriculum parece demasiado ambicioso y de hecho lo es. Particularmente, pienso que el enseñar a los niños conceptos elementales de computación no implica que la capacitación del maestro deba ser también elemental sino, por el contrario, para que todos los conceptos puedan ser explicados de una manera lo suficientemente clara y sin confusiones, el profesor debe comprenderlos completamente y dominarlos, así como poder manejar la computadora en cualquier momento en que sea necesario y finalmente, por que no, producir sistemas. Es necesario que se planteé todo un plan de actividades basadas en el uso de la

máquina en el salón de clases, así como evaluar y seleccionar el courseware que se piense implementar. Ambas actividades le corresponden a los maestros.

Con respecto al hardware es recomendable que, las escuelas que piensen en comprar algún equipo de cómputo, tomen más en cuenta la calidad de los programas que la calidad de la computadora que se les ofrece. El problema de la disponibilidad del hardware será cada vez menor al aumentar la producción de microcomputadoras y al disminuir su precio. En nuestro país un ejemplo de ello es la microcomputadora COMMODORE, producida masivamente, cuyo precio no es mayor a 40,000 pesos. A pesar de esto, una distribución equitativa de los equipos de cómputo será muy difícil en nuestra sociedad, pues existen aún muchas necesidades básicas y prioritarias por cubrir para el sector económicamente débil de nuestro país.

#### 1.7.- LA PUBLICIDAD DE EL USO DE LA COMPUTADORA.

---

En norteamérica, de 1981 a 1982, se duplicó el número de escuelas que utilizaban computadoras. Para el año siguiente la cifra volvió a duplicarse (Hasset 1984). También ha ido en aumento, considerablemente, el número de hogares donde ya se tiene un computador personal. Una de las razones por las cuales se está dando esta situación es simplemente por la necesidad, quizás creada y no real, de la presencia de la computadora en el aula, la cual se ha hecho, últimamente, más fuerte que el tener como meta la educación del alumno. Esto se debe a la publicidad que se hace de la computadora, a través de la cual se trata de convencer al adulto de que el uso de la máquina, por parte del niño, lo va a capacitar para el futuro.

Hay quienes argumentan que el uso de las computadoras en la escuela es un mercantilismo, con el cual se trata de convencer a los padres que la computación es algo necesario para sus hijos. Por otro lado, y en nuestro país se ha visto claramente, todo padre desea que sus hijos participen y se preparen para la "era de las computadoras".

En los Estados Unidos ha comenzado una reacción en contra del incremento de las computadoras. El Committee on Basic Skills Education propone que la mejoría en los planes de estudio y de materiales didácticos, tomando como meta el fortalecimiento del maestro y el énfasis en los elementos básicos del aprendizaje, debe lograrse antes de gastar el presupuesto en computadoras

(Hasset 1984).

En fin, lo cierto es que el número de máquinas aumenta día con día y desgraciadamente la gente tiene la creencia de que, para un futuro a corto plazo, se debe cultivar en la ciencia de la computación. Esto tiene sus ventajas, ya que se irá erradicando, poco a poco, la tecnofobia y el concepto erróneo que se tiene de la computadora. La mayoría tiene la idea de que, con ciertos conocimientos computacionales, se asegurará un buen trabajo y que, si no conoce nada al respecto, se verá marginado. La verdad es que al igual que el teléfono, la radio, la televisión, la videocasetera, etc, la computadora es, y seguirá siendo para la mayoría de la gente, una caja negra.

#### 1.8.- SOFTWARE Y EQUIPO.

-----

Al utilizar a las computadoras, algunos educadores han comenzado a incursionar y meterse de lleno en la computación. De la misma manera, los programadores comenzaron a comprender las necesidades de los maestros y alumnos en el ámbito de la educación. Esto ha sido muy bueno, pues los diseñadores de software deben estar pendientes de la carga cognoscitiva que puede absorber la mente del niño.

En contraste con la CAI tradicional, el software poderoso para el aprendizaje utiliza gráficas como elementos didácticos y no sólo para decorar la pantalla, así como también enseñar estrategias de aprendizaje y herramientas con las cuales el niño puede construir otras. El mejor software para el aprendizaje ofrece opciones, tales como editores de texto que permiten al niño crear sus propios juegos o la creación de gráficas originales (Piestrup 1984).

Algunos investigadores proponen que con medios electrónicos se aprende con mayor imaginación que con los textos y postulan que el software educativo no es un medio que complementa al texto, sino que debe ser algo independiente. Otros investigadores, quizás más realistas, proponen lo contrario. Generalmente, en las técnicas de CAI, se utiliza al computador con un complemento, ya sea películas, videos, cintas de audio, libros, etc.

En los inicios de 1970 fué desarrollado el sistema PLATO en computadoras CDC, en la Universidad de Illinois. PLATO es, por esencia, un sistema de CAI el cual utiliza capacidades de graficación y de terminales "sensibles al tacto" (Frenzel 1980).

La cantidad de courseware desarrollada para PLATO es grande. Al advenimiento de las microcomputadoras PLATO se adaptó a ellas, surgiendo el sistema MICRO-PLATO, ahora disponible en Apple, IBM PC, Atari 800, Commodore 64, TI 99/4A y CDC-110 (Gil 1984).

Una buena opción, para la producción de software educativo, es la "programación por ensayo" (Finzer y Gould 1984). Esta propone un medio ambiente visual de programación, enfocado a gente no experta en la elaboración de sistemas, para la creación de software educativo.

Se trata de un medio ambiente de programación en el cual se da énfasis en la programación visual; sólo cosas que pueden verse pueden ser manipuladas. El proceso de elaboración de un sistema educativo, utilizando esta metodología, es fácil y rápido, además de divertido. Se dice que un programa sencillo puede elaborarse en media hora.

El proceso de diseño y programación simula algo parecido a una obra de teatro. Consiste en el movimiento de "actores" (objetos) enseñándole a éstos a interactuar unos con otros mandándose "apuntes" alrededor de "escenarios".

El sistema está elaborado en SMALTALK-80, un lenguaje orientado a objetos. Esta idea se extendió para elaborar el sistema bajo la metáfora de ser un teatro con actores, acciones, animación y relaciones entre ellos. Todos los elementos que son utilizados son visibles en la pantalla. El manejo de los objetos se hace a través de un "mouse".

La experiencia, con diseñadores de sistemas, ha determinado que este medio ambiente es confiable para hacer accesible, a los no-programadores, la computadora como elemento de producción y presentación de material didáctico. El objetivo del sistema es eliminar el papel del programador convencional en la elaboración del courseware, dejando su lugar a los maestros.

Un elemento de software educativo importante son los "Conjuntos de juegos para el aprendizaje" (Piestrup 1984). Estos son una serie de programas estructurados, en donde las herramientas utilizadas y los elementos aprendidos por el niño en los primeros juegos del conjunto son utilizados como fundamento para los juegos posteriores. Estos conjuntos de juegos tienen la característica importante de guiar al niño a través de la experiencia del aprendizaje. \*

Otros paquetes muy interesantes son los "Constructores" (Piestrup 1984). Un "Constructor" es un programa con gráficas y animación, con el cual el niño, al interactuar con él, va uniendo partes para la construcción de algo. Se trata de gráficas funcionales. La invención y el razonamiento son fomentados con estos sistemas.

Los constructores proveen una metáfora al mundo real, con una geografía definida y consistente internamente, con elementos (generalmente imágenes a color) como unidades de construcción y con reglas de conexión entre ellos.

Se trata de sistemas en los cuales, al igual que en LOGO, se aplican los métodos activos de enseñanza en contraste con los programas ordinarios de CAI, en los cuales se requiere que el niño dé respuestas de una forma única y rígida. La gente que propone a los sistemas constructores indica las siguientes ventajas de su uso:

- No son punitivos.
- Desarrollan la persistencia.
- Dan al niño seguridad en sí mismo.
- Da al niño un sentimiento de maestría.
- Fomentan una habilidad creadora.

Anteriormente se tenía la carencia de software educativo en ciencias (Biología, Química, Física). Actualmente se ha hecho mucho en este sentido. Probablemente el ejemplo más importante es el del paquete de Atari Lab Series (O'Malley 1984), que conjunta elementos de software y de hardware, tales como medidores de temperatura, otros transductores y medidores, manuales de operación y software de simulación de procesos que, por razones de economía, no pueden elaborarse realmente en el laboratorio de la escuela.

Respecto a software educativo científico se tenían registrados en los Estados Unidos, para septiembre de 1984, 175 paquetes, con costos desde los 20 hasta los 250 dólares y abarcando materias como Biología, Química, Energía, Anatomía, Física, Ciencias de la Tierra, etcétera (O'Malley 1984).

Existe, por otra parte, un conjunto de sistemas educativos con una peculiaridad interesante: son sistemas "inteligentes". Se trata de sistemas que utilizan técnicas, tanto de programación como de manejo de datos que hacen presentar, a la computadora, como un tutor inteligente que va conociendo y aprendiendo los avances y limitaciones del estudiante. Estos sistemas educativos de Inteligencia Artificial representan una interesante y poderosa alternativa en el software educativo. De hecho son el aspecto a discutir en este trabajo. En el capítulo tercero se presentan los elementos y características de estos sistemas.

Para finalizar se presentan las máquinas más populares en los Estados Unidos, que utilizan software educativo, indicando un

tanto por ciento aproximado del total de paquetes que corren en ellas :

- 1.- Apple y Franklin.....82.85 %
- 2.- Commodore 64.....24 %
- 3.- Radio Shack.....21.71 %
- 4.- IBM Pc.....19.42 %
- 5.- Atari.....9.71 %
- 6.- Texas Instruments....0.57 %

En Francia el primer distribuidor de computadoras personales para las escuelas es Apple, con un número aproximado de 10,000 unidades. Se tienen máquinas francesas como la Thompson TD7 y M05. En la Gran Bretaña son tres las máquinas inglesas más populares: la BBC Acorn, la Sinclair ZX-Bi y la Link 480 ( Krivine 1984 ).

#### 1.9.- EL USO DE LAS COMPUTADORAS PARA LA EDUCACION EN MEXICO ----- Y OTROS PAISES. -----

Los fines de la educación deben ser determinados por la sociedad, ya que es ella misma quien la proporciona. El hombre, educado o no, es producto de ella.

En nuestro país, como en muchos otros, el número de computadoras utilizadas para la educación, tanto en la escuela como en el hogar, va en aumento día con día. En los Estados Unidos más de la mitad de las escuelas tienen, por lo menos, un equipo de cómputo. Los reportes del MDR (Market Data Retrieval) califican el crecimiento del uso de computadoras en la escuela como explosivo, duplicándose su número cada año. En los Estados Unidos, en escuelas públicas de nivel elemental y secundaria, el número de computadoras se elevó de 31,000 en 1981 hasta 325,000 en 1983. Esto lo reporta The National Center for Education Statistics (Booner 1984).

Se presuponen, actualmente, 90 alumnos por máquina. En nuestra sociedad, es de suponerse que una situación parecida se dé en los

próximos años.

Los gobiernos de varios países han planeado ya la incursión de las computadoras en el aula. El gobierno francés, por ejemplo, ha iniciado varios proyectos. Uno de ellos es el proyecto DIANE, a cargo de L' Agence de L' Informatique, produciendo software educativo el cual, desafortunadamente, solo corre en tres marcas de computadoras francesas.

Por otro lado, la primer ministro de Gran Bretaña, Margaret Thatcher, hace imperativa la educación computarizada, proclamando que todas las escuelas superiores contarán con equipos de computación para finales de 1983 y todas las primarias para fines de 1984 (Krivine 1984).

En nuestro país la educación está normada por un Plan Nacional de Educación. Cualquier herramienta que quiera ser utilizada para el fortalecimiento de la enseñanza, en este caso la computadora, debe estar acorde con dicho plan, para no provocar confusiones. De manera general la descentralización del Sistema Educativo Nacional y los lineamientos oficiales plantean nuevas posibilidades, tanto para los alumnos como para los educadores.

Se tienen tres situaciones en la educación:

- a) El aumento de los alumnos.
- b) La necesidad de reclutar maestros debidamente capacitados.
- c) Las necesidades económicas, científicas y tecnológicas de nuestra sociedad, en la cual se organiza la instrucción pública.

Respecto al uso de computadoras en la escuela, éste se encuentra limitado por:

- a) La disponibilidad de equipos.
- b) Software inadecuado e inaccesible.
- c) La capacitación de los maestros.
- d) La competencia, de parte de la sección administrativa de la escuela, en el uso de la máquina.

Son necesarias, entonces, estrategias para la incorporación de la computadora en el medio educativo, que permitan obtener los mejores resultados posibles. Estas estrategias deben ser acordes a nuestra realidad nacional, tomando muy en cuenta el uso racional del equipo de cómputo disponible.

Para que la implantación de las computadoras se lleve a cabo de manera satisfactoria es necesario, entre otros aspectos, el capacitar a los maestros.

#### 1.10.- EL PROBLEMA DEL PROFESORADO.

---

El problema del profesorado, tanto en nuestro país como en otros, es complejo. La profesión de educador no ha podido alcanzar un status en nuestra sociedad. Los maestros de primaria y secundaria no son considerados, a veces ni por ellos mismos, como especialistas, sino como transmisores de un conocimiento al alcance de cualquier persona. Desgraciadamente carecen de un prestigio intelectual (Piaget 1969).

La preparación del profesorado, a estos niveles, no guarda relación con el ambiente universitario. Piaget decía:

"El regimen tradicional ha contribuido a hacer al cuerpo de enseñanza primaria una especie de clase intelectual replegada sobre sí misma, privada de las valoraciones sociales a las que tiene derecho por su alejamiento, tanto de las corrientes científicas, como de una atmósfera de trabajo experimental".

Por una parte, la autonomía del maestro es limitada. Por otro lado, la labor del maestro se vuelve más dura cuando se implantan innovaciones en las escuelas, así como en cuanto son mejores los métodos de enseñanza mas difícil es el aplicarlos.

Puede pensarse que, generalmente, lo automatizado provoca un desplazo de trabajo y una alienación. Que razones podrían tener los maestros para creer que esto no sucederá con la incursión de las computadoras en la escuela?

Los profesores deben estar conscientes de que es inobjetable el hecho de que la presencia del maestro en el aula es necesaria, y que el uso de una máquina, como auxiliar, les será muy útil en varios aspectos, entre ellos :

- a.- Tendrán mas tiempo para dedicarse, de manera mas personal, a sus alumnos.

b.- Se capacitarán en el uso de computadoras lo cual les permitirá, a los que lo deseen, aplicar la computación en otras áreas y en beneficio personal.

La capacitación del profesorado en computación le ayudará a obtener, socialmente, el status que siempre le debe haber correspondido.

Definitivamente, el maestro no debe ser desplazado, pues su papel de educador no se limita únicamente a las materias que se imparten en la escuela, sino que es de suma importancia la enseñanza de un comportamiento correcto, de valores morales y éticos, de convivencia y cooperación, etcétera.

## CAPITULO II

### ELEMENTOS Y CARACTERISTICAS DE LOS SISTEMAS EXPERTOS.

El conocimiento del hombre respecto a la realidad que le rodea y el cómo dominarla ha sido almacenado, a lo largo de su historia, de manera pasiva, ya sea por medio de la escritura o de la palabra hablada.

Se han escrito una infinidad de libros, en los cuales el hombre ha impreso sus conocimientos y experiencias, con el fin de que la humanidad pueda resolver los problemas que se plantea y que se le presentan en sus actividades.

Con la aparición de la computadora y, actualmente, con el alto desarrollo tecnológico y las investigaciones en el campo de la Inteligencia Artificial, han aparecido los Sistemas Basados en Conocimiento. Con esto, el conocimiento del hombre, así como su experiencia, son almacenados electrónicamente para ser manejados de una manera activa y utilizados para la solución de problemas y obtención de nuevos conocimientos. Algunos de estos sistemas ya proporcionan, actualmente, beneficios prácticos y rentables para la humanidad.

El número de personas, capacitadas para el desarrollo de estos Sistemas basados en Conocimiento, va en aumento día con día. En los Estados Unidos de América las grandes compañías han creado ya sus propios programas de investigación, mientras que otras utilizan los servicios de Ingenieros en Conocimiento para el desarrollo de Sistemas Expertos (Kinnucan 1984).

Un Sistema Experto puede definirse, de una manera muy breve, como un sistema para la solución de problemas que, adquiriendo cierta experiencia, puede llegar a un nivel de comportamiento comparable al de un experto humano, exhibiendo conocimientos en un área limitada del saber.

La finalidad de la mayoría de los Sistemas Expertos es, primordialmente, la solución de problemas en un campo específico, por lo que la máquina necesita "conocer" lo que los humanos expertos saben al respecto. Los Sistemas Expertos son muy útiles en problemas a cuya solución no puede llegarse por medio de algoritmos, por lo que se utilizan técnicas como las heurísticas, etcétera. Los Sistemas Expertos han demostrado, a lo largo de su

corta vida, que se les puede tener la misma confianza que a un experto de carne y hueso.

El área de investigación de los Sistemas Expertos nació con el uso de metodologías desarrolladas en Stanford, con la aparición del primer Sistema Experto: DENDRAL.

Con respecto a los expertos humanos, los Sistemas Expertos presentan las siguientes ventajas:

- Son incansables.
- Son inmortales.
- Son mejorables ( a través de mantenimiento y expansión ).
- Pueden reproducirse indefinidamente.
- Son imparciales.
- Pueden explicar sus razonamientos.

Por otro lado, los Sistemas Expertos son incapaces de concebir necesidades, querencias, limitaciones, así como el comportamiento de los usuarios. Podría decirse que esto limita a un "desempeño inteligente" por parte del sistema. Sin embargo, no debemos perder de vista que LA INTELIGENCIA NO HACE AL EXPERTO, SINO LA EXPERIENCIA.

También el costo de producir y desarrollar un Sistema Experto es alto, respecto a la elaboración de programas convencionales. Actualmente han comenzado a aparecer los primeros Sistemas Expertos implementados en Computadoras Personales.

## 2.1.- LA INTELIGENCIA ARTIFICIAL.

-----

La Inteligencia Artificial ha sido la rama de mayor controversia en la ciencia de la computación. Los Sistemas Expertos son producto de las investigaciones en la rama de Inteligencia Artificial.

Según Winston, la Inteligencia artificial es "el estudio de las ideas que hacen capaz a la computadora de hacer cosas aparentemente inteligentes". También enfatiza que "los objetivos centrales de la Inteligencia Artificial son el hacer a las computadoras más útiles, así como comprender los principios que

hacen posible la inteligencia" (Winston 1977).

Se dice que un sistema es inteligente si puede adaptarse a nuevas situaciones por sí mismo, teniendo la capacidad de razonar, entender relaciones entre hechos, descubrir significados y reconocer lo que es verdadero (Findler 1976). Además de todo lo anterior, actualmente uno espera que un sistema que es inteligente sea capaz de aprender, mejorando su nivel de desempeño en base a la experiencia.

La definición del término "inteligencia" es un punto a discusión. No se hará ningún planteamiento formal respecto a lo que es la inteligencia, bastará con mencionar que "es la capacidad de seleccionar, de un número dado de alternativas, la más conveniente, de acuerdo a la situación actual". Es una "capacidad de autonomía y voluntad" y se caracteriza por "su amplitud y lo adecuado de su reacción frente a situaciones anómalas o imprevistas" (Guzmán 1984).

El campo abarcado por la Inteligencia Artificial puede dividirse en dos grandes ramas ( Findler 1976) :

#### 1.- Ingeniería.

Elaboración de sistemas sin tomar como base los métodos y técnicas utilizadas por el ser humano para la solución de problemas.

Ejemplos: Reconocimiento de patrones, traducción de textos de un idioma a otro, composición musical, etc.

#### 2.- Modelado.

Sistemas elaborados en base a la comprensión interna de los mecanismos de un sistema real (p.ej. humano), pudiendo explicar y predecir su comportamiento.

Ejemplos: Toma de decisiones, comportamiento de aprendizaje, modelado de redes neuronales, etc.

Puede hablarse de tres problemas fundamentales en la Inteligencia Artificial:

##### a.- La Representación.

- Selección y diseño del modelo.
- Transformación de una tarea particular en estructuras y procesos de información para un desempeño eficiente y efectivo.

##### b.- El conflicto entre Generalidad y Eficiencia.

- A mayor rango de aplicación de un sistema, mayor será

- su complejidad, siendo menor su eficiencia.
- Uno de los objetivos es hacer sistemas altamente eficientes y con un amplio rango de aplicación.

c.- La búsqueda de Soluciones.

- Cuando se presentan problemas complejos, el número de posibles soluciones puede ser muy grande, no siendo posible el efectuar una búsqueda exhaustiva.
- Se utilizan reglas heurísticas para el reconocimiento del conjunto de soluciones, disminuyendo así el tiempo de procesamiento y el espacio en memoria utilizado.

Dentro de las investigaciones y trabajos, desarrollados en Inteligencia Artificial, se encuentran:

- Juegos.
- Reconocimiento de formas.
- Robótica.
- Procesamiento de lenguaje natural.
- Aprendizaje e inferencia inductiva.
- Demostración de teoremas.
- Representación de conocimiento.
- Sistemas Expertos.
- Ingeniería en conocimiento.

Al comenzar nuestra década los sistemas desarrollados en Inteligencia Artificial, particularmente los de Ingeniería en conocimiento, han madurado, de tal manera que actualmente se tienen aplicaciones productivas, tanto gubernamentales como comerciales.

Uno de los textos clásicos en Inteligencia Artificial es (Winston 1977), aunque se tienen textos mas completos y actualizados como en (Rich 1983).

## 2.2.- DEFINICION DE SISTEMA EXPERTO.

-----

Podemos decir que las características de un experto de carne y hueso son las siguientes :

- Tiene conocimientos profundos en su materia.
- Tiene cierta experiencia en la solución de problemas referen-

tes a dicha materia.

- Llega a conclusiones, a través de razonamientos, utilizando el conocimiento que ha adquirido.
- Puede dar una explicación de dichas conclusiones, así como de su comportamiento.

Segun Guzmán Arenas, un Sistema Experto es un "programa que exhibe conocimientos profundos en un área limitada del saber humano" (Guzmán 1984).

Podemos decir que un Sistema Experto:

- a.- Contiene un conjunto de datos para la solución del problema.
- b.- Cuenta con el conocimiento correspondiente al área de aplicación, en la forma de un conjunto de reglas.
- c.- Cuenta con mecanismos para inferir, deducir, preguntar y explicar.

Por otro lado, para la definición de lo que es un Sistema Experto en particular, se determinan las siguientes características (Brachman 1983b):

- 1.- Inteligencia.  
Métodos de razonamiento y conocimiento de los principios fundamentales de un área de aplicación.
- 2.- Razonamiento por manejo de símbolos.
- 3.- Razonamiento acerca de sí mismo, especialmente para dar una explicación a sus conclusiones.
- 4.- Destreza.  
Alto desempeño y búsquedas eficientes.
- 5.- Dificultad y complejidad.
- 6.- Reformulación.  
Conversión de un concepto, de términos "seculares" a una forma manipulable por las reglas del sistema.
- 7.- Tipo de tarea.

Estas siete características son fundamentales para que el Sistema Experto cumpla con sus objetivos.

Podemos resumir que un Sistema Experto consiste de:

- a) Una estructura de datos, en la cual se representa la descripción del problema de una manera apropiada para su procesamiento. A esto se le denomina también como el contexto del sistema.
- b) Un conjunto de reglas, específicas de la materia de la cual el sistema es experto. Estas reglas pueden trabajar sobre la estructura de datos.
- c) Métodos de razonamiento, algunas veces utilizados cuando las reglas fallan o cuando el sistema requiere de razonar acerca de su propio conocimiento y carencias, con el fin de construir y reconstruir sus rutas de inferencia, para llegar a una conclusión o a una justificación, respectivamente.

### 2.3.- DIFERENCIAS ENTRE LOS SISTEMAS EXPERTOS Y OTROS SISTEMAS.

-----

La característica principal, que diferencia a los Sistemas Expertos de los sistemas convencionales, es la independencia existente entre:

- El modelo de solución del problema.
- El código del programa.

El modelo de solución del problema es una entidad separada, en la cual se encuentra el conocimiento de la materia de la disciplina para la cual el sistema fue diseñado. Este conocimiento se encuentra como un conjunto de reglas programadas. A esta entidad se le denomina Base de Conocimientos.

Existen muchos sistemas que, utilizando archivos convencionales, pueden proporcionar cierta información en base a consultas proporcionadas por el usuario, no conociendo la información que manejan, sino que sólo la analizan y la presentan. A este tipo de sistemas se les denomina Extractores de información.

Otros sistemas, llamados Especialistas ad-hoc, son diseñados

específicamente para la solución de problemas referentes a una disciplina en particular, pero el conocimiento sobre ésta se encuentra formando parte del código del resto del programa, por lo que un cambio en el conocimiento implica, forzosamente, un cambio en el programa (Guzmán 1984).

Respecto a otros sistemas de Inteligencia Artificial, los Sistemas Expertos tienen, con ellos, las siguientes diferencias (Hayes-Roth 1984):

- a.- Hacen énfasis en la solución de problemas de tipo específico, a diferencia de otros sistemas con métodos generales un tanto débiles.
- b.- Tienen un buen desempeño, a nivel de un experto, en tareas complejas.
- c.- Utilizan su propio conocimiento para razonamientos e inferencias.

También existen diferencias entre lo que es un Sistema Experto y un Sistema de Conocimiento. Dentro del campo de la Ingeniería en Conocimiento, los sistemas que se elaboran pueden clasificarse, de manera general, en dos grupos:

- 1.- Sistemas que se comportan como expertos humanos.
- 2.- Sistemas que elaboran tareas que requieren de cierta inteligencia, pero no necesariamente de experiencia.

En los Sistemas Expertos el conocimiento va creciendo gradualmente, conforme el sistema va interactuando, simulando así a un experto de carne y hueso. Por otra parte, los Sistemas de Conocimiento no necesariamente simulan a un experto humano.

En resumen, los Sistemas Expertos corresponden a una clase especial de los Sistemas de Conocimiento. Debemos referirnos a estos últimos como una categoría general, de la cual los Sistemas Expertos forman parte (Hayes-Roth 1984).

## 2.4.- FUNCIONES DE LOS SISTEMAS EXPERTOS.

---

Hemos definido lo que es un Sistema Experto, indicando sus componentes y características principales. Para ello hemos mencionado, de manera general, algunas de sus funciones.

Un Sistema Experto hace deducciones, inferencias y preguntas. Sin embargo, de una manera más específica, debemos mencionar que los Sistemas Expertos cumplen con ciertas características funcionales, indicadas anteriormente y que comentaremos a continuación (Brachman 1983b):

### a) Buen desempeño.

Un Sistema Experto se diseña para la solución de problemas en un campo específico. Su desempeño puede ser menor si el problema a resolver se aleja de ese campo de acción. El objetivo de un Sistema Experto es lograr el mayor nivel de desempeño posible.

### b) Búsqueda eficiente de soluciones.

Ante un problema, el Sistema Experto genera todo un conjunto de posibles soluciones al cual, la mayoría de las veces, es extenso. Por ello es necesario eliminar y decidir, inteligentemente, entre las hipótesis y posibilidades, para un funcionamiento óptimo. El sistema debe trabajar eficientemente y proporcionar resultados en un tiempo razonable. No puede perderse tomando en cuenta todas y cada una de las alternativas que se le presenten.

### c) Razonamiento por manejo de símbolos.

Los Sistemas Expertos usan un razonamiento simbólico, en base a una representación simbólica explícita de la realidad. El conocimiento puede representarse en un sistema simbólico, por lo que la acción de conocer consistirá, en gran parte, en la representación simbólica de objetos y relaciones referentes a la realidad.

### d) Razonamiento acerca del sistema mismo.

Se requiere que el Sistema Experto razone acerca de sus propios procesos y conclusiones. Puede analizar toda una cadena de razonamientos y comprenderla, produciendo una justificación y una explicación de su comportamiento. Una explicación implica la habilidad de ensamblar las inferencias que se han hecho. En algunos Sistemas Expertos esta habilidad de explicación no es

requerida.

e) Solución de problemas generales dentro de un dominio específico.

De acuerdo a lo extenso que sea el conjunto de principios y fundamentos con que cuenta el sistema, se determina un "ancho de cobertura". El sistema puede tener un conjunto de principios básicos o elementales, así como métodos de razonamiento de propósito general. Algunos investigadores desarrollan lo que se denominaría una "segunda generación" de Sistemas Expertos, los cuales utilizarán "primeros principios" respecto al comportamiento y leyes de la naturaleza. A estos sistemas también se les denomina Sistemas de Conocimiento Profundo (Kinnucan 1984).

f) Reformulación.

Generalmente un problema se encuentra en un estado inicial y arbitrario, por lo cual el Sistema Experto debe convertirlo a una forma apropiada para que sus reglas puedan entrar en acción.

g) Otras funciones.

Otras características de los Sistemas Expertos son:

- Capacidad de razonamiento por analogía.
- "Sentido común".
- Adquisición de conocimiento.
- Aprendizaje a través de su interacción con el usuario.

Estas dos últimas características son sumamente importantes, y se tratarán posteriormente a detalle.

## 2.5.- APLICACIONES.

-----

Los Sistemas Expertos tienen, potencialmente, una enorme conjunto de campos de aplicación. Presentamos algunas aplicaciones en el campo de la tecnología:

- Ciencia.
- Medicina, generalmente en diagnósticos.
- Finanzas.
- Configuración de equipos de cómputo.
- Exploración de minerales y petróleo.
- Aplicaciones militares.

- Reconocimiento de voz e imagen.
- Educación.

De acuerdo a sus aplicaciones, se pueden determinar ciertos tipos de Sistemas Expertos (Hayes-Roth 1983), los cuales presentamos a continuación a manera de categorías genéricas:

**a.- Interpretación.**

Infieren descripciones de situaciones proporcionadas. Se le da un significado simbólico a los datos.

**b.- Predicción.**

Infieren consecuencias de situaciones proporcionadas, generando un conjunto de situaciones posibles. Ejemplos: predicciones demográficas, cosechas estimadas, etc.

**c.- Diagnóstico.**

Infieren el mal funcionamiento de un sistema, ya sea médico, mecánico, electrónico, etcétera. Para ello se utilizan tablas asociativas entre el comportamiento de dicho sistema y ciertos diagnósticos posibles.

**d.- Diseño.**

Configuran formas y modelos bajo ciertas restricciones. Ejemplos: diseño de circuitos electrónicos, de edificios, etc.

**e.- Planeación.**

Configuran rutas de comunicación o transporte, hacen una programación automática de eventos, planes militares, proyectos, etc.

**f.- Monitoreo.**

Comparan la situación proporcionada con posibles acontecimientos, tomando en cuenta una cierta vulnerabilidad posible en el campo de acción. Son utilizados, por ejemplo, en las plantas nucleares, tráfico aéreo, manejos fiscales, etc.

**g.- Reparación.**

Administran y ejecutan planes para remediar algún problema. Son utilizados en el mantenimiento de automódviles, aviones, redes y sistemas de cómputo.

#### **h.- Instrucción.**

Se diagnostican las carencias del estudiante al interactuar éste con el sistema. Inician con una descripción de lo que el estudiante deberá aprender. Evalúan, diagnostican y proponen remedios a través de una interacción tutorial.

#### **i.- Control.**

Supervisan y gobiernan el comportamiento de un sistema, diagnosticando, monitoreando, prediciendo y planeando acciones.

### **2.6.- CONFIGURACION GENERAL.**

-----

Podemos clasificar en dos grupos a los componentes de un Sistema Experto, respecto a la función que desempeñan cada uno de ellos:

- Componentes de organización, representación y almacenamiento de conocimiento.
- Componentes de interacción y comunicación.

En lo que se refiere al primer grupo, tenemos como componentes:

- A.- Base de Conocimientos.
- B.- Máquina de Inferencia.

La Máquina de Inferencia es el elemento de control del sistema. En ella se interpretan las reglas contenidas en la Base de Conocimientos. La Máquina de Inferencia está programada para el procesamiento de símbolos, los cuales representan conceptos y relaciones.

La Base de Conocimientos contiene el saber específico en la disciplina de la cual el sistema es experto. Consiste en un conjunto de hechos (datos) y de reglas programadas. No contiene información específica de un problema en particular.

Otro componente, también correspondiente al primer grupo de clasificación, es el

### C.- Espacio de Trabajo.

En el Espacio de Trabajo se almacenan resultados, hipótesis y decisiones intermedios, así como el estado en que el problema se encuentra en un momento dado. En él se tienen las posibles soluciones al problema que se está resolviendo, así como las reglas de la Base de Conocimientos que han sido escogidas para ser aplicadas sobre los datos.

En adición a estos tres componentes, se tienen otros elementos importantes en la configuración de un Sistema Experto, concernientes al segundo grupo cuyo objetivo es ayudar a la comunicación e interacción del sistema con el mundo exterior. Estos son:

D.- Interfaz con el Usuario.

E.- Subsistema de Adquisición de Conocimiento.

F.- Subsistema de Explicación.

La interfaz con el Usuario es el componente de comunicación entre el Sistema Experto y las personas que lo utilizarán. Generalmente se trata de un procesador de lenguaje natural.

El Subsistema de Adquisición de Conocimiento tiene, como función, la de extraer el conocimiento de un experto humano o de cualquier otra fuente de experiencia, transfiriéndolo a la Base de Conocimientos en forma de reglas. Esta tarea puede efectuarla el Ingeniero en Conocimiento.

Finalmente, mencionamos al Subsistema de Explicación, el cual encadena una serie de inferencias que han llevado al sistema a alguna conclusión, con el fin de presentar una justificación de su comportamiento.

Presentamos gráficamente (fig. No. 1) la configuración correspondiente a un Sistema Experto ideal. Las flechas del diagrama indican el flujo del conocimiento a través del sistema.

A continuación se analizarán, con mayor detalle, cada uno de los seis componentes mencionados.

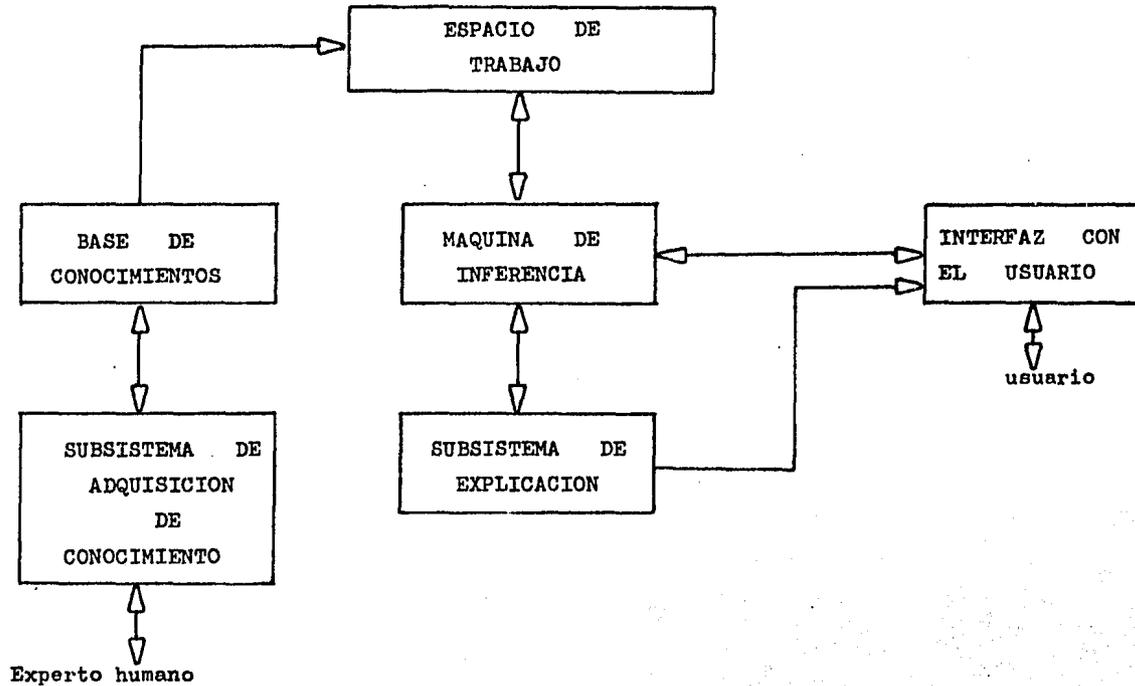


FIGURA No. 1

Configuración de un Sistema Experto ideal.

## 2.7.- LA BASE DE CONOCIMIENTOS.

---

Una de las características de los Sistemas Expertos es la independencia existente entre el programa y el conocimiento de la disciplina correspondiente, el cual se encuentra codificado en un conjunto de reglas y conceptos. Este conjunto es la Base de Conocimientos del sistema.

La Base de Conocimientos es una entidad dinámica. Va cambiando con el tiempo, durante el desarrollo del Sistema Experto, simulando el proceso de acumulación de experiencia. Aquí podemos puntualizar una peculiaridad muy importante de los Sistemas Expertos: son capaces de aprender. La Base de Conocimientos se va expandiendo y modificando, con la finalidad de convertir al sistema en un experto.

El conocimiento es el factor CLAVE para el desempeño de un Sistema Experto, y puede clasificarse en dos categorías (Feigenbaum y Mc.Corduck, 1983) :

a.- Conocimiento de conceptos.

Es el denominado "conocimiento compartido", el que se encuentra en los libros de texto, en artículos, etc.

b.- Conocimiento heurístico.

Es el producto de la práctica y la experiencia, el que se encuentra en la mente del experto humano.

Los elementos de una Base de Conocimientos son dos (Hayes-Roth 1983) :

1.- Conceptos.

2.- Reglas.

### 2.7.1.- LOS CONCEPTOS.

Los conceptos de la Base de Conocimientos son los datos que se han obtenido y almacenado a lo largo del desarrollo del sistema.

Incluyen descripciones de objetos, relaciones, valores numéricos correspondientes a probabilidades, valores booleanos correspondientes a la veracidad o falsedad de proposiciones, así como grados de confianza de las reglas.

Los conceptos, de la base de conocimientos, son descripciones de objetos y relaciones entre ellos, con los cuales se generan las reglas. Los conceptos son representados de manera declarativa y no por procedimientos, como lo es en el caso de algunas reglas.

Existen varias técnicas de representación declarativa de conocimiento (Kobsa 1984), con las cuales pueden estructurarse los datos del Sistema Experto. Estas son:

- 1.- Predicados Lógicos de Primer Orden.
- 2.- Marcos.
- 3.- Redes Semánticas.

Con el uso de fórmulas, en la lógica de predicados de primer orden, es posible representar conceptos y relaciones. Los predicados de primer orden son una herramienta fundamentada formalmente en el cálculo proposicional de la lógica matemática.

La representación se tiene utilizando un predicado con sus argumentos correspondientes, por ejemplo, el concepto "Juan ayuda a Pedro" se representaría:

AYUDAR ( Juan , Pedro )

Pueden tenerse predicados mucho más complejos y generales. Presentamos algunos ejemplos (Dhal 1983):

Si en PROLOG tenemos las descripciones:

AYUDA ( Juan , Pedro )  
AYUDA ( HERMANO ( X ) , X )  
AYUDA ( Pedro , X ) SI AYUDA ( X , Pedro )  
AYUDA ( Pedro , Pepe )

podríamos hacer las siguientes consultas:

AYUDA ( Pedro, Juan ) ?

R: YES

AYUDA ( Z , Pedro ) ?

R: Z = Juan  
Z = HERMANO ( Pedro )

AYUDA ( Pedro , Z )

R: Z = Juan  
Z = HERMANO ( Pedro )  
Z = Pepe

Podríamos preguntar "quién ayuda y es ayudado por Pedro ?"

AYUDA ( Z , Pedro ) AND AYUDA ( Pedro , Z ) ?

R: Z = Juan  
Z = HERMANO ( Pedro )

Finalmente la pregunta : "quién ayuda a quién ?"

AYUDA ( Z , Y ) ?

R: Z = Juan	Y = Pedro
Z = HERMANO ( Pedro )	Y = Pedro
Z = Pedro	Y = Juan
Z = Pedro	Y = HERMANO ( Pedro )
Z = Pedro	Y = Pepe

Los marcos son estructuras de datos en las cuales se almacena toda la información de un objeto o situación en particular (Nau 1983).

Un marco es un prototipo con el cual se representan objetos y situaciones, asignándole los datos correspondientes. Se usan marcos diferentes para objetos y situaciones de diferente categoría.

Podríamos tener una analogía, comparando un marco con una forma impresa en la cual, bajo una estructura dada, hay que llenar ciertos campos de datos. Para el concepto "Juan ayuda a Pablo" podría tenerse el siguiente marco:

NOMBRE DEL MARCO : ayudar  
TIPO DE MARCO : laboral  
SUJETO : Juan  
AFECTADO : Pedro

Podemos citar otros ejemplos, utilizados en (Barstow y varios 1983) para ejemplificar la representación de conocimiento con RLL. RLL es un sistema para el diseño, construcción y uso de Sistemas expertos. Los marcos siguientes corresponden a la descripción de un sistema de alcantarillado:

M6-3

ES: alcantarilla  
ALIMENTAHACIA M6-2  
LOCALIZADABAJO (MainStreet and OakAvenue)

M6-2

ES: alcantarilla  
ALIMENTADE (M6-3)  
LOCALIZADABAJO (MainStreet and RidgeRoad)

alcantarilla

EB: conjunto  
GENERALIDADES fisico, circular, artefacto  
EJEMPLOS (M6-2, M6-3)  
DESCRIPCION Representa la coleccion de todas las alcantarillas.  
PROTOTIPO alcantarilla tipica  
NUEVOS CAMPOS ALIMENTAHACIA, ALIMENTADE, LOCALIZADABAJO.

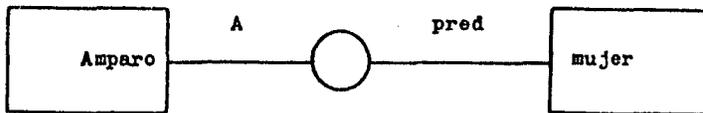
Un escrito clásico, en lo que a marcos se refiere, es el de Marvin Minsky, el cual se encuentra en (Winston 1975), (ver bibliografía, inciso b). El aspecto teórico de los marcos, usos y ventajas, es también presentado en (Winston 1977).

Finalmente, mencionaremos a las redes semánticas, las cuales pueden ser utilizadas en la representación declarativa de conocimiento. Los nodos de la red serán los objetos, mientras que las ligas representarán las relaciones entre ellos. Se trata de estructuras reticulares para la descripción de conceptos.

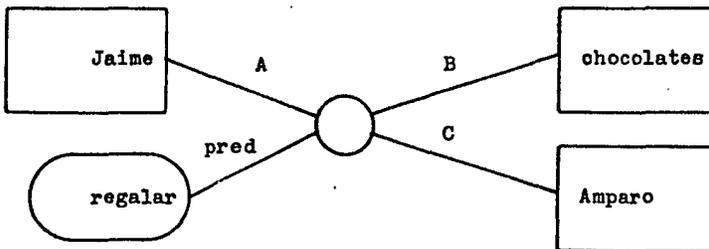
Los nodos pueden tomar un significado genérico (por ejemplo: HUMANO), o también individual (por ejemplo: Juan).

Las redes semánticas se caracterizan por su poder expresivo. Presentamos (fig. No.2) algunas gráficas sencillas correspondientes a conceptos representados con redes semánticas.

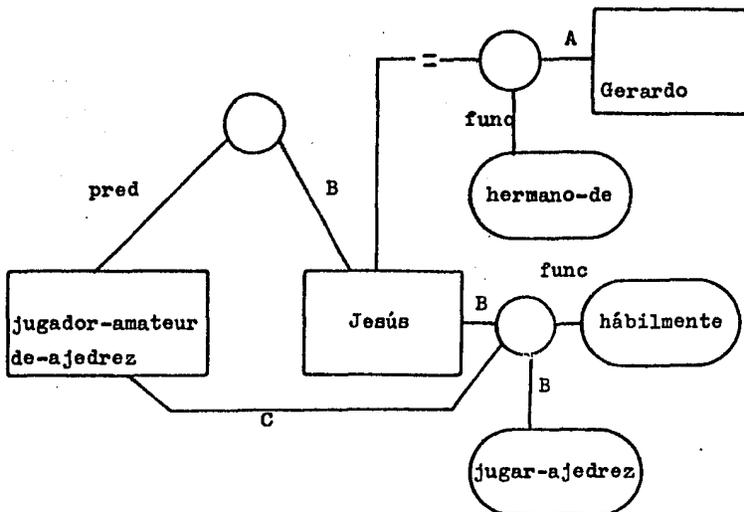
Se tienen cuatro referencias importantes que presentan fundamentos teóricos, ideas y ejemplos de redes semánticas. Estas son (Levesque y Mylopoulos 1979), (Brachman 1979), (Hendrix 1979) y (Shubert, Goebel y Cercone 1979), las cuales están recopiladas en una sola obra.



" Amparo es mujer "



" Jaime regala chocolates a Amparo "



" Jesús, hermano de Gerardo, juega hábilmente ajedrez como amateur ".

FIGURA No. 2

Algunos ejemplos de redes semánticas.

## 2.7.2.- LAS REGLAS.

Las reglas de la Base de Conocimientos activan rutinas, también llamadas procedimientos, que ejecutan una acción sobre los datos del sistema, de acuerdo al cumplimiento de alguna condición de la regla, correspondiente al estado actual del problema que se está resolviendo.

El conocimiento o experiencia del sistema, respecto a la disciplina para la cual es diseñado, está contenido en las reglas bajo alguna técnica de representación de conocimiento por procedimientos, a diferencia de las técnicas de representación declarativa utilizadas para los datos.

Las técnicas de representación de conocimiento por procedimientos más utilizadas son:

- a.- Predicados Lógicos de Primer Orden.
- b.- Reglas de Producción.
- c.- Redes Semánticas de Procedimientos.

Los predicados lógicos de primer orden son muy utilizados, ya que se puede hechar mano a las leyes de la lógica formal para la acción de inferir, aplicándolos a proposiciones. El cálculo proposicional proporciona un fundamento formal para las inferencias y las pruebas.

Con el lenguaje PROLOG pueden elaborarse Bases de Conocimiento bajo estas características de representación (Dhal 1983). Ejemplo de ello sería la regla:

```
ESTADO ( Y,W,Y,C ) IF ( ESTADO ( X,W,X,C ) AND
                        OPUESTO ( X,Y ) AND
                        ACEPTABLE ( Y,W,Y,C ) )
```

donde se tienen tres predicados ( ESTADO, OPUESTO y ACEPTABLE ) trabajando con un conjunto de argumentos ( C,W,X,Y ) que representan objetos en un orden.

Por otro lado, las reglas de producción son la técnica más utilizada en los Sistemas Expertos. Una regla de producción consta de dos elementos:

- a.- Antecedente.
- b.- Consecuente.

El antecedente es una expresión condicional referente al estado actual de los datos en el problema. El consecuente es la especificación de la acción que se ejecutará sobre dichos datos para modificar ese estado actual. La estructura de una regla de producción es:

SI < antecedente > ENTONCES / consecuente /

Las reglas de producción son un tipo particular de los llamados programas-modelo invocados (pattern-invoked programs). Un sistema configurado por reglas de producción, se le conoce como Sistema de Producción.

Se tienen varias maneras de elaborar reglas. Presentamos a continuación la descripción del Lenguaje para la construcción de reglas en EMYCIN (Barstow y varios 1983). EMYCIN es un sistema que se usa como herramienta para elaborar Sistemas expertos (ej: MYCIN).

```

regla ::= IF antecedente THEN accion (ELSE accion)
antecedente ::= AND condicion
condicion ::= OR condicion | predicado terna-asociativa
terna-asociativa ::= <atributo, conjunto, valor>
accion ::= consecuente | <procedimiento>
consecuente ::= terna-asociativa <factor de certeza>

```

Vamos que una condición puede constar de :

- La disyunción de una o más condiciones ó
- Un predicado, aplicado a una terna de atributo, objeto, valor. Los predicados pueden estar negados.

El factor de certeza se asocia a cada terna. Se trata de una probabilidad normalizada dentro del rango de -1 y +1 :

- 1 aseveración totalmente falsa.
- 0 no hay opinión.
- +1 aseveración verdadera incuestionablemente.

Pueden tenerse funciones de Conjuntos Borrosos (Fuzzy Sets) para indicar una grado de certidumbre.

Las redes semánticas de procedimientos (Mylopuolos, Shibahara y Tsotsos 1983) son una buena alternativa para la configuración de la Base de Conocimientos. Una característica, de las redes semánticas, es la posible interpretación de una estructura reticular en términos de la semántica de la lógica clásica. Por otro lado, puede hacerse referencia a (Brachman, Fikes y Levesque 1983) en donde se presenta una herramienta para la representación de conocimiento a través de procedimientos, en base a redes semánticas.

Los elementos generales de este tipo de redes se denominan como Clases y Relaciones. Una clase será un conjunto de objetos con propiedades comunes, correspondiendo a un nodo de la red. Una relación será una función de mapeo entre clases, correspondiendo a una liga de la red. Un procedimiento será una clase cuyos objetos son acciones sobre los datos del sistema (Levesque y Mylopuolos 1979).

Las reglas pueden tenerse en redes configuradas con marcos como nodos. La representación de una regla, utilizando marcos, es utilizada (Barstow y varios 1983) para ejemplificar a RLL:

#### REGLA # 332

EB: regla  
DESCRIPCION Decir al usuario que retenga la respiracion  
si el quimico es toxico.  
SI-TRABAJA-EN- Indagar-peligro-inminente.  
TAREA  
SI-POTENCIALMENTE-  
PERTINENTE (toxicidad alta?)  
SI-VERDADERAMENTE-  
PERTINENTE (quimico cerca del usuario?)  
ENTONCES-INDICAR-  
AL-USUARIO "No respire !"  
ENTONCES-ADICIONAR-  
A LA AGENDA (requerimiento ambulancia)  
PRIORIDAD alta  
IMPORTANCIA 900  
TIEMPO-CORRIDA 0.1 segs  
FRECUENCIA DE  
USO Considerada 985 veces, usada 4 veces  
GENERALIDADES (REGLA # 899, REGLA # 45)  
ESPECIALIZACION (REGLA # 336)

JUSTIFICACION Respiracion y muerte.  
 AUTOR Johnson.  
 FECHA CREACION 17:30,9-julio-81

El código de la regla se encuentra en diferentes campos del marco. Esto permite que haya más de un tipo de interpretación para la regla.

Una descripción más detallada, de cada una de las técnicas de representación de conocimiento, diferencias, usos, ventajas y desventajas, se encuentra tanto en (Kobsa 1984) como en (Ayala 1984).

### 2.7.3.- OTRA ALTERNATIVA EN LA REPRESENTACION DE CONOCIMIENTO.

Una alternativa interesante, en la representación de conocimiento, es el uso de matrices booleanas, manejadas como mapas de Karnaugh.

En este tipo de matrices pueden representarse varias reglas, donde cada uno de los minterminos se referirá a una proposición del antecedente. La ventaja de este tipo de representación es el espacio reducido de memoria utilizado. Sin embargo, la desventaja radica en una previa determinación del número de valores que cada atributo puede tomar.

A	B	COLOR	C	MATERIAL
-----			-----	
0	0	rojo	0	madera
0	1	azul	1	plástico
1	0	verde		
1	1	amarillo		
D	TAMANO		E	DUEÑO
-----			-----	
0	esférico		0	María
1	cúbico		1	Juan

Supongamos las siguientes reglas :

SI es azul, de plástico y esférico

ENTONCES es de Juan

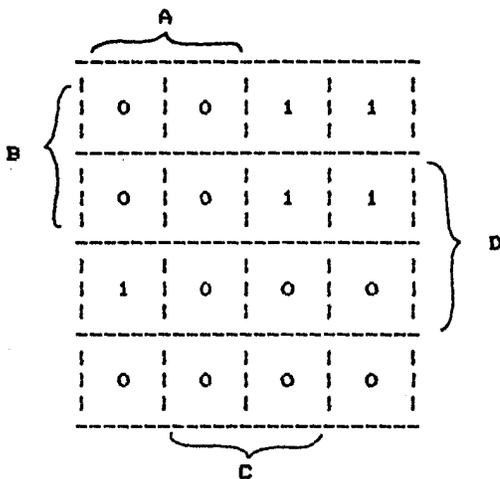
SI es azul, de plástico y cúbico  
ENTONCES es de Juan

SI es verde, de madera y cúbico o es azul y de madera  
ENTONCES es de Juan

Estas reglas se representan así :

$$\begin{aligned} E &= \bar{A}BC\bar{D} \\ E &= \bar{A}BCD \\ E &= A\bar{B}\bar{C}D + \bar{A}B\bar{C} \end{aligned}$$

El diagrama es el siguiente :



Si hacemos la reducción tendremos que :

$$E = \bar{A}B + A\bar{B}\bar{C}D$$

SI es azul o  
es verde, de madera y cúbico  
ENTONCES es de Juan

Ahora, proponemos un caso en el cual se tienen 5 variables para determinar el valor de una sexta :

A	B	COLOR	C	D	# de lados
0	0	rojo	0	0	3
0	1	azul	0	1	4
1	0	verde	1	0	5
1	1	amarillo	1	1	más de 5

E	LADOS IGUALES	F	DUEÑO
0	no	0	María
1	sí	1	Juan

Supongamos las siguientes reglas :

Regla # 1 :

$$F = ABCDE$$

Si es amarillo y tiene 4 lados iguales  
ENTONCES es de Juan

Regla # 2 :

$$F = ABCDE + ABCDE$$

SI es amarillo y tiene 3 lados iguales ó  
es amarillo y tiene 3 lados no iguales  
ENTONCES es de Juan

Regla # 3 :

$$F = ACDE + ACDE$$

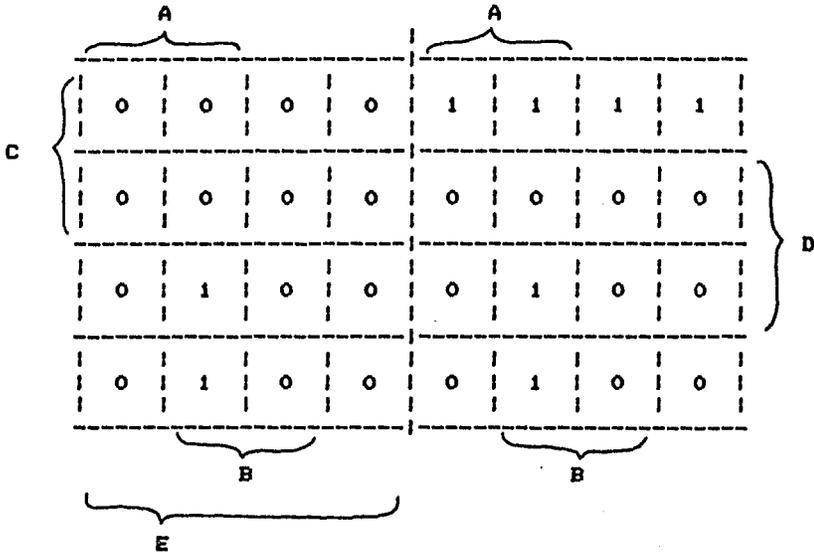
SI es verde ó amarillo y tiene 5 lados no iguales ó  
es rojo ó azul y tiene 5 lados no iguales  
ENTONCES es de Juan

Regla # 4 :

$$F = ABC\bar{D}\bar{E}$$

SI es amarillo y tiene 4 lados no iguales  
 ENTONCES es de Juan

Tenemos el diagrama siguiente:



Del diagrama puede verse que las cuatro reglas pueden simplificarse en sólo una :

$$E = C\bar{D}\bar{E} + ABC$$

SI tiene 5 lados no iguales ó  
 es amarillo y tiene 3 ó 4 lados  
 ENTONCES es de Juan

Un aspecto interesante en este tipo de representación es la reducción del conocimiento en reglas cada vez más generales. Esta operación ayuda al mecanismo de inferencia y, de hecho, forma parte de él.

El espacio de memoria, utilizado para esta representación, es reducido. Debido a la naturaleza de la representación, vemos que el número de bits necesarios para representar todas las posibles proposiciones que nos lleven a un consecuente es de :

$$2^n$$

Donde n es el número de variables que son utilizadas en los antecedentes. Así hemos visto que, en nuestro primer ejemplo, hemos utilizado 4 variables, y por ello una matriz de 16 bits. Para el segundo ejemplo se utilizó una matriz de 32 bits, pues se trataba de 5 variables.

También resulta interesante el número de posibles proposiciones que una estructura de este tipo puede manejar. Cada una de las proposiciones es un mintermino de la función. Tomando en cuenta la proposición IDENTIDAD, se tiene que el número de proposiciones posibles, para n variables, es de :

$$3^n$$

Por ejemplo, para n=3 tenemos :

I (identidad)

A	B	C	$\bar{A}$	$\bar{B}$	$\bar{C}$									
AB	AC	$A\bar{B}$	$A\bar{C}$	BC	$B\bar{A}$	$B\bar{C}$	$C\bar{A}$	$C\bar{B}$	$\bar{A}\bar{B}$	$\bar{A}\bar{C}$	$\bar{B}\bar{C}$			
ABC	$ABC$	$A\bar{B}C$	$A\bar{B}\bar{C}$	$\bar{A}BC$	$\bar{A}\bar{B}C$	$\bar{A}\bar{B}\bar{C}$	$\bar{A}BC$	$\bar{A}\bar{B}C$						

que son 27 minterminos, o sea,  $3^3$ .

Lo mismo sucede para otros valores de n. Si llamamos a np como el número de proposiciones representables (minterminos) tenemos que :

Para n = 2, np = 9  
 n = 3, np = 27  
 n = 4, np = 81

$n = 5, \quad np = 243$

etc...

El número de reglas, que se pueden representar con este tipo de estructuras, será menor o igual al número de proposiciones ( $np$ ). Pueden tenerse hasta  $np$  reglas con un solo mintermino como antecedente o una regla con  $np$  antecedentes.

## 2.8.- EL ESPACIO DE TRABAJO.

-----

El espacio de trabajo de un Sistema Experto es un área en memoria en donde se tienen los datos y la situación del problema, ya sea que se hayan proporcionado por el usuario o sean producto de inferencias del sistema a lo largo de la consulta. En el espacio de trabajo se encuentra la descripción del problema y el conjunto de posibles soluciones. A este conjunto se le llama ESPACIO DE ESTADOS, pues contiene los posibles estados del problema, así como los resultados intermedios.

### 2.8.1.- EL ESPACIO DE ESTADOS.

Los métodos de Inteligencia Artificial, para la solución de problemas, tienen su base en la representación de un espacio de estados o de soluciones, también denominado ESPACIO DE BÚSQUEDA. Este espacio permite la definición formal del problema para llegar a un estado final, partiendo de un estado inicial, a través de un conjunto de acciones u operadores. El espacio de estados permite que se defina un proceso de búsqueda para la solución del problema, siendo la búsqueda una técnica necesaria cuando no se tienen técnicas directas. Para tener una definición formal del problema es necesario contar con:

- a.- Uno o más estados iniciales.
- b.- Uno o más estados finales.
- c.- Un conjunto de reglas, en las cuales están definidas las acciones en sus consecuentes.
- d.- Tener el espacio de estados, quizá no explícitamente, sino de manera implícita.

Más tarde veremos que el espacio de búsqueda se representa generalmente en forma de árbol, en lo que se denomina gráfica del problema.

### 2.8.2.- EL USO DE UNA PIZARRA EN ALGUNOS SISTEMAS EXPERTOS.

Sistemas Expertos, como AGE, HERSAY II y HERSAY III, utilizan una pizarra como espacio de trabajo en la cual el sistema va dejando "avisos", reglas y soluciones intermedias para la solución del problema. De manera general una pizarra consta de tres secciones:

- 1.- Plan.  
Contexto de la situación, estado del problema, metas y planes actuales de solución.
- 2.- Agenda.  
Conjunto de reglas que esperan ser utilizadas, relevantes para alguna decisión tomada previamente.
- 3.- Solución.  
Hipótesis y decisiones candidatas y sus relaciones.

Una descripción mas detallada de cada una de ellas puede encontrarse en (Hayes-Roth, Waterman y Lenat 1983).

## 2.9.- LA MAQUINA DE INFERENCIA.

---

La máquina de inferencia, a la cual se hace referencia también como el "esqueleto" del Sistema Experto o "estructura de control", es un conjunto de rutinas y estructuras tanto para la producción como para el control de decisiones, así como para el manejo de preguntas al usuario. En esta parte el Sistema Experto no se debe contener, por lo menos teóricamente, información acerca del dominio específico del problema para el cual ha sido diseñado. Debe existir cierta independencia que permita un grado de generalidad.

A pesar de lo anterior, la base de conocimientos y la máquina de inferencia están íntimamente ligadas, dado que la primera presenta la organización del conocimiento en base al cual la segunda va a trabajar. Existe entonces una estrecha relación entre la organización del sistema experto y su control. Una estructura de control requiere de una organización en la base de conocimientos correspondiente al sistema, teniendo así mecanismos de inferencia apropiados para la búsqueda y aplicación del conocimiento (Hayes-Roth 1984).

Los métodos de razonamiento, implementados en la máquina de inferencia, no pueden ser completamente independientes del tipo de problemas a resolver, sin embargo, estas rutinas de razonamiento preferentemente no deben ser tan especializadas, pues no podrían entonces aplicarse a otros problemas (Weiss y Kulikowski 1984).

El razonamiento del Sistema Experto debe ser guiado de alguna manera, con el fin de proporcionar una secuencia de respuestas válidas y coherentes al usuario, presentando un medio ambiente aceptable. Para tal guía es necesaria un conjunto de estrategias de control. Para un Sistema Experto pueden tenerse tres estrategias (Weiss y Kulikowski 1984) las cuales son:

- 1.- Estrategias de razonamiento.
- 2.- Estrategias de explicación.
- 3.- Estrategias de interpretación.

El poderío de un Sistema Experto podría medirse de acuerdo a que tan correctamente son aplicadas las técnicas de razonamiento sobre el conocimiento almacenado. El éxito del sistema radicará en encontrar una buena respuesta a un problema en base a los recursos con que se dispongan. La eficiencia en la búsqueda de soluciones afecta directamente al éxito del sistema.

En (Hayes-Roth, Waterman y Lenat 1983) se presenta un conjunto de factores que ayudan a la eficiencia del Sistema Experto, así como algunos que incrementan la dificultad del problema. Las ayudas para la eficiencia son las siguientes:

- 1.- Un conocimiento almacenado que sea aplicable y correcto.
- 2.- La eliminación de alternativas de solución que lleven al sistema a un "callejón sin salida", o sea, que no le lleven a ninguna solución.
- 3.- La no redundancia en los procesos y en los datos.
- 4.- El incremento de rapidez en las operaciones.
- 5.- La existencia de fuentes de conocimiento.
- 6.- El razonamiento a varios niveles de abstracción.

Por otro lado, existen factores que incrementan la dificultad del problema que se desea resolver. Estos son:

- 1.- El almacenamiento de conocimiento y datos erróneos.
- 2.- Los cambios en los datos, que corresponden al carácter dinámico de la situación del problema.
- 3.- El número de posibilidades a evaluar.
- 4.- Los procedimientos complejos que implica el desechar una posibilidad.

El Sistema Experto, para la solución de algún problema, debe buscar una solución. Esta tarea le corresponde directamente a la máquina de inferencia. El acceso directo a una solución única no es posible en la mayoría de los casos. Generalmente son utilizadas técnicas de búsqueda, así como generadores de soluciones. Estos últimos permiten al sistema tener una vista de todas las posibles soluciones al problema, pudiendo probar cada una de ellas hasta encontrar la apropiada.

Basándonos en (Michaelsen, Michie y Boulanger 1985) podemos decir que en el esquema de control de un Sistema experto pueden encontrarse cuatro actividades importantes:

- a.- Selección de reglas y datos.
- b.- Evaluación y comparación de reglas con la situación del problema.
- c.- Planeación de la ejecución de las reglas seleccionadas que satisfacen la situación del problema. Si existe algún conflicto utiliza heurísticas.
- d.- Ejecución de la regla escogida, para modificar la situación del problema en ese momento.

Existen varios métodos para la búsqueda de soluciones. Uno de ellos, diríamos el más simple, es el de analizar todas las alternativas una por una. A este método se le denomina de "fuerza bruta". El problema grande con esta técnica es la existencia, en algunas ocasiones, de espacios de búsqueda de gran tamaño. Otro método de búsqueda es el denominado "de poda", con el cual se eliminan algunas alternativas, mejorando así la eficiencia respecto al caso anterior. Existen también métodos heurísticos, los cuales son más efectivos y de mayor ayuda (Hayes-Roth 1984).

Las heurísticas mejoran la eficiencia de las rutinas de búsqueda, algunas de ellas son usadas en la definición de la máquina de inferencia mientras otras residen en las propias reglas (Rich 1983).

#### 2.9.1.- BUSQUEDA DE SOLUCIONES.

Existen varios métodos para encontrar soluciones en un espacio de búsqueda, todos ellos variaciones de búsquedas heurísticas. Un proceso de búsqueda puede verse como el recorrer una gráfica, en la cual los estados son representados por los nodos. La búsqueda consiste en encontrar una ruta en dicha gráfica, comenzando de un estado inicial a uno o mas estados finales. Raras veces la gráfica completa se construye de manera explícita; generalmente se representa implícitamente en las reglas y se genera explícitamente tan solo la parte del espacio de búsqueda sobre el que se trabajará. Algunas características de las técnicas de búsqueda son:

- a.- Dirección en que se conducirá la búsqueda.
- b.- Selección de reglas.
- c.- Uso de una función heurística para guiar la búsqueda.

Con respecto a la dirección dos métodos son los más utilizados:

- a.- Búsqueda hacia adelante.
- b.- Búsqueda hacia atrás.

Ambos metodos son fáciles de aplicar y entender en sistemas de producción. Explicaremos cada uno de ellos suponiendo la generación de un árbol de soluciones (Rich 1983).

### 2.9.2.- BUSQUEDA HACIA ADELANTE.

- Se tiene la raíz del árbol (evidencia).
- Encontrar todas las reglas cuyos antecedentes igualen a la raíz.
- Utilizar los consecuentes de dichas reglas para crear nuevos nodos y ramas, generando el segundo nivel.
- Crear el siguiente nivel del árbol tomando cada nodo del nivel anterior y aplicarle todas las reglas cuyos antecedentes le igualen.
- Repetir el paso anterior hasta generar un nodo igual al estado final o meta.

### 2.9.3.- BUSQUEDA HACIA ATRAS.

- Se tiene la raíz del árbol (hipótesis).
- Encontrar todas las reglas cuyos consecuentes igualen a la raíz.
- Utilizar los antecedentes de dichas reglas para crear nuevos nodos y ramas, generando el segundo nivel.
- Crear el siguiente nivel del árbol tomando cada nodo del nivel anterior y encontrando todas las reglas cuyos consecuentes le igualen.
- Utilizar los antecedentes de dichas reglas para generar nuevos nodos.
- Repetir los dos pasos anteriores hasta encontrar el nodo correspondiente a un estado final (evidencia).

A la estrategia de búsqueda hacia adelante se le denomina también como:

- Búsqueda conducida por datos (data driven search).
- Búsqueda conducida por antecedentes (antecedent driven search).
- Búsqueda de tipo Top-Down.

La técnica de búsqueda hacia adelante implica partir de una evidencia, o sea, de uno o varios valores de atributos de objetos reconocidos como ciertos (Kinnucan 1984). Se debe tomar en cuenta cada elemento de la evidencia para llegar a una conclusión.

A la técnica de búsqueda hacia atrás se le denomina también como:

- Búsqueda conducida por objetivos ( Goal driven search ).
- Búsqueda conducida por consecuentes ( Consequent driven search ), pues se hace referencia al consecuente de

- las reglas como base de la búsqueda.
- Búsqueda de tipo Bottom-Up.
- Encadenamiento hacia atrás ( Backward chaining ).

La búsqueda se inicia tomando al estado final o meta y aplicando los operadores de inferencia inversos hasta llegar a un estado inicial, generando una ruta. El aplicar el inverso de un operador de inferencia a un estado cualquiera implica el encontrar los estados intermedios que hicieron que, a través de aplicar el operador original a un estado inicial, se llegara a dicho estado cualquiera.

A esos estados intermedios se les denomina sub-metas, las cuales son alcanzadas a partir de un estado final, el cual generalmente es un HIPOTESIS. La máquina de inferencia trabajará en base a una solución hipotética buscando la evidencia que la soporte, formulando y probando las mencionadas sub-metas que no son más que hipótesis intermedias (Kinnucan 1984).

El sistema propondrá una regla inicial, la cual será invocada "de reversa". El sistema comienza con examinar el consecuente de la regla, tomándolo como objetivo, meta o hipótesis. Presentamos, en pseudocódigo, una rutina de búsqueda "hacia atrás", basándonos en la utilizada por EMYCIN:

```

procedimiento ATRAS( atributo , objeto )
inicio
  s:recobrar las reglas cuyos consecuentes sostengan a
  VALOR( atributo,objeto ).
  s:MIENTRAS ( exista una regla sin probar ) Y
    ( la certeza de VALOR( atributo,objeto ) < 1 HAZ
  inicio
    s:PARA CADA condicion C en el antecedente de la
    regla HAZ
    inicio
      s:obtener atri y obj al evaluar la condicion C
      s:PARA CADA par (atri,obj) HAZ
      inicio
        SI VALOR( atri,obj ) no esta determinado
        ENTONCES invocar ATRAS( atri,obj )
      fin
    fin
  SI toda condicion C es verdadera
  ENTONCES ejecutar el consecuente de la regla
  fin
SI VALOR( atributo,objeto ) no es conocido
ENTONCES preguntar al usuario por el valor
fin.

```

Como podemos apreciar, la rutina trabaja para determinar si un valor propuesto para un atributo de algún objeto es válido. Para ello primeramente se debe tener un conjunto de reglas cuyos consecuentes soporten la veracidad de dicho valor. Las reglas serán aplicadas hasta establecer, con plena certeza, que dicho valor es válido o hasta terminar de analizar todas las reglas del conjunto. Si finalmente la búsqueda no fué exitosa, el sistema deberá preguntar el valor al usuario.

Debe notarse que primeramente se debe establecer si el antecedente de la regla es verdadero. Esto se hace examinando la veracidad de cada una de las condiciones que lo compongan. Para hacer esto, la rutina tendrá que establecer los valores de otros atributos de objetos, con lo cual aparecen sub-metas, las cuales son resueltas utilizando el mismo algoritmo. Aquí radica el carácter RECURSIVO de la rutina.

#### 2.9.4.- BÚSQUEDA BIDIRECCIONAL.

Existe otra estrategia comúnmente utilizada, consistente en combinar los dos mecanismos anteriormente presentados. Se le denomina búsqueda bidireccional, pues la dirección cambia de adelante hacia atrás, según convenga. La técnica consiste en:

- Elaborar una búsqueda hacia adelante proponiendo una o más hipótesis basadas en datos iniciales.
- Considerar dichas hipótesis como base a una búsqueda hacia atrás.
- Si se descubre una nueva evidencia, en el proceso de búsqueda hacia atrás, el sistema activará la búsqueda hacia adelante.

En (Rich 1983) se presentan tres factores para determinar cuál de los dos métodos (hacia atrás o hacia adelante) conviene utilizar:

- 1.- Contemplar si habrán más estados iniciales o finales. Es mas fácil buscar partiendo de un conjunto reducido de estados.
- 2.- El factor de ramificación ( número de nodos en promedio que pueden alcanzarse desde un solo nodo). Se debe determinar en que dirección será mayor el factor de ramificación.
- 3.- Decidir si es necesario que el sistema justifique su razonamiento. Es importante diseñar un sistema que proceda en la dirección correspondiente a la manera de pensar de el usuario.

### 2.9.5.- REDUCCION DEL PROBLEMA.

Los métodos de reducción de problema son estrategias de control de alta eficiencia utilizados en algunos sistemas expertos, como DENDRAL. El problema se divide en sub-problemas, los cuales se descomponen a su vez, hasta que el sistema cuenta con un conjunto de pequeños problemas "primitivos", los cuales pueden resolverse directamente (Nau 1983).

Utilizando una gráfica AND/OR es posible representar dicha descomposición, teniendo con ello una "gráfica de reducción del problema". Esta gráfica cuenta con dos ramas:

OR : representa cuáles son las alternativas de descomposición.  
AND : representa un camino particular.

Se presenta un ejemplo de gráfica AND/OR en la cual puede verse como se diferencia entre ambas ramas (fig. No.3).

Cuando se tiene una solución al problema, ésta puede representarse en una gráfica de solución, que viene a ser una parte de la gráfica de reducción del problema. Esta gráfica de solución se define de la siguiente manera (Nau 1983):

Sean:

$n$  : un nodo en la gráfica AND/OR.

$N$  : el conjunto de nodos terminales, que representan problemas primitivos directamente solucionables.

$G$  : gráfica de solución de  $n$  a  $N$ .

Tenemos que

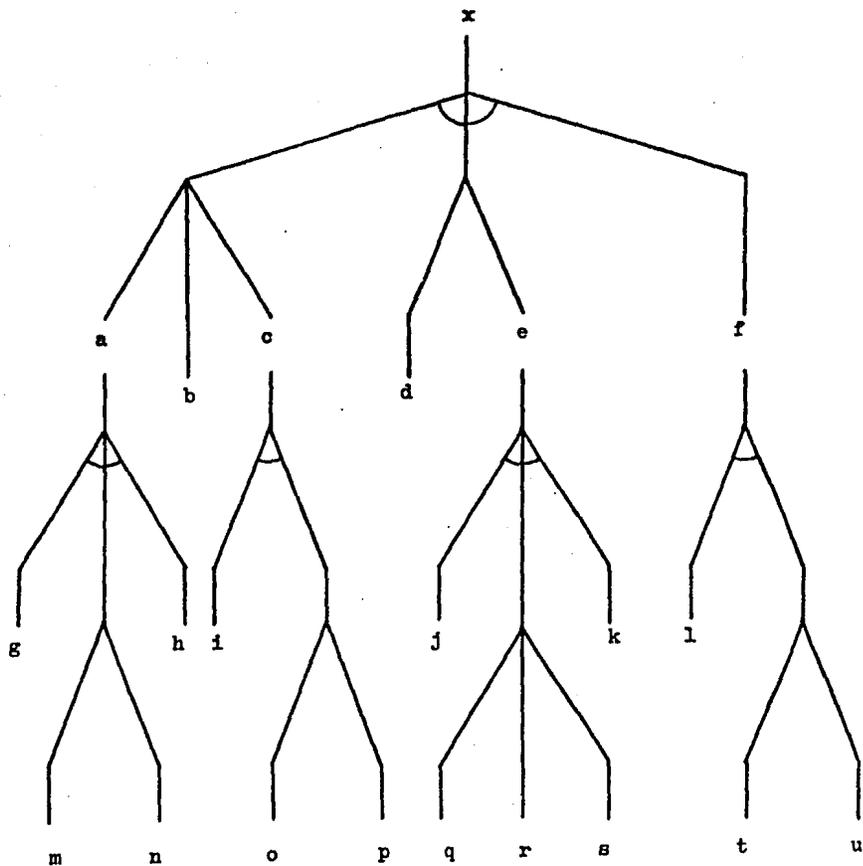
a.- SI  $n \in N$  ENTONCES  $n \in G$

b.- SI  $n \notin N$  y  $n$  es terminal ENTONCES  ~~$n \in G$~~

c.- SI  $n \notin N$  y  $n$  no es terminal y la rama de  $n$  a sus hijos es OR ENTONCES  $n$  tiene solución si alguno de sus hijos tiene solución, con lo que siendo  $(n_1, n_2, n_3, \dots, n_k)$  los hijos de  $n$  se dice que la gráfica de solución es  $G$ , de  $n$  a  $N$ , con lo que

SI  $\exists G_i \forall i$  ENTONCES  $G = G_i + n + (n, n_i)$ , donde  $(n, n_i)$  es la rama de  $n$  a  $n_i$ .

d.- SI  $n \notin N$  y  $n$  no es terminal y la rama de  $n$  a sus hijos es AND ENTONCES  $n$  tiene solución si todos sus hijos tienen solución, con lo que siendo  $(n_1, n_2, n_3, \dots, n_k)$  los hijos de  $n$  se dice que



$x = abc \text{ OR } de \text{ OR } f$   
 $a = g \text{ OR } mn \text{ OR } h$   
 $c = i \text{ OR } op$

$e = j \text{ OR } qrs \text{ OR } k$   
 $f = l \text{ OR } tu$

FIGURA No. 3

Gráfica AND/OR

$$\text{SI } \exists G_i \forall i \text{ ENTONCES } G = \sum_{j=1}^k G_j + n + \sum_{j=1}^k (n, n_j)$$

Es importante tomar en cuenta que la búsqueda en un espacio reducido de posibles resultados puede ser una tarea más sofisticada y compleja que una búsqueda simple en un espacio sin reducir (Nau 1983).

#### 2.9.6.- SELECCION DE REGLAS APLICABLES.

Hemos visto que para cada estrategia de búsqueda es necesario el seleccionar un conjunto de reglas para trabajar con ellas. Para ello es necesario una equidad entre la situación del problema y las condiciones de las reglas. Existen varios métodos para la selección de reglas. Presentaremos brevemente tres de ellos:

##### 1.- Manejo de índices.

Se tendrá un indicador dentro de las reglas, el cual corresponde a un estado o situación posible. Al presentarse dicha situación, las reglas correspondientes son seleccionadas.

##### 2.- Igualación con variables.

Usado cuando no hay un valor fijo para una situación particular, pero si hay propiedades para la situación. Un ejemplo sería un conjunto de relaciones en PROLOG expresadas en variables (ver capítulo anterior).

##### 3.- Igualación compleja y aproximada.

Si el antecedente de la regla requiere de propiedades no explícitas en la situación actual del problema, algunas propiedades deben ser inferidas en base a otras. Se tendrá que las condiciones de la regla igualarán aproximadamente a la situación. Puede utilizarse un factor de tolerancia.

#### 2.9.7.- EVALUACION DE REGLAS.

A grandes rasgos se tienen tres tipos de evaluación:

##### a.- Evaluación booleana.

Aplicable a las reglas que sólo contienen evidencias en sus antecedentes, los cuales pueden manejarse de acuerdo a métodos booleanos.

b.- Evaluación probabilística.

Aplicable a reglas cuyos antecedentes presentan una probabilidad.

c.- Evaluación con factores de confianza.

Los métodos heurísticos implican interpretaciones cuasi-probabilísticas. Hay formalismos lógicos como los de la LOGICA BORROSA (fuzzy logic) para describir el razonamiento del experto. Se utiliza una función de membresía de valores múltiples para denotar la membresía de un objeto a una clase, a diferencia de los valores booleanos. Se utiliza un rango de números reales, desde -1 a +1, utilizados como heurísticas sin asociarles un criterio de optimización.

El uso de factores de confianza, también llamados factores de certeza, se encuentra en varios Sistemas Expertos; uno de ellos es MYCIN. A cada una de las reglas de producción se le asocia un factor de confianza (Nau 1983). Se tendrían reglas como éstas:

SI a AND b AND c THEN d ( fc=90% )

SI h AND i AND j THEN d ( fc=70% )

Podemos obtener el factor de certeza del concepto d, si suponemos los factores de certeza :

a ( fc=95% )

b ( fc=20% )      fc=20%       $0.9 * 0.2 = 0.18$

c ( fc=50% )      (mínimo)

d ( fc=56% )

h ( fc=80% )

(máximo)

i ( fc=95% )      fc=80%       $0.7 * 0.8 = 0.56$

j ( fc=90% )      (mínimo)

Para explicación mas amplia de estos tipos de evaluación se recomienda consultar ( Weiss y Kulikowski 1984). Por otra parte, en (Barstow y varios 1983) se presentan las características principales de algunas máquinas de inferencia para sistemas expertos, como lo son EMYCIN, KAB, EXPERT etc.

## 2.10.- LA INTERFAZ CON EL USUARIO.

---

La comunicación de cualquier sistema, con su medio ambiente, es de suma importancia. La comunicación forma parte de la solución del problema. La interfaz con el usuario es la parte del Sistema Experto a través de la cual debe existir un diálogo entre el experto electrónico y la persona que solicita sus servicios. Este módulo del sistema, también denominado como procesador de lenguaje y, en algunos casos, interfaz a lenguaje natural, tiene como finalidad:

- Facilitar la entrada de datos, por parte del usuario, para la descripción del problema.
- Preguntar al usuario por información.
- Proporcionar al usuario los resultados al problema que se determinó, así como la explicación de su razonamiento.

Podemos decir que sus funciones primordiales son:

- a.- Interpretar preguntas, comandos e información.
- b.- Formatear la información presentada por el sistema.
- c.- Manejar preguntas y respuestas.
- d.- Presentar una justificación de su comportamiento.

Muchos Sistemas Expertos han sido diseñados para trabajar en base a una consulta interactiva, preguntando al sistema por información acerca del problema, tanto datos como el factor de confianza de los mismos. En algunos sistemas el usuario puede preguntar el porqué se le ha formulado una pregunta o porqué se le da una respuesta específica, interrumpiendo el proceso. Se dice que dichos sistemas trabajan en un "mixed-initiative mode".

Las reglas de producción facilitan la tarea de la interfaz al usuario, ya que puede tomarse la condición de la regla y presentarse como pregunta. Así mismo pueden utilizarse los componentes de las reglas para la generación de explicaciones de razonamiento.

### 2.10.1.- INTERFAZ A LENGUAJE NATURAL.

Cuando se utiliza una interfaz que trabaja con un subconjunto de un lenguaje natural completo, como lo es el español, se dice que se está usando una INTERFAZ A LENGUAJE NATURAL. Las interfaces a lenguaje natural son muy atractivas, dado su poderío. Sin embargo éste implica dificultades de tipo

práctico.

Utilizar interfaces a lenguaje natural no es siempre lo más adecuado. En (Rich 1984) se presentan 7 factores a considerar para el uso de una interfaz a lenguaje natural dentro del sistema:

a.- Facilidad de aprendizaje.

El esfuerzo, por parte del usuario para comunicarse con el sistema, es reducido. Mientras más restringido sea el lenguaje de la interfaz, mayor será el esfuerzo del usuario, quien deberá aprender las restricciones y limitaciones en la comunicación.

b.- Precisión.

Existe ambigüedad en muchas oraciones, las cuales pueden tener más de un significado. Una interfaz a lenguaje natural no sería apropiada si la precisión es importante.

c.- Brevedad.

El número de caracteres de los elementos del lenguaje es directamente proporcional al tiempo requerido para capacitar al usuario en el manejo del sistema. Se ha visto que los usuarios, con el tiempo, prefieren utilizar comandos cortos y breves. Debemos tomar en cuenta si nuestros usuarios son gente novata, con experiencia o casuales.

d.- Complejidad semántica.

Cuando el conjunto de posibles oraciones es grande los lenguajes naturales son concisos y eficientes. Una interfaz compleja a lenguaje natural es utilizada si el sistema debe enviar un gran conjunto de mensajes.

e.- Imágenes.

Algunos conceptos requieren imágenes para establecer la comunicación con el usuario, preferentemente al uso de palabras. Para ello existen dispositivos periféricos.

f.- Coherencia con el resto del sistema.

El usuario esperará un comportamiento sofisticado por parte del sistema si está interactuando con una interfaz sofisticada.

#### g.- Costo.

Su diseño e implementación son más costosos que para una interfaz de lenguaje restringido.

De manera general, tres son los componentes de un sistema de comprensión de lenguaje natural (Rich 1984):

- 1.- Elementos de la oración (léxico).
- 2.- Estructura de la oración (gramática).
- 3.- Significado de la oración (semántica).

Para comprender una oración hay que hacer un ANALISIS LEXICOGRAFICO, dividiendo la oración en sus componentes. Para diseñar una interfaz a lenguaje natural es necesario seleccionar un léxico o diccionario, lo cual implica:

- a.- Definir los conceptos, de acuerdo a las acciones del sistema.
- b.- Seleccionar palabras para la representación de los conceptos. Debe distinguirse entre verbos, adjetivos, sujetos y conjunciones y proposiciones.

A la oración se le asocia una estructura, la cual corresponderá, probablemente, a la estructura de sus significado. Al proceso de asignar dicha estructura a la oración se le denomina ANALISIS SINTACTICO o "PARSING". Dicha estructura se representa, generalmente, en forma de árbol. Un ejemplo de ello está en la (fig. No. 4).

En algunas ocasiones la gramática se define por un conjunto de reglas, por lo que el análisis sintáctico se referirá a dicho conjunto. El proceso de "parsing" es sólo uno de los pasos en el proceso total de comprensión de oraciones. Su importancia radica en que determina la facilidad del siguiente paso, el cual es el de asignar un significado a la oración.

El asociar un significado a una oración se le denomina PROCESO SEMANTICO. Debe determinarse una acción por parte del sistema, de acuerdo a la oración que ha recibido. Se le denomina PROCESO PRAGMATICO al cual determina qué hacer al respecto del significado de la oración.

En (Rich 1984) se presentan tres propuestas de interfaces :

#### 2.10.2.- INTERFAZ BASADA EN MENUS.

Si le indicamos al usuario las opciones disponibles, eso nos

garantizaría que las oraciones que el usuario forme tengan un significado para el sistema. Esto puede hacerse a través de un conjunto de menús. Un sistema con estas características es NLX de Texas Instruments. En (Rich 1984) se presenta un ejemplo de menú, el cual tenemos a continuación:

---

COMANDOS:	Encontrar	Encontrar-el	Encontrar-todo
CARACTS:		CONECTORES:	CALIFICADORES:
# de parte		y	provisto por
nombre de la parte		o	cuyos colores son
cantidad		de	cuyo precio es
nombre del proveedor		el promedio	con envío #
direcc del proveedor		el menor	cuyo nombre es
# del proveedor			cuya dirección es
precio			quien provee
color			
NOMBRES:		COMPARACIONES:	ATRIBUTOS:
partes		entre	<numero de parte>
proveedores		igual a	<cantidad>
envíos		no igual a	<proveedor>
		<, >, <=, >=	<precio>
			<color>

---

Después de que el usuario escogió una palabra del menú, la pantalla debe actualizarse. Pueden existir menús que indiquen alternativas de palabras posteriores, presentando dichas opciones para la formación de oraciones válidas por parte del usuario.

El proceso de análisis sintáctico implicará:

- 1.- Construir la estructura de la oración.
- 2.- Dar la información correspondiente para actualizar los menús, con el fin de que éstos contengan las palabras siguientes a completar la oración.

El análisis semántico puede efectuarse cuando se esté generando la estructura de la oración.

Esta propuesta de interfaz a lenguaje natural tiene sentido cuando el número de alternativas es pequeño. Generalmente trabaja eficientemente, pues las opciones disponibles al usuario son restringidas.

### 2.10.3.- GRAMATICAS SEMANTICAS.

La idea, al utilizar gramáticas semánticas, es la comprensión de la oración en dos pasos solamente:

- Análisis lexicográfico.
- Análisis de sintáxis y semántica.

Una gramática semántica consiste, como toda gramática, en un conjunto de reglas. Presentamos una, basándonos en la utilizada en el sistema Ladder, utilizada como ejemplo en (Rich 1984):

```
edo. inicial ::= CONSULTA PROPIEDAD de BARCO
CONSULTA      ::= Cual es | Dime
PROPIEDAD    ::= la PROP-BARCO | PROP-BARCO
PROP-BARCO   ::= velocidad | longitud | tiempo
BARCO        ::= el NOMBRE | el mas rapido barco BARCO2
NOMBRE       ::= Quetzalcoatl | Kennedy | H.M.S. Castro | ...
BARCO2       ::= NACION que sea BARCO3 | BARCO3
BARCO3       ::= TIPO LOCALIZACION | TIPO
TIPO         ::= portaviones | petrolero | crucero | ...
NACION       ::= norteamericano | cubano | mexicano | ...
LOCALIZACION ::= en el golfo de Mexico | en el caribe | ...
```

### 2.10.4.- GRAMATICAS SINTACTICAS.

Si en la interfaz se utiliza un fragmento mayor de un lenguaje natural, es importante codificar, lo mejor posible, la regularidad del lenguaje que se esté utilizando. Esto implica la existencia de una gramática sintáctica, la cual genera estructuras como el árbol presentado en la ( fig. No. 4 ).

En este tipo de interfaz se utiliza la programación de dos tipos de reglas:

- Las reglas gramaticales usadas por el analizador sintáctico para la asignación de un árbol a cada oración.
- Las reglas para transformar el árbol en acciones del sistema.

Para concluir con este tema, cabe señalar que la construcción

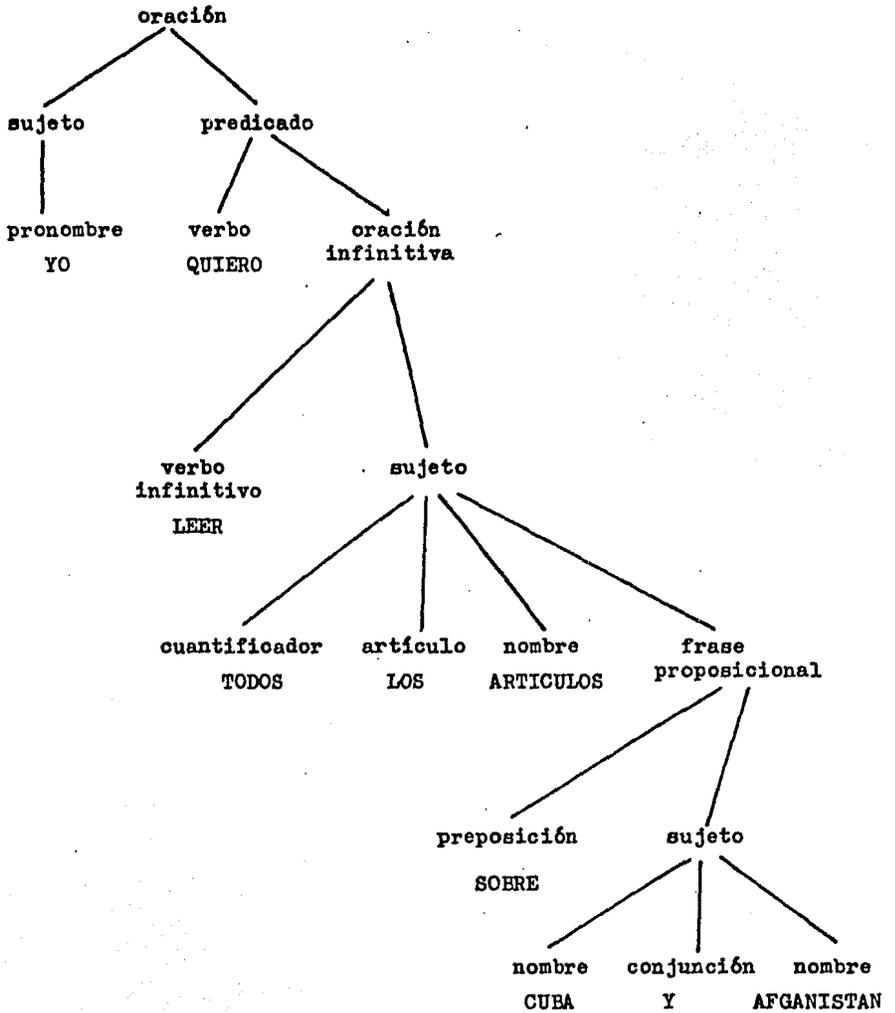


FIGURA No. 4

Arbol de análisis sintáctico.  
tomado de ( Rich 1984).

de una interfaz a lenguaje natural es una tarea que puede llevar tiempo. A pesar de que actualmente se cuenta con herramientas de software disponibles, para ayudar en su construcción, el problema no deja de ser algo complejo.

#### 2.10.5.- COMUNICACION ENTRE EXPERTOS.

Los Sistemas Expertos no únicamente pueden comunicarse con el usuario a través de una interfaz, sino que pueden comunicarse con otros sistemas, ya sea con Bases de Datos o con Bases de Conocimiento.

En (Guzmán 1984) se propone la interacción entre Sistemas Expertos, presentando la opción de una Red de Expertos la cual trabajaría, de manera multidisciplinaria, en la solución de un problema. Esto sería posible si se diseñara un protocolo estándar para la comunicación entre Sistemas Expertos cualesquiera.

La bibliografía presentada en la tesis presenta, bajo el inciso "e", algunos escritos que se consideran importantes en relación a la interfase con el usuario y lenguaje natural.

En el siguiente punto a tratar hablaremos también acerca de la interacción de expertos, solo que se tratará de la comunicación entre el experto humano y el electrónico.

## 2.11.- SUBSISTEMA DE ADQUISICION DE CONOCIMIENTO.

---

El Sistema Experto debe permitir, a sus constructores y algunos usuarios, la creación y modificación de su base de conocimientos. El sistema debe tomar el conocimiento del experto humano de alguna manera. Este proceso de adquisición, si se hace manualmente, puede resultar una tarea laboriosa, por lo cual es necesaria la existencia de un módulo de adquisición de conocimiento como parte del Sistema Experto. La tarea de adquisición de conocimiento puede definirse como "la transferencia y transformación de la experiencia para la solución de problemas, a partir de alguna fuente de conocimiento, hacia un programa" (Buchanan y varios 1983). Dicha experiencia se encuentra en un conjunto de hechos, procedimientos y reglas.

### 2.11.1.- LA LABOR DEL INGENIERO EN CONOCIMIENTO.

La persona que sirve como intermediario entre el Sistema Experto y el experto humano, para la construcción de la base de conocimientos del primero, es el INGENIERO EN CONOCIMIENTO. El ingeniero en conocimiento debe extraer el conocimiento del experto humano y seleccionar una representación óptima de conocimiento para el sistema. Puede también expandir y modificar la base de conocimiento del sistema, de acuerdo a las necesidades que se presenten. Debemos recordar que el desempeño del sistema va a depender de las características cualitativas y cuantitativas del conocimiento que se tenga almacenado, por lo que debe procurarse tener un conocimiento específico, depurado y detallado.

Los conceptos manejados por el experto deben identificarse y formalizarse; con la ayuda del ingeniero en conocimiento es posible estructurar el conocimiento del experto. El experto humano utiliza un vocabulario que, en la mayoría de los casos, es inadecuado para la implementación del conocimiento y la solución del problema. Por ello, el ingeniero en conocimiento debe ayudar al experto en extender y refinar dicho vocabulario, para que pueda existir una comunicación óptima con el sistema.

### 2.11.2.- LA ETAPA DE ADQUISICION DE CONOCIMIENTO.

El proceso de extraer el conocimiento de una fuente de experiencia e implementarlo en un sistema de cómputo es una tarea importante y no fácil. Podría decirse que se trata del "cuello de botella" en la construcción del Sistema Experto. Dicho proceso implica, a grandes rasgos:

- 1.- La definición de los conceptos y relaciones que constituyen el conocimiento del experto.
- 2.- La implementación de la base de conocimiento.
- 3.- Las pruebas y ajustes de la base de conocimiento.

Podría decirse que la etapa de adquisición de conocimiento es "el paso del Sistema Experto desde su concepción a su adolescencia" (Buchanan y varios 1983).

Habíamos mencionado una transferencia y transformación de la experiencia, o conocimiento, para implementarlo en el sistema. Existen métodos manuales, parcialmente automatizados y completamente automatizados para lograrlo. Generalmente son utilizadas algunas herramientas para la construcción y mantenimiento de la base de conocimientos, ayudando así tanto al ingeniero en conocimiento como a los expertos humanos.

### 2.11.3.- METODOS.

Podemos hablar de cinco maneras de adquisición de conocimiento :

- a.- Manual.
- b.- A través del Ingeniero en conocimiento.
- c.- Utilizando un editor.
- d.- Con un programa de inducción.
- e.- Con un programa para la comprensión de textos.

Los primeros programadores de sistemas inteligentes de conocimiento no separaban al mecanismo de razonamiento del conocimiento del sistema. Las adiciones y cambios del conocimiento implicaban modificaciones al código del programa. El programador debería conocer el campo para el cual el sistema era construído. Dadas las dificultades que esto implicaba apareció, en base a esta necesidad, la ingeniería en conocimiento.

El experto, en lugar de tratar con un programador, trabajará ahora con una persona especializada en el manejo del conocimiento y experiencia, para implementarlos en la computadora. Esto se hace por medio de pláticas, elaborándose estructuras de datos y relaciones, así como reglas, las cuales estarán como una entidad separada al resto del sistema. Aparece entonces lo que será la base de conocimientos.

Puede prescindirse de la labor del ingeniero en conocimiento, utilizando un editor para interactuar con el experto. El uso de un editor facilita el proceso de introducir el conocimiento y disminuye los errores. Algunos editores sofisticados permiten:

- Auxilio en errores de sintaxis (p.ej: EMYCIN, STAMMER, ROSIE).
- Chequeo de inconsistencias semánticas, entre el conocimiento que se esta proporcionando y el que se tiene ya almacenado (p.ej: UNITS, KAS, RLL).

Existen algunos editores denominados "inteligentes", como el utilizado en TEIRESIAS.

Después de un tiempo el Sistema Experto, por sí mismo, puede adquirir conocimiento sin requerir de que un experto humano se lo proporcione en un momento dado. Esto se logra a través de un programa de inducción, simulando el proceso de aprendizaje por parte del sistema, en base a los datos y conocimiento que tiene almacenados. A pesar de su existencia, generalmente no son utilizados para la labor de adquisición de conocimiento.

Existen programas, como META-DENDRAL, que aprenden nuevos conocimientos en base a hechos y reglas previamente almacenados, generando nuevas reglas. Otro ejemplo es AQ11, creado para el diagnóstico de enfermedades en plantas. Este sistema es capaz de formar nuevas reglas correspondientes al diagnóstico, las cuales han resultado mas efectivas que las obtenidas por expertos humanos. El sistema EXPERT-EASE tiene la capacidad de generar una regla de conocimiento, ejecutable, en base a un conjunto de decisiones. Todo esto, en teoría, se basa en las denominadas conclusiones de Quinlan (Michaelsen, Michie y Boulanger 1985):

- a.- El sistema puede generar soluciones para problemas complejos de decisión por lo menos en el mismo tiempo que el programador utilizaría para desarrollarías.
- b.- La eficiencia de las estructuras de solución generadas es mayor en comparación a las programadas.

Actualmente son pocos los sistemas que adquieren conocimiento por ellos mismos, pues estas técnicas están aún en etapa de experimentación; sin embargo hay muchas investigaciones al respecto.

Finalmente, con miras a un futuro no lejano, puede pensarse en un programa para la comprensión de textos. De esta manera los conocimientos, datos y experiencia que adquirirá el sistema serán proporcionados de manera escrita, utilizando libros de texto, artículos, etc, con el manejo de un dispositivo de entrada adecuado. La programación de dichos sistemas implica una alta sofisticación.

#### 2.11.4.- PROBLEMAS EN LA ADQUISICION DE CONOCIMIENTO.

En el proceso de adquisición de conocimiento se presentan tres problemas principales:

- 1.- Desigualdad en la representación.  
Existen diferencias entre la forma en que el conocimiento es establecido por el experto humano y la manera en que debe representarse en la base de conocimientos. Una solución es una interfaz a lenguaje natural entre el experto y el sistema, con representaciones de conocimiento fácilmente propuestas en lenguaje natural.
- 2.- Conocimiento preciso, completo y consistente.  
Para el experto humano no es fácil describir su experiencia de una manera precisa completa y consistente. No olvidemos que la experiencia humana es, por naturaleza, subconciente por lo que puede ser una aproximación, quizás incompleta o inconsistente.
- 3.- La tecnología.  
Los lenguajes de representación, actualmente, son algo limitados en lo que respecta a sus capacidades expresivas. Aún cuando el experto humano sea capaz de proporcionar su experiencia al ingeniero en conocimiento, no resulta fácil caracterizar dicho conocimiento en un sistema de representación.

Una descripción mas detallada, y a la vez general respecto al proceso de adquisición de conocimiento, se tiene en (Buchanan y varios 1983).

## 2.12.-SUBSISTEMA DE EXPLICACION.

-----

En algunos casos es necesario que el Sistema Experto sea utilizado en forma interactiva. El usuario puede preguntar información acerca del estado del problema, así como cuestionar al sistema el porqué le hace alguna pregunta o porqué llega a una conclusión específica. A este componente de los Sistemas Expertos también se le denomina módulo justificador. El subsistema de explicación no es una parte indispensable de los sistemas expertos. Para algunas aplicaciones puede no ser necesario contar con que el sistema justifique su comportamiento.

Existe una relación directa entre la implementación de las reglas de la base de conocimientos y la facilidad con que el sistema puede llegar a encadenar sus inferencias y hacer preguntas al usuario. La utilización de reglas de producción permite una fácil elaboración de preguntas, simplemente tomando el antecedente. De la misma manera facilitan la generación de explicaciones. En algunos casos, como el de DENDRAL, la utilización de técnicas de inferencia y algoritmos poderosos no permiten, de manera general, determinar qué pasos de inferencia fueron requeridos para la obtención de una solución. Existe entonces un sacrificio de la capacidad explicativa del sistema por el poder de sus técnicas de inferencia.

De manera general, son dos las preguntas importantes que un usuario puede hacer al sistema:

- 1.- "Cómo llegaste a esa conclusión ?".
- 2.- "Para qué me preguntas eso ?".

Podemos hablar de dos técnicas de explicación:

- 1.- Reconstrucción simple.
- 2.- Razonamiento acerca del razonamiento.

### 2.12.1.- RECONSTRUCCION SIMPLE.

La manera más sencilla de dar una explicación es proporcionar al usuario la secuencia de pasos de inferencia que se dieron para llegar a una conclusión.

Podríamos decir que el proceso de explicación consiste en reconstruir los caminos de inferencia que el sistema ha recorrido para llegar a una conclusión. Para tener una explicación satisfactoria es recomendable tener una relación, de cada uno de

los pasos de inferencia encadenados, con principios fundamentales de la materia para la cual el sistema se ha diseñado (Brachman y varios 1983). Basándonos en el sistema experto propuesto en (Thompson y Thompson 1985) presentamos un ejemplo de como dar una explicación del comportamiento del sistema. Supongamos las siguientes reglas, correspondientes a la base de conocimientos de un Sistema Experto en botánica, muy simple :

- r1.- SI la clase es gimnosperma Y las hojas son escamosas  
ENTONCES la familia es ciprés.
- r2.- SI el tipo es árbol Y las hojas son anchas y planas  
ENTONCES la clase es angiosperma.
- r3.- SI el tipo es árbol Y las hojas no son anchas y planas  
ENTONCES la clase es gimnosperma.
- r4.- SI el tallo es verde  
ENTONCES el tipo es hierba.
- r5.- SI el tallo es de madera Y la posición es rastrera  
ENTONCES el tipo es enredadera
- r6.- SI el tallo es de madera Y la posición es vertical Y  
tiene un solo tronco  
ENTONCES el tipo es árbol.
- r7.- SI el tallo es de madera Y la posición es vertical Y  
no tiene solo un tronco  
ENTONCES el tipo es arbusto.

El sistema trabaja con una técnica de inferencia hacia atrás (backtraking). Supongamos que el usuario consulta al sistema, indicando que desea saber de qué familia es una planta.

- p1.- SIS: QUE ATRIBUTO DESEAS CONOCER ?  
usu: familia
- p2.- SIS: EL TALLO ES VERDE O DE MADERA ?  
usu: madera
- p3.- SIS: LA POSICION ES VERTICAL O RASTRERA ?  
usu: vertical
- p4.- SIS: TIENE UN SOLO TRONCO (SI/NO) ?  
usu: SI
- p5.- SIS: SUS HOJAS SON ANCHAS Y PLANAS (SI/NO) ?  
usu: SI
- p6.- SIS: CUAL ES LA FORMA DE LAS HOJAS ?  
usu: escamosa

c.-SIS: LA FAMILIA ES CIPRES.

Una manera práctica de almacenar los pasos de inferencia, desde el primero hasta el último que se dió y en ese orden, es utilizando una pila. En (Thompson y Thompson 1985) se propone el uso de dos pilas (stacks):

- Pila de metas, en la cual se encontrarán los objetivos a cumplir por la máquina de inferencia, así como una referencia a la regla de la que fueron tomados.
- Pila de contexto, en la cual se almacenan las conclusiones que el sistema ha producido durante la sesión, así como las respuestas del usuario a las preguntas que se le han hecho.

La pila de contexto, al concluir esta sesión de ejemplo, quedaría así :

```
-----  
| * familia      cipres |  
| + forma-hoja  escamosa |  
| * clase       gimnosperma |  
| + anchas-planas no |  
| * tipo        arbol |  
| + un-tronco   si |  
| + posicion    vertical |  
| + tallo       de madera |  
-----  
atributos      valores
```

Conclusiones :

- \* inferida por el sistema.
- + proporcionada por el usuario.

PILA DE CONTEXTO

Ante la pregunta :

usu : Cómo llegaste a esa conclusión ?

nosotros proponemos, para este sistema, una rutina sencilla de explicación a la pregunta del usuario. Sería una rutina como la siguiente :

```

inicio
bandera := sistema
imprime "Parece ser que "
MIENTRAS tope <> 0 HAZ
    SI el valor del tope es SI o NO ENTONCES
        imprime valor del tope
        imprime frase del atributo del tope
    DE LO CONTRARIO
        imprime frase del atributo del tope
        imprime " es "
        imprime valor del tope
    tope := tope - 1
    SI tope <> 0 ENTONCES
        SI el valor del tope lo dio el usuario ENTONCES
            SI bandera = usuario ENTONCES
                imprime " y que "
            DE LO CONTRARIO
                imprime ",porque me has dicho que"
            bandera := usuario
            fin del si
        DE LO CONTRARIO
            imprime ",ademas he inferido que"
            bandera := sistema
        fin del si
    fin del si
fin del mientras
fin

```

Cada uno de los atributos deberá tener una frase asociada, la cual proporcione un significado del atributo. Si suponemos las frases siguientes:

Atributos	Frases
familia	la familia de la planta
forma-hoja	la forma de sus hojas
clase	la clase de la planta
anchas-planas	tiene hojas anchas y planas
tipo	el tipo de planta
un-tronco	tiene un solo tronco
posicion	la posicion de la planta
tallo	el tallo de la planta

Entonces el sistema presentaría esta explicación :

Parece ser que la familia de la planta es cipres, porque me has dicho que la forma de sus hojas es escamosa, ademas he inferido que la clase es gimnosperma, porque me has dicho que no tiene hojas anchas y planas, ademas he inferido que el tipo de planta es arbol, porque me has dicho que si tiene un solo tronco y que la posicion de la planta es vertical y que el tallo de la planta es de madera.

De manera general, son dos las preguntas importantes que un usuario puede hacer al sistema:

- 1.- "Cómo llegaste a esa conclusión ?".
- 2.- "Para qué me preguntas eso ?".

Para que el sistema conteste al usuario el porque está haciéndole una pregunta, el sistema debe proporcionar una justificación. Esta puede darse, de acuerdo a este ejemplo, haciendo referencia al tope de la pila de metas y al tope de la pila de contexto. Supongamos, en un momento final de sesión en el sistema (p6), que las pilas contienen la siguiente información:

-----		-----		-----	
	* clase gimnosperma				
	+ anchas-planas no				
	* tipo arbol				
	+ un-tronco si				
	+ posicion vertical		forma-hoja		
	+ tallo de madera		familia		
-----		-----		-----	
atributos valores		atributos		indicador	
PILA DE CONTEXTO		PILA DE METAS		de regla	

Ante la pregunta del sistema :

SIS: CUAL ES LA FORMA DE LAS HOJAS?

El usuario podría preguntar :

usu: Para que me preguntas eso ?

Para que el sistema, en este ejemplo, diera una justificación,

proponemos lo siguiente:

inicio

imprimir "Dado que sabemos que "

SI valor del tope de contexto es SI o NO ENTONCES

imprimir valor del tope de la pila de contexto

imprimir frase del atributo del tope de la pila de contexto

DE LO CONTRARIO

imprimir frase del atributo del tope de la pila de contexto

imprimir " es "

imprimir valor del tope de la pila de contexto

fin del si

imprimir ", si me proporcionas "

imprimir frase del atributo del tope de la pila de metas

imprimir "conoceremos si "

imprimir frase del atributo del tope-1 de la pila de metas

imprimir " es "

imprimir valor del atributo correspondiente en la regla  
indicada en el tope-1 de la pila de metas

fin

La justificación sería la siguiente :

Por el momento conozco que la clase de la planta es gimnosperma,  
si me proporcionas la forma de sus hojas  
conoceremos si la familia de la planta es cipres

## 2.12.2.- RAZONAMIENTO ACERCA DEL RAZONAMIENTO.

Este método requiere de que el sistema reconstruya la línea de los argumentos de su razonamiento pero, a diferencia de la técnica anterior, esta reconstrucción tiene la característica de permitir un ambiente que se asemeja más a una "percepción general" que el simple rastreo de reglas.

El sistema trabajará con su propio razonamiento, como si el fuera un espectador. Esta técnica proporciona, en algunos casos, la construcción de una línea de razonamiento PARECIDA, pero no necesariamente igual, a la que se llevó a cabo en la sesión por parte del sistema.

Para que un Sistema Experto razone sobre su propio razonamiento generalmente se utilizan "metareglas", o sea, reglas que indican como manejar las reglas de la base de conocimiento. A esto se le

denomina METACONOCIMIENTO. El uso de "metareglas" para la elaboración de explicaciones tiene su mayor aplicación en el subsistema de adquisición de conocimiento, a través del cual el sistema explicará al experto humano, o al ingeniero en conocimiento, algunas cuestiones referentes a la base de conocimientos y al dominio para el cual el sistema está siendo diseñado. El razonamiento acerca del razonamiento, así como el uso de metaconocimiento en el Sistema Experto, se presentan ampliamente en (Lenat y varios 1983).

Un aspecto importante, tanto en el uso de metaconocimiento como en el rastreo de inferencias, es el implementar en el sistema un conjunto de PRINCIPIOS FUNDAMENTALES asociados a cada regla, los cuales podrán ser presentados al usuario cuando éste formule sus preguntas (Brachman y varios 1983 ). Para algunas aplicaciones de sistemas expertos puede no ser necesario contar con un subsistema de explicación.

## **2.13.- CONSTRUCCION DE SISTEMAS EXPERTOS.**

-----

El Ingeniero en Conocimiento es el encargado de la construcción del Sistema Experto. De manera general puede decirse que son cuatro sus funciones primordiales (Hayes-Roth 1984) :

- a.- Explotación.  
Actividad : Adquisición del conocimiento del experto.  
Producto : Conjunto de conceptos y reglas.
- b.- Modelado.  
Actividad : Diseño del Sistema.  
Producto : Estructura de representación de conocimiento.
- c.- Ensamblado.  
Actividad : Programación del conocimiento.  
Producto : Base de conocimientos y máquina de inferencia.
- d.- Refinamiento.  
Actividad : Depuración del conocimiento.  
Producto : Conceptos, reglas, base de conocimientos y máquina de inferencia depuradas.

### **2.13.1.- DESARROLLO DE UN SISTEMA EXPERTO.**

De acuerdo a las funciones del ingeniero en conocimiento, se han determinado cinco estados en el desarrollo de un sistema basado en conocimiento, de acuerdo a (Weiss y Kulikowski 1984) y (Hayes-Roth 1984) :

- 1.- Identificación.  
Se identifican las características referentes al dominio del cual el sistema será un experto. Se trata de la definición del problema. Se determinan también los recursos, participantes y funciones.  
Se obtienen los requerimientos.
- 2.- Conceptualización.  
Se definen los conceptos necesarios para la representación del conocimiento de la materia. Para ello es necesario una descripción detallada del problema, clasificarlo en sub-problemas si es necesario y determinar qué tanto afecta la

conceptualización en la posible implementación.  
Se obtiene el conjunto de conceptos.

**3.- Formalización.**

Se determinan las estructuras de diseño para la representación y organización del conocimiento. Se analizan estructuras alternativas.  
Se obtiene la estructura del sistema.

**4.- Implementación.**

Se hace la formulación de las reglas que relacionan los conceptos y que conformarán al conocimiento del sistema. Primeramente se implementa un sistema prototipo.  
Se obtiene el conjunto de reglas.

**5.- Pruebas.**

Se efectúa la validación de las reglas.  
Se obtiene la representación del conocimiento implementada y depurada.

Presentamos, a manera de esquema, estos estados de desarrollo de los sistemas basados en conocimiento (fig. No. 5).

### 2.13.2.- HERRAMIENTAS PARA LA CONSTRUCCION DE SISTEMAS EXPERTOS.

Actualmente existe software elaborado para facilitar la tarea del ingeniero en conocimiento. Las herramientas de software, para el ingeniero en conocimiento, reflejan el punto de una metodología determinada y un conocimiento general para la construcción de sistemas basados en conocimiento. Cada una de ellas ofrece una representación de conocimiento particular, por lo que se debe seleccionar la herramienta de acuerdo a la representación que el ingeniero en conocimiento crea más adecuada. Dichos sistemas, hasta hace poco, no se encontraban de manera comercial.

Los sistemas para la construcción de sistemas expertos cuentan con un lenguaje de programación de conocimiento, así como con un conjunto de utilerías para el procesamiento de éste. Un ejemplo es ROSIE. ROBIE es una herramienta de investigación para la elaboración de sistemas expertos. Cuenta con un lenguaje simbólico de programación de propósito general pero no proporciona una arquitectura para la resolución de problemas.

Algunos lenguajes, denominados de "bajo nivel" como PROLOG y LISP, no proveen estructuras. Ninguno de los dos proporciona restricciones al ingeniero en conocimiento ni proveen asistencia específica para la adquisición y representación de conocimiento, así como tampoco para la evaluación del sistema.

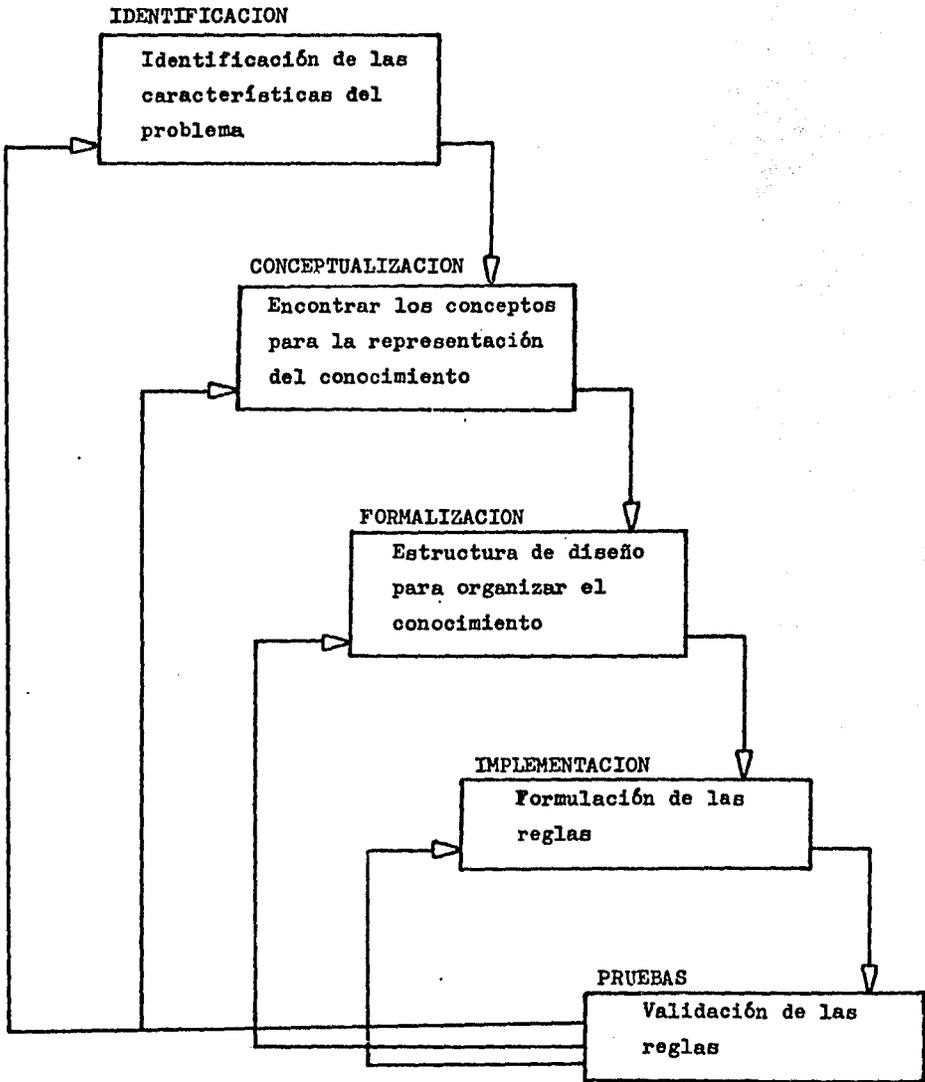


FIGURA No. 5

Estados de desarrollo de un sistema basado en conocimiento. Tomado de ( Hayes-Roth 1984 ).

## 2.14.- ALGUNOS SISTEMAS EXPERTOS.

-----

Presentamos a continuación una relación de los Sistemas Expertos más conocidos. Esta relación esta originalmente basada en una lista presentada en (Feingenbaum y McCorduck 1983), siendo una ampliación de la misma. Se indica el nombre y el área de aplicación para cada uno de los sistemas:

**ABEL**  
Diagnóstico de electrolitos ácido/base.

**AGE**  
Desarrollo de Sistemas Expertos.

**AIRPLAN**  
Planificación de tráfico aéreo.

**AL/X**  
Codificación de conocimiento de un dominio específico y generación de sistemas que lo manejen.

**AQ11**  
Diagnóstico de enfermedades en plantas.

**AUDITOR**  
Negocios.

**CADUCEUS**  
Diagnóstico diferencial en medicina interna.

**CALLISTO**  
Modelado, monitoreo y administración de proyectos.

**CASNET**  
Asociación de tratamientos con diagnósticos hipotéticos.

**CATS-1**  
Diagnóstico del mal funcionamiento de locomotoras diesel.

**CRIB**  
Diagnóstico de fallas en hardware y software.

**DART**  
Diagnósticos de fallas en sistemas de cómputo.

**DENDRAL**

Determinación de estructuras moleculares y elementos atómicos en base a interpretación de espectrómetros de masa.

**DIPMETER ADVISOR**

Análisis de información de pozos de petróleo.

**DRILLING ADVISOR**

Diagnóstico de problemas en pozos petroleros.

**EL**

Análisis de circuitos eléctricos.

**EMYCIN**

Sistema de inferencia básico, derivado de MYCIN. Utilizado para construir PUFF y SACON.

**EURISKO**

Diseño de circuitos de microelectrónica en tres dimensiones.

**EXPERT**

Exploración de petróleo y aplicaciones médicas.

**GENESIS**

Planeación y simulación de experimentos genéticos.

**GUIDON**

Instrucción respecto al conocimiento de MYCIN.

**HABP/SIAP**

Identificación y búsqueda de barcos detectados por sonar.

**HYDRO**

Resolución de problemas de recursos hidráulicos.

**INTERNIST**

Consulta médica.

**IBIS**

Control de almacén.

**KAS**

Sistema de adquisición de conocimiento para PROSPECTOR.

**KEPE**

Sistema de representación de conocimiento (comercial).

**KBVLSI**

Diseño de circuitos VLSI.

KB-300

Diagnóstico y asesoría industriales.

KM-1

Sistema de organización de conocimiento.

KMS

Consulta médica.

LDS

Modelado de procesos de toma de decisiones para abogados.

LOOPS

Sistema de representación de conocimiento usado por KBVLSI.

MDX

Consulta médica.

MACSYMA

Manejo de fórmulas matemáticas.

MEXEXP

Estructura de control para sistemas expertos. Mexicano.

MYCIN

Diagnóstico de meningitis e infecciones de la sangre.

MOLGEN

Análisis estructural y síntesis de DNA.

MRS

Sistema de representación de conocimiento y control de solución de problemas.

ONCOCIN

Tratamiento de cáncer.

OPS

Sistema básico de inferencia, aplicable en RI y AIRPLAN.

PROGRAMMER'S APPRENTICE

Construcción y depuración de software.

PROSPECTOR

Evaluación de depósitos minerales.

PSI

Construcción de programas en base a descripciones en inglés de las funciones a realizar.

**PUFF**

Análisis de datos del paciente para identificar anomalías pulmonares.

**RABBIT**

Ayuda de consultas a bases de datos.

**RAFFLES**

Diagnóstico en fallas de hardware y software.

**ROSIE**

Sistema básico de inferencia.

**R1**

Configuración de los sistemas VAX de Digital.

**SACON**

Identificación de estrategias de análisis para estructuras.

**SAGE**

Sistema básico de inferencia.

**SECS**

Planeación de síntesis orgánica, en química.

**SOPHIE**

Sistema de enseñanza en el diseño de circuitos electrónicos.

**SPEAR**

Análisis de la bitácora de errores en sistemas de cómputo.

**TAXADVISOR**

Negocios.

**TAXMAN**

Uso de reglas de impuestos y sugerencias a empresas al respecto.

**TEIREBIAS**

Adquisición de conocimiento.

**UNITS**

Sistema de representación de conocimiento, usado en MOLGEN junto con AGE.

**VM**

Monitoreo de pacientes de cuidados intensivos y consejo en terapia respiratoria.

**WAVES**

Consejero en análisis de datos sísmicos para la industria del petróleo.

**XCON**

**Configuración de los sistemas VAX de Digital.**

**XBEL**

**Selección de sistemas de cómputo apropiados (ventas).**

## CAPITULO III

### LOS SISTEMAS INTELIGENTES DE EDUCACION ASISTIDA POR COMPUTADORA.

=====

Los métodos de inteligencia artificial, además de aplicarse a varios campos, son ahora utilizados para la elaboración de sistemas educativos. A dichos sistemas se les conoce como SISTEMAS TUTORIALES INTELIGENTES o como SISTEMAS INTELIGENTES DE EDUCACION ASISTIDA POR COMPUTADORA. Les llamaremos "sistemas de ICAI" (Intelligent Computer-Assisted Instruction).

Los sistemas de ICAI aplican principios de inteligencia artificial, generando diálogos en lenguaje natural, haciendo uso de métodos de inferencia y utilizando estructuras y técnicas de representación de conocimiento, tanto de la materia que se va a enseñar como del conocimiento del estudiante.

#### 3.1.- CARACTERISTICAS DE LOS SISTEMAS DE ICAI.

-----

Entre las características que presentan los sistemas inteligentes educativos tenemos las siguientes (Wyer 1984):

- a.- Tratan de reproducir las características de un tutor humano.
- b.- Presentan varias estrategias tutoriales.
- c.- Generan un modelo correspondiente al aprendizaje del alumno.
- d.- Hacen que el alumno desarrolle sus propias técnicas de solución de problemas.
- e.- Promueven un aprendizaje activo, por descubrimiento.
- f.- Presentan una diferencia entre el material a enseñar y los métodos de enseñanza.
- g.- Permiten una flexibilidad en el ambiente del aprendizaje, tanto para el alumno como para el profesor.

#### 3.2.- ELEMENTOS DE UN SISTEMA DE ICAI.

-----

Primeramente hay que indicar que se tienen cuatro clases de conocimiento que, tanto el humano como la computadora, deben

utilizar para una enseñanza efectiva ( Woolf y McDonald 1984 ) :

- Conocimiento de la materia a enseñar.
- Conocimiento de la comprensión del estudiante.
- Conocimiento de estrategias de enseñanza.
- Conocimiento de la comunicación con el alumno.

De manera general podemos decir que los sistemas de ICAI, de acuerdo a las funciones que desempeñan, constan de tres componentes principales ( Roberts y Park 1983 ) :

- a.- Contenido a enseñar.
- b.- Estrategia de enseñanza.
- c.- Mecanismo de comprensión de habilidades del estudiante.
- d.- Comunicación con el alumno.

El objetivo de un sistema de ICAI será trabajar en base a estos tres componentes, con el fin de producir el ambiente de aprendizaje más efectivo posible. A cada uno de estos tres componentes le corresponde un módulo en el sistema ( Woolf y McDonald 1984 ) :

- a.- Módulo de experiencia.
- b.- Módulo tutorial.
- c.- Módulo de representación del estudiante.
- d.- Módulo de comunicación.

### 3.2.1.- EL MÓDULO DE EXPERIENCIA.

El objetivo del módulo de experiencia es el generar preguntas y evaluar las soluciones correspondientes que el estudiante propone a un problema en particular. Además, el sistema puede dar una explicación del razonamiento que ha utilizado para la resolución de algún problema. La existencia de un módulo experto, en estos sistemas, se propuso primeramente como un esfuerzo de introducir un componente inteligente en los sistemas de CAI ( Woolf y McDonald 1984 ). Al incluirse técnicas heurísticas los sistemas tutoriales tuvieron más flexibilidad y un mayor rango de acción pudiendo, en algunos casos, tanto generar como comprender diálogos en lenguaje natural.

Debe tenerse una representación de conocimiento para:

- a.- El contenido que va a enseñarse.
- b.- El cómo utilizar el conocimiento para la solución de problemas.

El contenido, o conocimiento de la materia que se desea impartir con un sistema ICAI, puede representarse utilizando varios métodos, entre ellos:

- 1.- Reglas de producción.
- 2.- Redes semánticas.
- 3.- Procedimientos.

### 3.2.2.- EL MODULO TUTORIAL.

Este módulo se refiere al conjunto de especificaciones de cómo se presentará al estudiante el contenido a enseñar. Aquí se indican las estrategias, procesos y reglas correspondientes a la interacción del sistema con el alumno.

Las funciones principales de este módulo son:

- Presentar al estudiante los problemas seleccionados, para que los resuelva.
- Monitorear las actividades del alumno ante el problema.
- Criticar el desempeño del estudiante.

Pueden mencionarse tres métodos en los cuales se basa la estrategia del módulo tutorial:

- a.- Método de preceptor.
- b.- Método socrático.
- c.- Diagnóstico.

En el método de preceptor el alumno adquiere ciertas habilidades a través de su interacción con algún juego o actividad. Esto le permitirá desarrollar cierta destreza para la solución de problemas. El método socrático consiste en motivar al estudiante a razonar y hacer críticas acerca de su propio conocimiento. Esto permite que el alumno modifique sus concepciones, cuando éstas sean erróneas, o las reafirme en caso contrario. Con el método de diagnóstico el sistema hace una depuración sobre las concepciones erróneas del alumno.

### 3.2.3.- EL MÓDULO DE REPRESENTACION DEL ESTUDIANTE.

En este módulo se tiene almacenada la información acerca del estudiante individual. La comprensión del estudiante, respecto a lo que se le está enseñando, se representa en este módulo. Con la representación del conocimiento del estudiante es posible que el sistema elabore hipótesis acerca de las concepciones erróneas del alumno, así como de las estrategias que éste utiliza. Con ello, el sistema tendrá elementos para hacer sugerencias al alumno. Otras aplicaciones de este módulo son el posible planteamiento del nivel de entendimiento del alumno, así como de su particular manera de aprender.

Podríamos decir que se tienen 4 tipos de fuentes de información, para la representación del conocimiento del estudiante. Estos son:

- 1.- Estructural.  
La dificultad del material que se presenta.
- 2.- Histórico.  
Las aceveraciones respecto a la experiencia del alumno.
- 3.- Explícito.  
Las preguntas del alumno.
- 4.- Implícito.  
El comportamiento del alumno ante el sistema.

El hacer una representación del conocimiento del estudiante implica :

- Tener un patrón, con el cual se compararán las respuestas del estudiante.
- Utilizar "banderas" sobre la estructura que representa el conocimiento de la materia enseñada, que indiquen que conceptos son comprendidos por el estudiante y cuales no.

Los primeros sistemas de ICAI no utilizaban un modelo sofisticado del estudiante, sino que se limitaban a utilizar únicamente las "banderas" o indicadores de conceptos conocidos y desconocidos por el alumno. A este tipo de modelado se le denomina MODELADO DE COBERTURA.

Otro tipo de modelado es el MODELADO DE HABILIDADES, con el cual se prueba la destreza del alumno, de acuerdo a una estructura de representación de conocimiento referente a un

conjunto de habilidades que el sistema propone. También se tiene un MODELADO DE ERRORES, que consiste en registrar las concepciones erróneas del estudiante, codificadas de alguna manera.

### 3.2.4.- EL MODULO DE COMUNICACION.

Un sistema efectivo requiere de una comunicación inteligente con el usuario, en este caso el alumno. Una comunicación efectiva no implica, necesariamente, una interfaz a lenguaje natural sino determinar lo que, tanto el sistema como el estudiante, quieren realmente comunicar. Esto es un verdadero problema en el momento en que el alumno se refiere al conocimiento de distinta manera a como lo hace el sistema.

La función del módulo de comunicación es generar un diálogo correcto gramática y semánticamente, ayudando así a las funciones del módulo tutorial e interpretando las respuestas del alumno, de manera que el módulo tutorial pueda trabajar con ellas ( Woolf y McDonald 1984 ).

Son pocos los sistemas que tienen, como componente, un módulo inteligente para la comunicación entre el sistema y el estudiante. Uno de ellos es GUIDON.

GUIDON hace una selección de diálogos apropiados para ser utilizados en el momento propicio, basándose en las inferencias del sistema y en el desarrollo de la sesión, de acuerdo a lo que el estudiante ya conoce o lo que se ha inferido con el sistema.

### 3.3.- VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS DE ICAI.

---

Los sistemas inteligentes de instrucción asistida por computadora, presentan una serie de ventajas, así como también algunos problemas y limitaciones. Entre las ventajas de estos sistemas están:

- a.- La modularidad e independencia de los componentes del sistema, los cuales mencionamos anteriormente.
- b.- La ayuda que presentan este tipo de sistemas a las investigaciones referentes al proceso de aprendizaje en el ser humano. Con el uso de estos sistemas puede investigarse como los estu-

diantes aprenden y como hacer mas eficiente y eficaz la ense#anza.

- c.- A diferencia de los sistemas de CAI tradicional, presentan estrategias de ense#anza de acuerdo a la situación particular del estudiante en un momento dado.
- d.- Su estructura presenta un modelo para los futuros sistemas educacionales basados en computadoras.
- e.- Tienen un mayor potencial, respecto a otros sistemas, para ofrecer discernimientos acerca de los componentes del proceso ense#anza-aprendizaje.

Entre los problemas y limitaciones de los sistemas ICAI podemos mencionar los siguientes :

- a.- Los estudios e investigaciones en ICAI estan aún en su etapa de germinación. Son pocos los proyectos y las referencias que pueden encontrarse en este campo.
- b.- Existe el eterno problema de los recursos económicos, junto con la necesidad de un mayor poder de cómputo para la implementación de estos sistemas. A pesar de la existencia de las computadoras personales, se costo es aún alto para los consumidores.
- c.- Se presenta el problema del diálogo entre el alumno y la computadora. Preferentemente se plantea el utilizar una interfaz a lenguaje natural, sin embargo, la mayoría de los sistemas actuales sólo presentan interfaces restringidas, dada la complejidad de implementación de un módulo de comprensión de lenguaje natural.
- d.- Los requerimientos, tanto de hardware como de software. Varios sistemas de ICAI requieren de máquinas que proporcionen un poderío en el manejo de lenguajes como LISP. La postura actual es la de utilizar máquinas de procesamiento paralelo. Esto, naturalmente, incrementará el poderío de los sistemas, pero también su costo. Sin embargo el problema económico, dado el decremento natural en el costo del hardware, con el tiempo se resolvería.
- e.- La mayoría de los sistemas de ICAI se han limitado a la ense#anza de materias cuyo contenido, por decirlo de alguna manera, esta estructurado, tales como matemáticas, electrónica, juegos, etc. Es necesario probar y utilizar este tipo de sis-

temas en otros dominios.

- f.- El esfuerzo y tiempo necesarios, para el diseño, construcción y desarrollo de este tipo de sistemas, son altos. Se habla, para sistemas grandes, de varias personas-año.

### 3.4.- EXPERIENCIAS CON SISTEMAS TUTORIALES.

---

En el artículo correspondiente a (Anderson y Reiser 1985), los autores indican que se han elaborado una serie de estudios respecto al uso de programas tutoriales. Dichos estudios indican una mayor efectividad de este tipo de sistemas en relación a la instrucción utilizando el salón de clases.

También se hace referenmcia a ( Bloom 1984 ), en donde se presenta que "psicólogos y educadores han observado que el uso de programas tutoriales de enseñanza individual presenta ventajas, respecto al uso del salón de clases". Se habla de un estudio que comparó a un conjunto de estudiantes, quienes estuvieron interactuando durante un tiempo específico con sistemas tutoriales, con otro conjunto de alumnos que recibieron su instrucción en el aula. Se indica que el 98% (!) de los estudiantes que utilizaron los sistemas tutoriales presentaron mejores resultados, respecto a los muchachos que tomaron clase en el salón.

Otro estudio, referente al uso de un sistema inteligente para el aprendizaje de LISP (Anderson y Reiser 1985 ), presenta que los estudiantes que utilizaron dicho sistema necesitaron sólo de 11 horas, en promedio, para aprender los mismos conceptos que otros muchachos que recibieron su instrucción en el salón de clases, los cuales necesitaron de 43 horas.

### 3.5.- ALGUNOS SISTEMAS DE ICAI.

---

#### a.- BIP.

Sus siglas se refieren a Basic Instructional Program. Fue desarrollado en la Universidad de Stanford, en 1978, por Barr. El sistema está dedicado a la enseñanza del lenguaje BASIC.

BIP evalúa los conocimientos del alumno, en base a las herramientas de programación que éste utiliza en la solución de un problema. El sistema irá proponiendo nuevos problemas, enfocados a ampliar el conjunto de herramientas de programación que el estudiante utiliza.

El sistema plantea al estudiante una serie de problemas de programación. Cada problema propuesto tendrá un grado de dificultad específico. BIP va creando, a través de su experiencia con el estudiante, un modelo del conocimiento y la destreza de éste acerca del lenguaje. En base a este modelo, el sistema propone los problemas que el alumno deberá resolver ( Wyer 1984 ).

#### b.- BUGGY.

Elaborado en 1978, BUGGY nace como una aplicación de los principios del sistema SOPHIE. Este sistema se creó con la idea de que los maestros, generalmente, piensan que los alumnos cometen errores porque no son capaces de seguir, paso a paso, una metodología para la resolución de problemas, principalmente aritméticos. Los autores de BUGGY piensan lo contrario. Ellos creen que el niño sí es capaz de seguir un procedimiento determinado. Entonces, cómo es que el alumno se equivoca ?

Lo que sucede es que los procedimientos a seguir por el alumno son, en algunos casos, procedimientos erróneos. El estudiante puede no comprender totalmente el procedimiento que se le proporciona e "inventa" el suyo, el cual es una concepción equivocada del método verdadero y que, generalmente, no lo lleva a resultados satisfactorios.

El objeto del sistema es modelar y emitir un diagnóstico del estudiante para el aprendizaje de matemáticas básicas. Se trata de un sistema-juego tutor, que enseña al estudiante cómo localizar los errores (bugs) de sus razonamientos, en la solución de operaciones aritméticas.

El sistema presenta al alumno un conjunto de problemas aritméticos. Si el estudiante los resuelve de manera incorrecta BUGGY descubre la causa del error y se la indica al alumno. BUGGY cuenta con un sistema de explicación, utilizado para indicar el porqué el alumno comete ciertos errores aritméticos, no limitándose únicamente a indicarlos.

BUGGY es útil tanto para el alumno como para el maestro. El instructor puede practicar los diagnósticos y conocer las causas más comunes de los errores de sus alumnos, dado que el sistema

presenta ejemplos de errores y sus causas. BUGGY puede reproducir cientos de errores que pueden aparecer al resolver operaciones, como sumas y restas ( Woolf y McDonald 1984 ).

c.- EXCHECK.

Elaborado por Suppes, se trata de un sistema que analiza demostraciones matemáticas. A diferencia de otros sistemas, EXCHECK no maneja un modelo del conocimiento del alumno. Sus conclusiones, referentes al razonamiento del estudiante, se basan en el uso de una máquina de inferencia y de un módulo de experiencia ( Roberts y Park 1983 ).

d.- GUIDON.

Elaborado por Clancey en 1979, GUIDON es, sin duda, el sistema de ICAI más popular. Se trata de un sistema experto, el cual utiliza la base de conocimientos de MYCIN, para ser un tutor inteligente en el área médica. Anteriormente, para este mismo fin educativo, se utilizó al propio MYCIN, dadas las características de instrucción que tienen los sistemas expertos con un subsistema de explicación. Sin embargo la interacción de MYCIN, como tutor, era pasiva. Simplemente proponía diagnósticos. Por otra parte, en el sistema GUIDON se toma la base de conocimientos de MYCIN y se adopta a un sistema tutorial.

Guidon se implementó, primeramente, tomando las 450 reglas de las que constaba MYCIN. Esta implementación no fructificó, pues se manejaba a los diagnósticos médicos "como si fueran recetas de cocina" ( Woolf y McDonald 1984 ). Además ni los estudiantes ni los médicos hacen diagnósticos utilizando cientos de reglas. Lo que se hizo entonces fué lo siguiente : Si el conocimiento de MYCIN es un conocimiento, por decirlo así, "compilado", entonces GUIDON "decompilaría" y aumentaría la base de conocimientos con las hipótesis de diagnósticos que el estudiante utilice. Posteriormente las 450 reglas de MYCIN fueron corregidas y adicionadas, lo cual llevó también a una nueva versión de GUIDON ( Woolf y McDonald 1984 ).

El estudiante, al trabajar con GUIDON, se encuentra con un "paciente". El sistema pregunta por las reglas y diagnósticos del estudiante y los critica, en base a su conocimiento. En el sistema se tienen tanto reglas tutoriales como las reglas de diagnóstico, separadas y agrupadas jerárquicamente.

Con GUIDON se esclareció la necesidad de tener un base

cognitiva del módulo experto de los sistemas tutoriales. Se definió la importancia del uso de abstracciones relacionales y lógicas, ambas importantes en la tutela y el aprendizaje de la base de conocimientos del sistema experto.

e.- SCHOLAR.

SCHOLAR es el primer sistema ICAI. Fue creado en 1970 por Carbonell. El objetivo del sistema fue, originalmente, la enseñanza de la geografía de Sudamérica, aunque su estructura le permite ser aplicado a diversos campos. El estudiante se comunica con el sistema a través de un "lenguaje natural", cuya generación está basada en el uso de redes semánticas y en estudios de comprensión de lenguaje.

La estructura de SCHOLAR es una red, en la cual se encuentran conceptos y procedimientos. Los elementos de la red definen palabras y eventos, utilizando una estructura jerárquica. La característica por lo que SCHOLAR fue tomado en cuenta es el uso de lenguaje natural restringido. Sin embargo, el sistema no diagnóstica ni evalúa al estudiante.

f.- SELF-IMPROVING QUADRATIVE TUTOR.

Sistema desarrollado en 1979 por O'shea, en la Universidad de Leeds, Inglaterra. Como lo indica su nombre, es un tutor que se mejora a sí mismo.

El objetivo del sistema es la experimentación de estrategias de enseñanza. Consta de dos componentes principales:

- 1.- Módulo de mejoramiento del sistema, el cual modifica las reglas de producción de un programa de enseñanza.
- 2.- El programa de enseñanza, que no es más que un conjunto de reglas de producción.

Las modificaciones de las reglas de producción, de las cuales resulte una mejoría en el desempeño del alumno, serán incorporadas al sistema. Este sistema resulta ser interesante, sin embargo no se han hecho más investigaciones en base a su utilización y resultados. Un sistema semejante, anterior a éste, es el SELF IMPROVING TUTOR FOR SYMBOLIC INTEGRATION, creado por Kimball en 1973 ( Roberts y Park 1983 ).

#### g.- SOPHIE.

Elaborado por Brown, Burton y Bell en 1975. El nombre del sistema es una abreviatura de "Sophisticated Instructional Environment".

La filosofía del sistema es hacer que el estudiante adquiriera la habilidad de resolver problemas, en base a probar sus propias ideas, en lugar de estar recibiendo una instrucción por parte de la computadora. El estudiante desarrolla ideas, experimenta con ellas y, en caso de que no sean correctas, las modifica.

El objetivo es la enseñanza en el diseño y comportamiento de circuitos electrónicos, localiza las fallas que pueden presentarse en diseños erróneos. El sistema presenta al alumno un circuito con una falla específica. El estudiante, por su parte, le pide información al sistema, como serían mediciones en algunos puntos del circuito, tanto de voltajes como de corrientes. Después de esto, el alumno propone una hipótesis de diagnóstico, indicando dónde está la falla en el circuito y SOPHIE checará dicha hipótesis.

Con SOPHIE se procura guiar el aprendizaje del estudiante por medio de la interacción de éste con la computadora.

El conocimiento de SOPHIE, respecto a las posibles fallas en circuitos electrónicos, es utilizado junto con estrategias heurísticas para así contestar las preguntas del alumno. El sistema almacena las rutas de razonamiento del alumno para la solución del problema y, si es el caso, las critica y genera las suyas.

#### h.- WEST.

Sistema que tiene como objetivo el aprendizaje, por parte del alumno, del manejo de expresiones aritméticas. Esto se hace en un ambiente de juego, con el cual se pretende hacer, de manera discreta, un diagnóstico de los problemas que presenta el niño.

WEST es un preceptor para uno de los juegos del sistema PLATO. La idea es plantear al estudiante que el sistema es un juego, no una herramienta para aprender aritmética y desarrollar estrategias lógicas. Se pretende que la finalidad del alumno sea ganar el juego, no directamente aprender. El estudiante se divierte con el sistema, proponiendo libremente decisiones y estrategias en el juego, el cual también implica conocimientos de

aritmética.

El juego consiste en movimientos a través de un plano, representado en el video del computador, con la finalidad de resolver una expresión algebraica. El estudiante debe escribir y resolver dicha ecuación. En cada uno de los movimientos, la destreza del alumno es comparada con la solución planteada por el sistema. Si ambas metodologías difieren, el sistema acude en ayuda del estudiante, planteándole nuevos caminos de solución para ganar el juego.

#### i.- WHY.

Se trata de una extensión del sistema SCHOLAR. El objetivo del sistema es presentar al niño las causas de la lluvia como un conjunto de conceptos y eventos relacionados. Al igual que SCHOLAR, WHY utiliza lo que se denomina heurísticas socráticas, basadas en lógica formal, las cuales describen las estrategias que los tutores humanos utilizan para guiar un diálogo con el alumno ( Roberts y Park 1983 ).

#### j.- WUMPUS.

Creado por Goldstein y Carr en 1977. WUMPUS tiene, como finalidad el desarrollo, en el estudiante, de una habilidad en la elaboración de inferencias tanto lógicas como probabilísticas, en base a ciertos datos e información proporcionados ( Roberts y Park 1983 ).

## CAPITULO IV

### PEQUEÑO SISTEMA EXPERTO DE PROPOSITO GENERAL.

Habiendo determinado y comprendido cada uno de los de los elementos y características de los sistemas expertos se elaboró, en base al sistema propuesto en (Thompson y Thompson 1985), un sistema en lenguaje PASCAL con las características primarias de un sistema experto. El sistema que presentamos a continuación trabaja con una máquina de inferencia en base a la técnica de razonamiento "hacia atrás". Cuenta con una rutina de explicación y trabaja con una base de conocimientos definida en términos de reglas del tipo IF-THEN, la cual puede crecer de acuerdo a la experiencia del sistema en su interacción con el usuario.

Este pequeño sistema ha sido programado en PASCAL por varias razones. Entre ellas se encuentra la flexibilidad del lenguaje en la creación de estructuras de datos dinámicas, la ventaja que presenta el tener un sistema completamente modular, la facilidad en la definición de las estructuras de los archivos y la posible incursión de programadores, no necesariamente en LISP o PROLOG, en la creación de este tipo de sistemas.

Además de lo anterior, resulta interesante la creación "a pié" de un pequeño sistema experto, sin utilizar herramientas para la construcción de este tipo de sistemas. Con la construcción de este sistema en PASCAL se programaron los componentes del Sistema Experto, lo cual no sucede al utilizar lenguajes especiales para construir sistemas expertos. Esto permitió esclarecer algunos conceptos definidos anteriormente.

Le hemos denominado como un sistema experto de propósito general, dado que la base de conocimientos del sistema puede cambiarse modularmente por cualquier otra, definida a través del programa de adquisición de conocimiento.

#### 4.1.- CARACTERISTICAS GENERALES.

La base de conocimientos del sistema consta de:

- 1.- Conceptos.
- 2.- Reglas.

Los conceptos deberán representar hechos ú objetos a los cuales se les pueda asignar valores, así como proposiciones que puedan ser falsas o verdaderas. Algunos conceptos podrían ser:

- " El color de la casa "
- " El número de caras del cuerpo "
- " El año del acontecimiento "
- " Tiene clima tropical "
- " Profesan alguna religión "
- " Su religión "

El sistema, para no utilizar demasiada memoria, maneja los conceptos a través de un número de referencia y de un nemónico. Las frases anteriormente presentadas serían las interpretaciones de los nemónicos siguientes:

color-casa  
numero-caras  
año  
tropical  
hay-religion  
religion

A los conceptos se les asignarán valores. De acuerdo a los ejemplos de conceptos antes presentados se podría pensar en los valores siguientes:

rojo  
seis  
1492  
si  
no  
budista

Los valores que un concepto puede tomar serán cadenas alfanuméricas de hasta 15 caracteres. Para poder generar un diálogo con el usuario cada uno de los conceptos tiene una pregunta asociada. Para nuestro ejemplo las preguntas podrían ser :

- " Cual es "
- " Cual es "
- " Cual es "

" Sí o no "  
" Sí o no "  
" Como se llama "

Las reglas tienen la estructura siguiente:

regla ::= SI antecedente ENTONCES consecuente  
antecedente ::= proposicion ( Y proposicion )  
consecuente ::= proposicion  
proposicion ::= concepto ES valor

Un ejemplo de regla podría ser :

SI  
    el color de la casa ES rojo  
Y  
    profesan alguna religion ES si  
ENTONCES  
    su religion ES budista

#### 4.2.- CONFIGURACION DE LOS ARCHIVOS.

-----

El sistema utiliza cuatro archivos para el almacenamiento de datos. Estos archivos son :

- a. - Archivo de preguntas.
- b. - Archivo de conceptos.
- c. - Archivos de antecedentes.
- d. - Archivo de reglas.

a.- Archivo de preguntas.

Se trata de un archivo de cadenas alfabéticas de 40 caracteres de longitud. Se tienen almacenadas las preguntas correspondientes a cada uno de los conceptos definidos. Al archivo de preguntas, dentro del sistema, se le hace referencia bajo el nombre de QUESTIONS.

b.- Archivo de conceptos.

Este archivo contiene, en cada uno de sus registros:

- 1.- El nombre del concepto.
- 2.- La interpretación del concepto.

El nombre del concepto es una cadena alfabética de 15 caracteres de longitud. Se trata de un nemónico con el cual se denomina al concepto. El significado o explicación del concepto se tiene en una cadena alfabética de 40 caracteres. Al archivo de conceptos se le denominó OBJETOS en el sistema.

#### c.- Archivos de antecedentes.

Se tendrá un archivo de causalidad para cada una de las reglas definidas en el sistema. Este archivo consta de todos los conceptos que intervienen en el antecedente de la regla, junto con sus valores correspondientes. Se tienen almacenados:

- 1.- El número de concepto.
- 2.- El valor del concepto.

El número del concepto es un entero, mientras que el valor del concepto es una cadena de 15 caracteres alfabéticos. A estos archivos, en el programa, se manejan bajo el nombre de CAUSALIDAD.

#### d.- Archivo de reglas.

En este archivo se tiene almacenado, en cada registro, el consecuente de cada regla y el nombre físico del archivo de causalidad donde se encuentra su antecedente. Se tienen almacenados:

- 1.- El número de concepto del consecuente.
- 2.- El valor correspondiente a dicho concepto.
- 3.- El nombre del archivo de causalidad.

En el programa nos referimos a este archivo bajo el nombre de RULES.

#### 4.3.- CONFIGURACION DE LOS DATOS EN MEMORIA.

---

Se tienen tres estructuras principales para el manejo de los datos en memoria. Dos de ellas representan el ESPACIO DE TRABAJO del Sistema Experto, mientras la tercera se refiere a la BASE DE

## CONOCIMIENTOS.

### \* Espacio de trabajo:

- a.- Pila de contexto.
- b.- Pila de metas.

### \* Base de conocimientos:

- c.- Arreglo de reglas.
- d.- Arreglo de conceptos.

### a.- Pila de contexto.

Se tiene una pila, o stack, en el cual se almacenan las proposiciones que se han considerado como válidas, en el desarrollo de un problema específico por parte del sistema. La pila contendrá, en cada uno de sus elementos, la siguiente información:

- 1.- Número del concepto.
- 2.- Valor del concepto.
- 3.- Campo de indicación booleano.

En esta pila de contexto se tienen almacenados aquellos conceptos, con sus valores correspondientes, que se consideran válidos, ya sea por que el usuario los haya proporcionado o porque el sistema los haya inferido. En el campo de indicación se tiene registrado si la proposición fué inferida por el sistema, o si la dió el usuario como un dato válido.

El hecho de tener almacenadas las proposiciones válidas, para un estado particular del problema, es útil para determinar cuáles reglas siguen siendo vigentes y cuáles deben descartarse por contradecir lo que el sistema supone como verdadero.

El objeto de tener una pila, para almacenar la proposiciones válidas, es aprovechar la estructura LIFO (ultima-entrada, primera-salida), presentando una posible explicación de cómo el sistema ha llegado a una conclusión. Esto se logra simplemente sacando cada una de las proposiciones de la pila, en el orden inverso en que fueron concebidas.

A la pila de contexto, en el programa experto, se le denominó como CONTEX.

### b.- Pila de metas.

En la pila de metas el sistema almacena los conceptos cuyo valor debe determinarse. Dada la configuración de las reglas, para determinar el valor de un concepto, puede ser necesario obtener los valores de otros conceptos, por lo que puede hablarse de una meta principal y un conjunto de metas secundarias. El orden en que dichas metas se van presentando es registrado en el stack, por lo que las últimas metas que se presentan son las primeras en resolverse.

Los elementos almacenados en la pila de metas son apuntadores a parejas concepto-valor, las cuales constituyen el antecedente de las reglas. Se trata de un stack de apuntadores. De tal manera se tiene almacenado el concepto cuyo valor debe definirse y a la vez el antecedente que debe probarse.

De esta manera, con la pila de contexto y la pila de metas, el sistema tiene una representación de la situación del problema en un momento dado, teniendo las proposiciones que se han determinado como válidas y las metas que se deben resolver. En el programa experto se hace referencia a la pila de metas bajo el nombre de METAS.

### c.- Arreglo de reglas.

Esta es la estructura en la cual se tiene el conocimiento del sistema en memoria. Se trata de un arreglo de reglas, donde cada uno de los elementos del arreglo tiene, como información, tres campos :

- 1.- Cabeza de lista de antecedentes.
- 2.- Cabeza de lista de consecuentes.
- 3.- Campo de validez.

Debido a que el antecedente de una regla puede conformarse por una o varias condiciones, no importa el número, la regla debe ser flexible de acuerdo al número de condiciones. Por ello se pensó, como se indica básicamente en (Thompson y Thompson 1985), en tener una lista dinámica de elementos ligados para representar el antecedente de la regla. De tal manera no hay un desperdicio de espacio en memoria, si se diera el caso de reglas con pocas condiciones en sus antecedentes. Por ello, en el elemento del arreglo correspondiente a la regla, se tiene la cabeza o apuntador inicial de la lista del antecedente. Sucede lo mismo con el consecuente. Sin embargo, en este caso, el sistema actualmente sólo utiliza un solo consecuente para cada regla.

Posteriormente se piensa en enriquecer las reglas utilizando más de una proposición como conclusión, y es por ello que la estructura la hemos dejado así.

Por otro lado, es necesario conocer cuando una regla resulta coherente con la situación del problema, o sea, con la pila de contexto. Para ello se tiene el campo de validez, que es un booleano que indica si la regla es válida o falsa.

En algunas ocasiones, a través de la rutina APRENDE, el sistema adiciona reglas a la estructura, incrementando su conocimiento en base a su experiencia. En la ( fig. No.6 ) presentamos, a manera de esquema, la arquitectura de la estructura para la base de conocimientos del sistema. A este arreglo se le denominó REGLAS en el programa.

#### d.- Arreglo de conceptos.

En las reglas los antecedentes y consecuentes son parejas de concepto-valor, las cuales representan proposiciones. Los valores asignados a los conceptos pueden ser cualquiera, pero los conceptos deben de estar definidos como los "átomos" de la base de conocimientos. Para ahorro de memoria, en el arreglo de reglas, se hace referencia al concepto a través de un número y no de su nemónico. Cada concepto tiene un número asignado, el cual se le da secuencialmente al momento de definirse. Cada elemento de arreglo de conceptos cuenta con dos campos:

- 1.- nemónico del concepto.
- 2.- número de referencia.

#### 4.4.- CONFIGURACION DE LOS PROGRAMAS DEL SISTEMA.

---

El sistema se compone de dos programas. Entre los dos conforman los componentes del sistema experto:

- 1.- Programa de adquisición de conocimiento.
  - Subsistema de adquisición de conocimiento.
- 2.- Programa de inferencia, explicación y aprendizaje.
  - Máquina de inferencia.

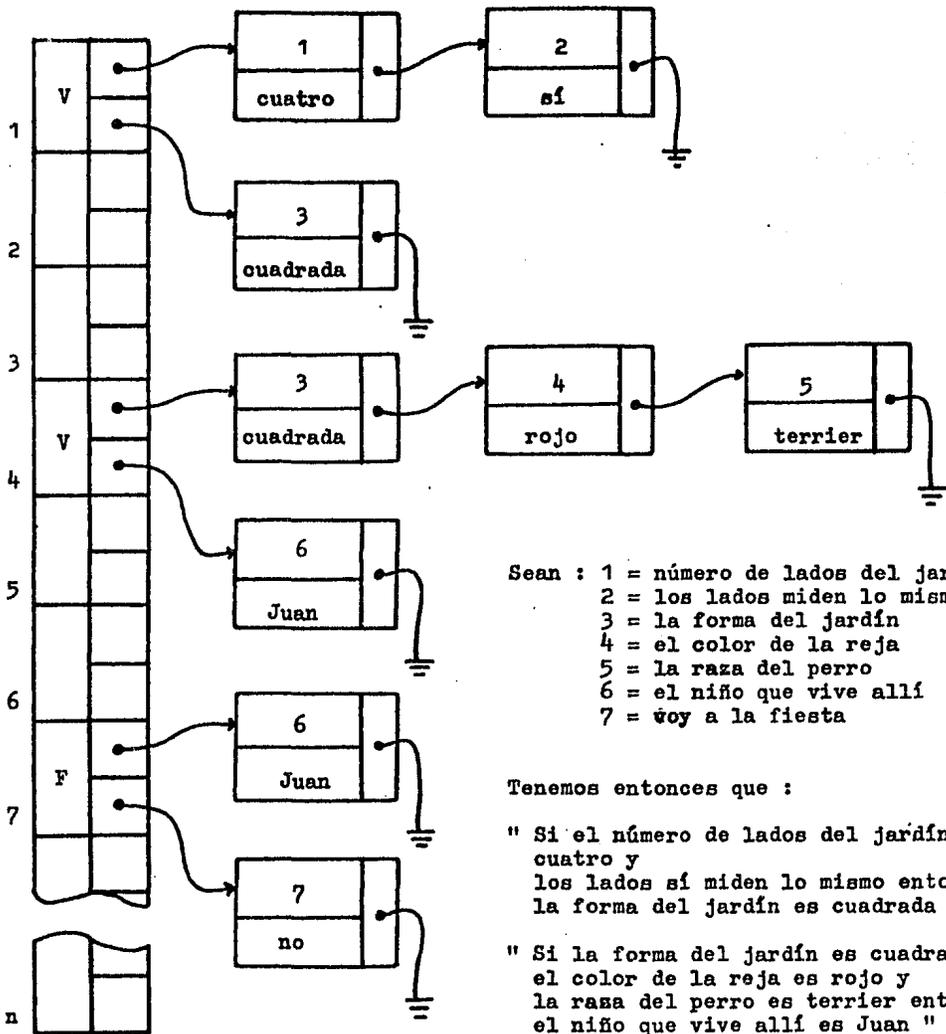


FIGURA No. 6

Configuración y ejemplo del arreglo de reglas.

- Subsistema de explicación.
- Interfaz con el usuario.

#### 4.4.2.- PROGRAMA DE ADQUISICION DE CONOCIMIENTO.

Este programa consta de varios módulos los cuales, a su vez, pueden constar de otros :

- ADQUISICION
  - MENU
    - RAYA
  - TOMATRIBUTOS
    - RAYA
    - LEE
  - LEEATRI
  - SORT
  - TOMAREGLAS
    - LEE
    - ESCRIBE
    - BORRA
    - BUSCA
    - LIMPANTALLA

Presentamos, brevemente, las funciones de cada uno de ellos:

- ADQUISICION .... El programa principal tiene como función:
- 1.- Invocar la presentación del menú de opciones.
  - 2.- Si la opción es adicionar conceptos se invoca al procedimiento TOMATRIBUTOS.
  - 3.- En cambio, si la opción es adicionar reglas entonces se llama a LEEATRI, SORT y TOMAREGLAS en ese orden.
  - 4.- Se regresa al inciso 1 hasta que la opción sea finalizar.
- MENU ..... Presenta el menú de opciones, entre adicionar conceptos, adicionar reglas o finalizar.
- TOMATRIBUTOS ... Procedimiento para la captura de cada uno de los conceptos. El experto humano indica el nemónico, la interpretación y la pregunta asociada a cada concepto. Estos son grabados en los archivos de conceptos y de preguntas.

- RAYA ..... Simplemente dibuja una línea continua en la pantalla, en el renglón que se indique.
- LEE ..... Pequeña rutina para leer opciones cuyos valores son 's', 'S', 'N' o 'n', referentes a preguntas cuya respuesta sea Sí o No.
- LEEATRI ..... Se leen, del archivo de conceptos, cada uno de ellos y se almacenan en un arreglo. También se genera un arreglo de las interpretaciones de los conceptos.
- SORT ..... Ordenamiento "quicksort" para ordenar el arreglo de conceptos de acuerdo a su nemónico.
- TOMAREGLAS ..... Rutina de captura de las reglas que se quieran definir en la base de conocimientos, grabándolas en el archivo de reglas.
- ESCRIBE ..... Procedimiento que presenta, en pantalla y para referencia, los conceptos que se tienen definidos junto con sus interpretaciones.
- BORRA ..... Procedimiento que borra una porción de la pantalla para evitar la sobreposición de mensajes.
- BUSCA ..... Rutina de búsqueda binaria. Se usa para encontrar un concepto en el arreglo, confirmando así su existencia. Únicamente pueden elaborarse reglas con conceptos previamente definidos.
- LIMPANTALLA .... Procedimiento que borra una parte de la pantalla con el fin de presentar otra lista de conceptos.

Un diagrama modular, del programa de adquisición de conocimiento, puede verse en la ( Fig. No.7 ).

#### 4.4.3.- PROGRAMA DE INFERENCIA, EXPLICACION Y APRENDIZAJE.

Este programa contiene las funciones correspondientes a:

- La máquina de inferencia.
- El subsistema de explicación.
- La comunicación con el usuario.
- El aprendizaje del sistema.

Los módulos que constituyen al programa son:

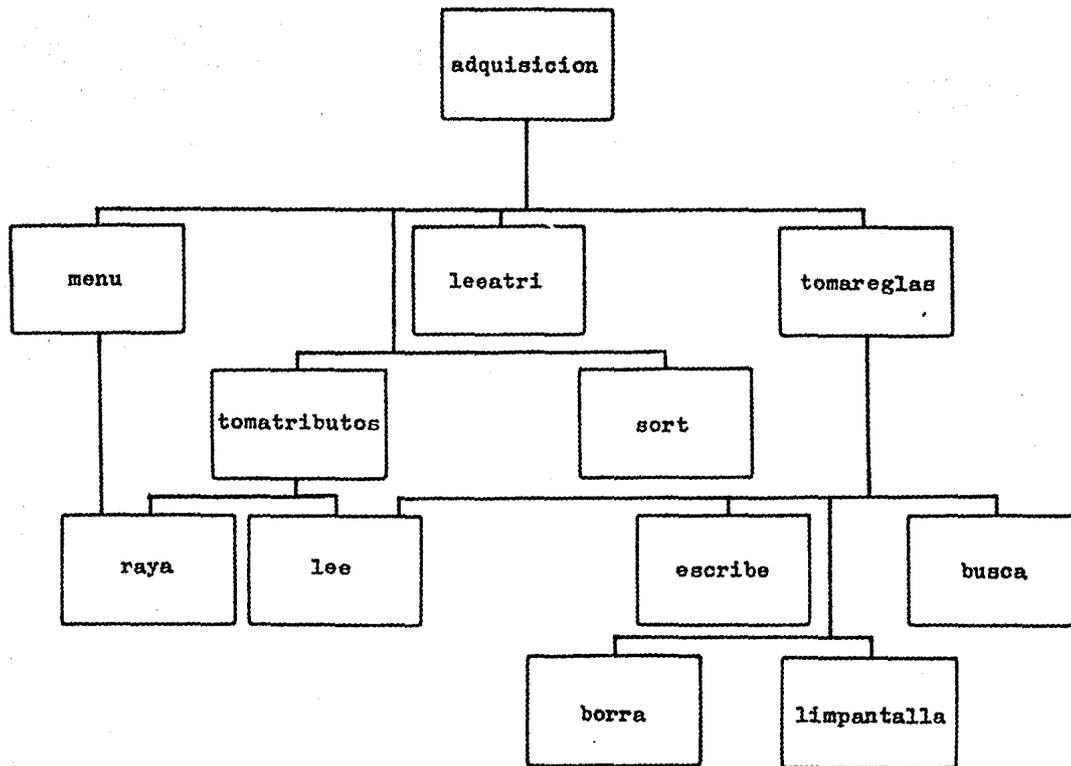


FIGURA No. 7

Diagrama modular del programa de adquisición de conocimiento.

- EXPERTO
  - LEEATRI
  - LEEREGLAS
    - GENERAREGLAS
    - LLENAPU
  - SORT
  - HAYCONSULTA
    - LEE
  - LEEMETA
    - ESCRIBEYLEE
      - RAYA
      - IMPRIME
    - BUSCA
  - INFERENCIA
    - PUSHMETA
    - ENCONTRAR
    - VALDR
    - NOHAYMETAB
    - DESCARTE
    - JUSTIFICA
    - PUSHCONTEX
    - PREGUNTAR
  - HAYMETA
  - EXPLICA
  - APRENDE
    - PREGUNTAR
    - OTROSITEMS
    - CREAANTECE
  - RENUEVA
  - PRESENTA
    - PON

A este programa se le denominó EXPERTO. Al igual que para el programa de adquisición de conocimiento, presentamos brevemente las funciones de cada uno de los módulos del programa EXPERTO :

- EXPERTO .... El programa principal :
- 1.- Invoca a la rutina de lectura de atributos.
  - 2.- Invoca a la rutina de lectura de reglas.
  - 3.- Ya con la base de conocimientos definida manda ordenar el arreglo de conceptos.
  - 4.- Mientras haya consultas :
  - 5.- Leerá la meta principal.
  - 6.- Accionará a la máquina de inferencia.
  - 7.- Si llega a inferir una meta, entonces tendrá que explicar como se llegó a esa conclusión.
  - 8.- Si no llega a inferir nada, entonces aprende como llegar a concluir la meta final.
  - 9.- Renovara las reglas, haciéndolas verdaderas para

regresar al inciso 4.

10.- Finalmente presenta las reglas de la base de conocimientos.

- LEEATRI ..... Procedimiento que genera un arreglo de elementos que contienen el número de concepto y su nemónico, así como un indicador de si el concepto se encuentra como consecuente en alguna regla. Los datos son leídos del archivo de objetos.
- LEEREGLAS ... Procedimiento que controla la generación la estructura de representación de conocimiento. Esta estructura es el arreglo de reglas. La estructura se genera dinámicamente como un arreglo de listas ligadas. Cada antecedente de las reglas es una lista de parejas concepto-valor, generada dinámicamente. El consecuente, por ahora, es una pareja concepto-valor y puede ser también una lista ligada de dichas parejas.  
Los datos para la generación de la estructura se leen del archivo de reglas.
- GENERAREGLAS Este procedimiento es invocado por LEEREGLAS y tiene como función la generación de una regla específica. Para ello se leen las condiciones de la regla en el archivo de antecedentes correspondiente a esta.
- LLENAPU ..... Procedimiento utilizado por GENERAREGLAS, que tiene como función la asignación de los datos para una pareja concepto-valor, la cual corresponde a una condición del antecedente.
- SDORT ..... Procedimiento de ordenamiento "quicksort". Ordena el arreglo donde se tienen almacenados los nemónicos y los números de los conceptos. Se trata de un ordenamiento alfabético de los conceptos, respecto al nemónico correspondiente.
- HAYCONSULTA . Esta función booleana se torna verdadera cuando el usuario contesta que sí quiere consultar el sistema. La función hace la pregunta y recibe la respuesta del usuario.
- LEE ..... Pequeña rutina utilizada para obtener respuestas de tipo "Sí" o "No" por parte del usuario.
- LEEMETA ..... Procedimiento que tiene como fin la lectura de la meta principal que tendrá el sistema. Controla la ejecución de la rutina ESCRIBEYLEE.
- ESCRIBEYLEE Procedimiento que invoca a la rutina que despliega

los conceptos involucrados en los consecuentes de las reglas y pregunta de que concepto se va a determinar su valor. El usuario proporciona dicho concepto, determinándose así la meta principal.

- IMPRIME ....** Rutina que presenta en pantalla los nemónicos e interpretación de los conceptos que se encuentran involucrados como consecuentes en las reglas, lo cual implica que su valor puede ser conocido a través de las mismas.
- BUSCA .....** Procedimiento de búsqueda binaria sobre el arreglo de conceptos, para determinar si el concepto indicado por el usuario se encuentra definido.
- INFERENCIA** Procedimiento correspondiente a la máquina de inferencia del sistema experto. El procedimiento se basa en un "encadenamiento hacia atrás", comentado anteriormente en el capítulo segundo cuando hablamos sobre las máquinas de inferencia. En términos generales la rutina hace lo siguiente:
- a.- Toma la meta principal.
  - b.- Busca una regla cuyo consecuente se refiera al concepto que se tiene como meta.
  - c.- Si encuentra la regla la evalúa.
  - d.- Si la regla es cierta eso implica que el sistema ha inferido una conclusión en base a la regla, y dicha conclusión se guarda en la pila de contexto. Se elimina dicha meta de la pila de metas.
  - e.- Si la regla es falsa, se descarta.
  - f.- Si la regla es indeterminada, implica que alguna de las condiciones no se ha determinado, por lo cual el concepto de dicha condición pasará a ser la siguiente meta.
  - g.- Si no se encontró ninguna regla, el sistema no podrá inferir el valor del concepto, por lo que tendrá que preguntárselo al usuario. Antes de plantear la pregunta el sistema justifica su comportamiento.
  - h.- Se regresa al paso "b" mientras la meta principal no se haya logrado.
- PUSHMETA ....** Este procedimiento tiene como objetivo el poner un nuevo apuntador, a parejas concepto-valor, en la pila de metas.
- ENCONTRAR ...** Esta función booleana se encarga de buscar, en la base de conocimientos, aquella regla que tenga, como consecuente, al concepto involucrado en el tope de la pila de metas. Las reglas se analizan desde la última regla definida hasta la primera, o sea, tomando en

cuenta primero las últimas experiencias.

VALOR ..... Esta función, que evalúa a la regla en cuestión, proporciona tres valores posibles :

- cierta
- falsa
- indeterminada

Una regla es cierta si todas las parejas concepto-valor de su antecedente se encuentran también en la pila de contexto.

Una regla es falsa si una o más de las parejas concepto-valor del antecedente contradicen alguna proposición de la pila de contexto. Esto no es sino determinar si en la pila de contexto se encuentra el mismo concepto que en el antecedente, pero con un valor diferente.

Finalmente, una regla es indeterminada si alguno de los conceptos involucrados en el antecedente de la regla, no se encuentra en la pila de contexto, por lo cual definir el valor de dicho concepto se convertirá en la siguiente meta.

- NOHAYMETAS . Función booleana que determina si ya no hay elementos en la pila de metas.
- DESCARTE ... Procedimiento simple que marca como inválida a una regla evaluada como falsa. La regla no se tomará en cuenta en lo que resta de la sesión.
- JUSTIFICA .. El sistema, al no poder inferir una conclusión, debe preguntarle al usuario el valor del concepto indicado como meta. Esta rutina indica al usuario el "para qué" se necesita conocer el valor del concepto.
- PREGUNTAR .. Rutina que hace la pregunta al usuario. Se utiliza el archivo de preguntas, generado en el programa de adquisición de conocimiento.
- PUSHCONTEX . Rutina que adiciona una pareja concepto-valor en la pila de contexto, ya sea porque el valor se infirió o porque el usuario lo haya proporcionado.
- HAYMETA .... Función booleana que determina si el sistema llegó a inferir el valor del concepto indicado como meta principal.
- EXPLICA .... Procedimiento utilizado para que el sistema de una explicación al usuario de porque llegó a inferir una conclusión específica. Esto se hace tomando las parejas concepto-valor almacenadas en la pila de contexto y haciendo referencia si dicho valor lo dió el usuario o lo obtuvo el sistema.

- APRENDE ....** Este procedimiento tiene como objetivo la generación de nuevas reglas en la base de conocimiento. La rutina manda preguntar el valor del concepto al usuario, así como también los valores de otros conceptos que, de acuerdo a las reglas, puedan ser necesarios para definir el concepto en cuestión. La rutina va creando el antecedente de la regla, tanto en el arreglo de reglas como en archivo.  
El consecuente de la regla será la pareja concepto-valor determinada por el usuario, mientras que el antecedente será conformado por las parejas concepto-valor que se encuentren en la pila de contexto; no todas las parejas concepto-valor serán tomadas en cuenta, solamente se considerarán la última que se haya sido inferida por el sistema y las siguientes proporcionadas por el usuario, hasta llegar al tope de la pila.
- CREANTECE ..** Procedimiento invocado por APRENDE. Su función es llenar de datos las parejas concepto-valor generadas como antecedente de la nueva regla, grabando los datos tanto en el archivo de antecedentes de la nueva regla como en el arreglo de reglas.
- OTROSITEMS .** Rutina que permite al usuario el adicionar condiciones a las que se han determinado, a lo largo de la sesión, como elementos que conformen el antecedente de la nueva regla. Dichas condiciones adicionales son las que el sistema conoce que pueden ser necesarias para la determinación de el valor del concepto, dado a que aparecen como condiciones en la regla que tiene, como consecuente, al concepto correspondiente de la conclusión final.
- RENUEVA ....** Rutina que define a todas las reglas como válidas, determinando como verdadero el campo de validez de los elementos del arreglo de reglas. Esto se hace debido a que, a lo largo de la sesión, algunas reglas se descartan. El descarte de las reglas consiste en hacer falso dicho campo de validez.
- PRESENTA ...** Procedimiento que presenta cada una de las reglas que conforman a la base de conocimientos.
- PON .....** Rutina que escribe proposiciones, presentando la interpretación del concepto y su valor correspondiente.

Un diagrama modular de este programa se presenta en las ( Fig. No.8 ) y ( Fig. No.9 ).

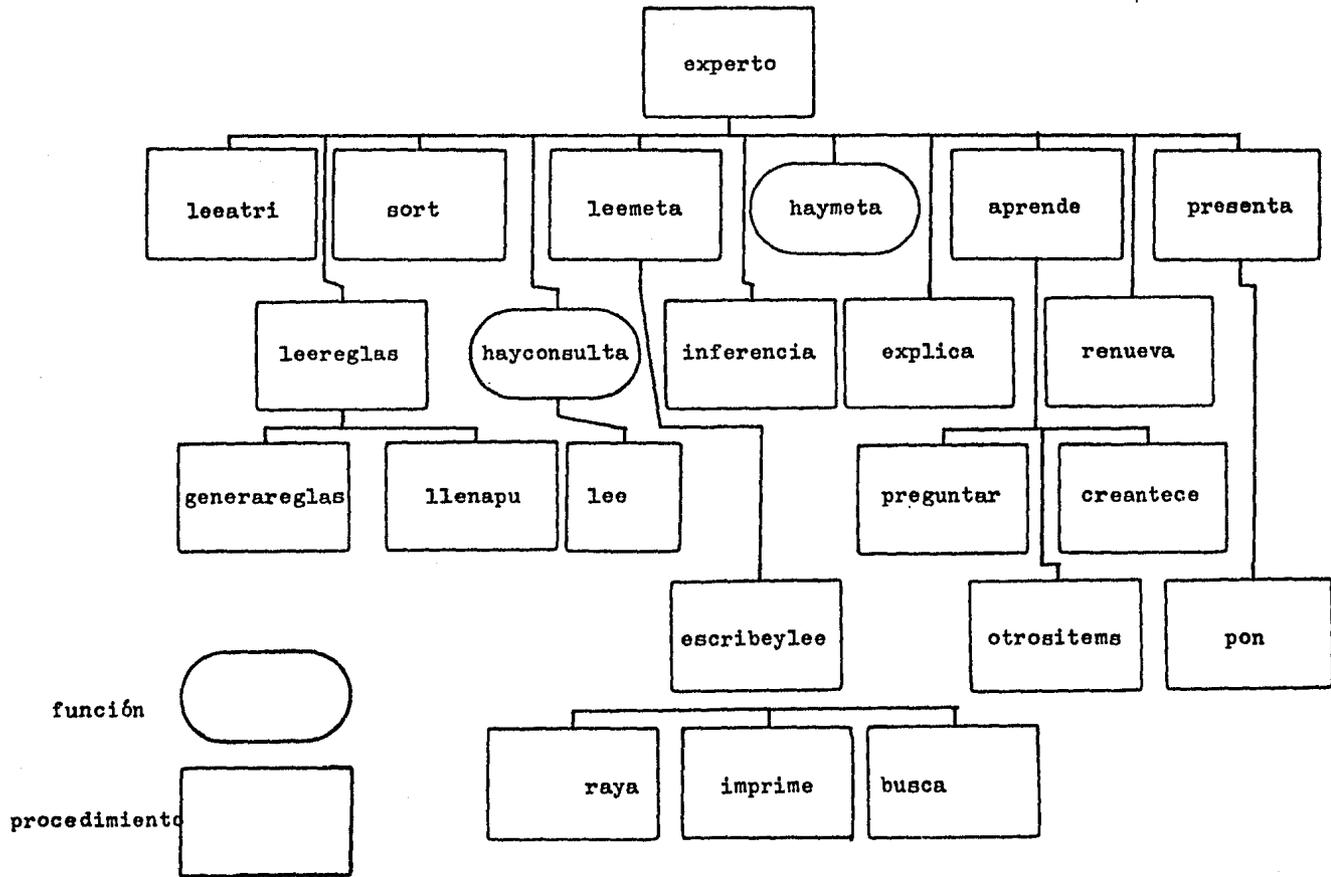


FIGURA No. 8

Diagrama modular del programa de inferencia, explicación y aprendizaje.

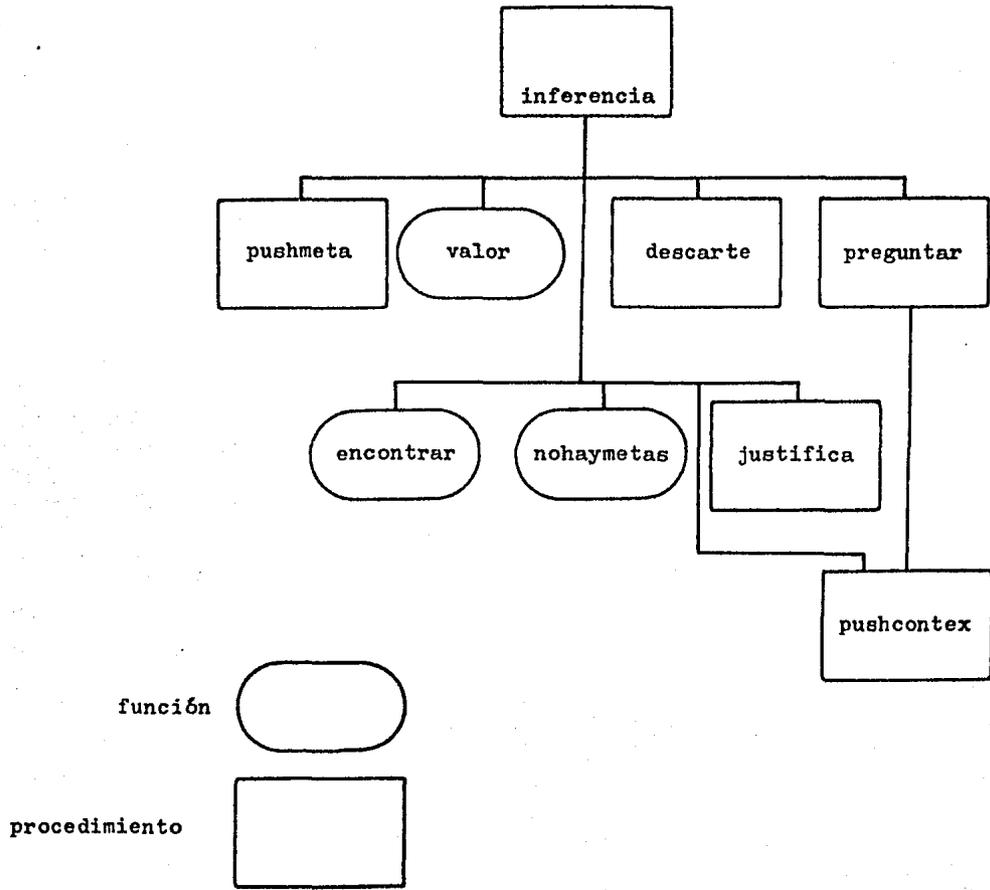


FIGURA No. 9

Diagrama modular del procedimiento de inferencia.

A continuación presentamos una sesión de ejemplo de nuestro pequeño sistema experto de propósito general, utilizándolo como un sistema constructor.

El estado inicial de la base de conocimiento es el siguiente conjunto de reglas :

\*\*\*\*\* regla # 1 \*\*\*\*\*  
SI la forma de su jardín es cuadrada  
y el idioma que se habla en su casa es español

ENTONCES el color de su casa es rojo

\*\*\*\*\* regla # 2 \*\*\*\*\*  
SI el número de lados de su jardín es 4  
y si miden lo mismo sus lados

ENTONCES la forma de su jardín es cuadrada

\*\*\*\*\* regla # 3 \*\*\*\*\*  
SI su país de origen es México  
y si sigue hablando su idioma natal

ENTONCES el idioma que se habla en su casa es español

\*\*\*\*\* regla # 4 \*\*\*\*\*  
SI el color de su casa es rojo  
y la raza de su perro es gran danés  
y su religión es católica

ENTONCES la posición de su casa en la cuadra es segunda

\*\*\*\*\* regla # 5 \*\*\*\*\*  
SI la posición de su casa en la cuadra es segunda  
y el sexo de la persona es femenino  
y su estado civil es soltero  
y la persona, respecto a su edad, es adolescente

ENTONCES su nombre es María

\*\*\*\*\* regla # 6 \*\*\*\*\*  
SI la edad mínima que podría tener es 12  
y la edad máxima que podría tener es 17

ENTONCES la persona, respecto a su edad, es adolescente

\*\*\*\*\* regla # 7 \*\*\*\*\*  
SI la edad maxima que podria tener es 11

ENTONCES la persona, respecto a su edad, es nino

✱

La sesión sería la siguiente:

Quieres consultar el sistema S/N) ? s

Lista de conceptos :

color ... el color de su casa  
fase-vida ... la persona, respecto a su edad,  
forma ... la forma de su jardin  
idioma ... el idioma que se habla en su casa  
nombre ... su nombre  
posicion ... la posicion de su casa en la cuadra

Dime el concepto que deseas conocer o enseñarme : nombre  
cual es el numero de lados de su jardin ? 4

Por ahora sabemos que el numero de lados de su jardin es 4  
si me dices (si/no) miden lo mismo sus lados  
podemos saber la forma de su jardin

(si/no) miden lo mismo sus lados ? si

Por ahora sabemos que la forma de su jardin es cuadrada  
si me dices cual es su pais de origen  
podemos saber el idioma que se habla en su casa

cual es su pais de origen ? mexico

Por ahora sabemos que su pais de origen es mexico  
si me dices (si/no) sigue hablando su idioma natal  
podemos saber el idioma que se habla en su casa

(si/no) sigue hablando su idioma natal ? si

Por ahora sabemos que el color de su casa es rojo  
si me dices cual es la raza de su perro  
podemos saber la posicion de su casa en la cuadra

cual es la raza de su perro ? gran danes

Por ahora sabemos que la raza de su perro es gran danes  
si me dices cual es su religion  
podemos saber la posicion de su casa en la cuadra

cual es su religion ? catolica

Por ahora sabemos que la posicion de su casa en la cuadra es segunda  
si me dices cual es (femenino, masculino) el sexo de la persona  
conoceremos su nombre

cual es (femenino, masculino) el sexo de la persona ? femenino

Por ahora sabemos que el sexo de la persona es femenino  
si me dices cual es ( soltero, casado, etc ) su estado civil  
conoceremos su nombre

cual es ( soltero, casado, etc ) su estado civil ? soltero

Por ahora sabemos que su estado civil es soltero  
si me dices cual es la edad maxima que podria tener  
podemos saber la persona, respecto a su edad,

cual es la edad maxima que podria tener ? 17

Por ahora sabemos que la edad maxima que podria tener es 17  
si me dices cual es la edad minima que podria tener  
podemos saber la persona, respecto a su edad,

cual es la edad minima que podria tener ? 12

Parece ser que ....

su nombre es Maria, ademas he inferido que  
la persona, respecto a su edad, es adolescente, porque me has dicho  
que la edad minima que podria tener es 12 y que  
la edad maxima que podria tener es 17 y que  
su estado civil es soltero y que  
el sexo de la persona es femenino, ademas he inferido que  
la posicion de su casa en la cuadra es segunda, porque me has dicho  
que su religion es catolica y que  
la raza de su perro es gran danes, ademas he inferido que  
el color de su casa es rojo, ademas he inferido que  
el idioma que se habla en su casa es espanol, porque me has dicho que  
si sigue hablando su idioma natal y que  
su pais de origen es mexico, ademas he inferido que  
la forma de su jardin es cuadrada, porque me has dicho que  
si miden lo mismo sus lados y que  
el numero de lados de su jardin es 4

Quieres consultar el sistema S/N) ? s

Dime el concepto que deseas conocer o ense#arme : idioma

cual es su pais de origen ? mexico

Por ahora sabemos que su pais de origen es mexico  
si me dices (si/no) sigue hablando su idioma natal  
conoceremos el idioma que se habla en su casa

(si/no) sigue hablando su idioma natal ? si

Parece ser que ....

el idioma que se habla en su casa es espanol, porque me has dicho que  
si sigue hablando su idioma natal y que

su pais de origen es mexicoQuieres consultar el sistema S/N) ? s  
Dime el concepto que deseas conocer o ense#arme : idioma

cual es su pais de origen ? Rusia

No puedo llegar a ninguna conclusion, lo siento.  
Necesito aprender mas y quisiera que me dijeras

cual es el idioma que se habla en su casa ? ruso

Dime, para este caso, (s/n) es necesario que yo conozca  
sigue hablando su idioma natal ? s

(si/no) sigue hablando su idioma natal ? si  
Quieres consultar el sistema S/N) ? s

Dime el concepto que deseas conocer o ense#arme : forma

cual es el numero de lados de su jardin ? 3

No puedo llegar a ninguna conclusion, lo siento.  
Necesito aprender mas y quisiera que me dijeras

como es la forma de su jardin ? triangular

Dime, para este caso, (s/n) es necesario que yo conozca  
miden lo mismo sus lados ?Quieres consultar el sistema S/N) ? s

Dime el concepto que deseas conocer o ense#arme : nombre

cual es el numero de lados de su jardin ? 3

Por ahora sabemos que la forma de su jardin es triangular  
si me dices cual es el color de su casa  
podemos saber la posicion de su casa en la cuadra

cual es el color de su casa ? azul

Por ahora sabemos que el color de su casa es azul  
si me dices cual es la posicion de su casa en la cuadra  
conoceremos su nombre

cual es la posicion de su casa en la cuadra ? primera  
No puedo llegar a ninguna conclusion, lo siento.  
Necesito aprender mas y quisiera que me dijeras

cual es su nombre ? Pedro

Dime, para este caso, (s/n) es necesario que yo conozca  
el sexo de la persona ? n  
tambien debo conocer su estado civil ? n  
tambien debo conocer la persona, respecto a su edad, ? n

Quieres consultar el sistema S/N) ? s

Dime el concepto que deseas conocer o ense#arme : nombre  
cual es el numero de lados de su jardin ? 4

Por ahora sabemos que el numero de lados de su jardin es 4  
si me dices (si/no) miden lo mismo sus lados  
podemos saber la forma de su jardin

(si/no) miden lo mismo sus lados ? si

Por ahora sabemos que la forma de su jardin es cuadrada  
si me dices cual es su pais de origen  
podemos saber el idioma que se habla en su casa

cual es su pais de origen ? mexico

Por ahora sabemos que su pais de origen es mexico  
si me dices (si/no) sigue hablando su idioma natal  
podemos saber el idioma que se habla en su casa

(si/no) sigue hablando su idioma natal ? si

Por ahora sabemos que el color de su casa es rojo  
si me dices cual es la raza de su perro  
podemos saber la posicion de su casa en la cuadra

cual es la raza de su perro ? gran danes

Por ahora sabemos que la raza de su perro es gran danes  
si me dices cual es su religion  
podemos saber la posicion de su casa en la cuadra

cual es su religion ? catolica

Por ahora sabemos que la posicion de su casa en la cuadra es segunda  
si me dices cual es (femenino, masculino) el sexo de la persona  
conoceremos su nombre

cual es (femenino, masculino) el sexo de la persona ? masculino  
No puedo llegar a ninguna conclusion, lo siento.  
Necesito aprender mas y quisiera que me dijeras

cual es su nombre ? Juan

Dime, para este caso, (s/n) es necesario que yo conozca  
su estado civil ? n  
tambien debo conocer la persona, respecto a su edad, ? n  
Quieres consultar el sistema S/N) ? n

Quieres consultar el sistema S/N) ? n

Finalmente el sistema presenta el nuevo estado de la  
base de conocimientos. Presentamos las nuevas reglas  
que el sistema contiene después de la sesión :

\*\*\*\*\* regla # 8 \*\*\*\*\*

SI su pais de origen es Rusia  
y si sigue hablando su idioma natal

ENTONCES el idioma que se habla en su casa es ruso  
\*\*\*\*\* regla # 9 \*\*\*\*\*

SI el numero de lados de su jardin es 3

ENTONCES la forma de su jardin es triangular  
\*\*\*\*\* regla # 10 \*\*\*\*\*

SI la forma de su jardin es triangular  
y el color de su casa es azul  
y la posicion de su casa en la cuadra es primera

ENTONCES su nombre es Pedro  
\*\*\*\*\* regla # 11 \*\*\*\*\*

SI la posicion de su casa en la cuadra es segunda  
y el sexo de la persona es masculino

ENTONCES su nombre es Juan

## CONCLUSIONES.

=====

Hemos analizado dos de las aplicaciones, actualmente más importantes, de la computadora: como auxiliar en el proceso de enseñanza aprendizaje y como elemento de organización y manejo inteligente de conocimiento.

Estas dos aplicaciones han llegado a compartir varios puntos, objetivos y metas, de lo cual ha nacido lo que yo denominaría como la aplicación más prometedora de la computación: el uso de sistemas de inteligencia artificial para la educación. En este caso no se trata, como en otras aplicaciones de la computadora, en desarrollar sistemas que le proporcionen al hombre un mejor nivel de vida y le faciliten sus tareas diarias, lo cual ha sido muy criticado bajo los argumentos de deshumanización y desplazamiento del hombre en el proceso productivo por parte de la máquina. Aquí se trata, por otra parte, de desarrollar lo más importante para el ser humano: su pensamiento, ideas y conocimiento.

A fin de cuentas parece ser que las investigaciones, en Inteligencia Artificial en general y en Sistemas Expertos en particular, llegarán a proporcionar al ser humano un mayor y más completo conocimiento, tanto de la realidad que le rodea como de sí mismo.

Por otro lado, pienso que el uso de Sistemas Expertos en la educación debe plantearse de dos maneras distintas:

- a) Como sistemas constructores.
- b) Como sistemas tutores.

### a) COMO SISTEMAS CONSTRUCTORES.

El uso de Sistemas Expertos como sistemas constructores se refiere, en particular, al uso de la computadora como un agente que permita un aprendizaje de manera activa, creadora.

Este tipo de Sistemas Expertos Constructores tendrían la peculiaridad de que sus bases de conocimiento serían generadas por el sujeto, formándose así una especie de "micromundo" (siguiendo la idea de Papert) cuyos conceptos y reglas son determinados por el niño.

En este caso me refiero a niños en particular, ya que con estos sistemas se propone ayudar al niño en su paso de pensamiento

concreto a pensamiento formal, trabajando con conceptos y relaciones lógicas en un nivel de abstracción.

Estos sistemas presentan, como puede verse en el capítulo cuarto, una estructura de red o lattice, que es muy similar a la estructuras operatorias propias del niño que llega al pensamiento formal. Lo que distingue a esta red de una clasificación simple es la posible combinatoria de los elementos de la estructura.

Las materias ideales para apoyar con estos sistemas constructores son : lógica, matemáticas, geometría y física.

#### b) SISTEMAS TUTORES.

Como ya hemos visto en el capítulo tercero, la mayoría de los sistemas de ICAI son sistemas tutores. Podemos decir que estos sistemas presentan un cierto grado de "pasividad". Algunos de ellos son literalmente, como su nombre lo indica, sistemas de CAI inteligentes, con los cuales la interacción del sujeto con el sistema es más coherente y lógica que en los sistemas de CAI tradicional, ya que el sistema se comporta de acuerdo a las limitaciones y avances del sujeto. Sin embargo la actitud del alumno no es más creadora, sino que sigue siendo, de alguna manera, pasiva.

Este tipo de sistemas, por su naturaleza, están enfocados no sólo a niños, sino a cualquier persona en general. Podría pensarse en la capacitación de trabajadores en una planta industrial, cursos de lenguajes de computación, de historia, geografía, filosofía, literatura, idiomas, etcétera.

Por lo anterior, pienso que el uso de Sistemas Expertos en la educación es una alternativa interesante y que debe tomarse en cuenta al introducir a la computadora en el salón de clases.

Por otra parte, para finalizar y hablando ya de Sistemas Expertos en general, creo que es de importancia mencionar la necesidad de comenzar a preparar recursos humanos en el campo de la Ingeniería en Conocimiento, los cuales sean capaces de construir sistemas basados en conocimiento que ayuden al desarrollo de nuestra sociedad.

ANEXO 1  
=====

LISTADO DEL PROGRAMA DE ADQUISICION DE CONOCIMIENTO.  
-----

```
program adquisicion(input,output);
const
  limite=100;
type
  cadena=string[15];
  cond=packed record
    atr:integer;
    val:cadena;
  end;
  (*end record*)
  palabras=string[40];
  archivo=file of palabras;
  archi=file of cond;
  registro=packed record
    numconc:integer;
    valconc:cadena;
    condiciones:string[8];
  end;
  (*end record*)
  basecon=file of registro;
  conceptos=packed record
    nombre:cadena;
    interpret:palabras;
  end;
  (*end record*)
  archiatr=file of conceptos;
  concep=packed record
    nom:cadena;
    refer:integer;
    escon:boolean;
  end;
  (*end record*)
  arreglo=packed array[1..limite] of concep;
  vector=packed array[1..limite] of palabras;

(*****)
var
  inte:vector;
```

```

rules;basecon;
questions:archivo;
objetos:archiatr;
nreglas,natributos:integer;
atributos:arreglo;
op:char;

```

```

(*****
procedure lee(c,r:integer;var op:char);
begin
  repeat
    gotoxy(c,r);
    read(kbd,op);
  until (op='s') or (op='S') or (op='n') or (op='N');
end;
(*end lee*)

```

```

(*****
procedure raya(n:integer);
var
  i:integer;
begin
  gotoxy(1,n);
  for i:=1 to 80 do
    write('_');
  (*end for*)
end;
(*end raya*)

```

```

(*****
procedure menu(var op:char);
begin
  clrscr;
  raya(15);
  raya(22);
  gotoxy(15,17);
  write('C .- Adicionar Conceptos.');
```

```

  gotoxy(15,19);
  write('R .- Adicionar Reglas.');
```

```

  gotoxy(15,21);
  write('F .- Finalizar.');
```

```

  gotoxy(1,24);
  write('Opcion : ');
  repeat
    gotoxy(10,24);
    read(kbd,op);
  until (op='c') or (op='C') or (op='r') or (op='R') or
    (op='f') or (op='F');
```

```

  clrscr;
end;

```

```

(*****
procedure tomatributos;
var
  op:char;
  i:integer;
  conce:conceptos;
  pregunta:palabras;
begin
  assign(objetos,'b:atributo.dat');
  assign(questions,'b:pregunta.dat');
  reset(objetos);
  reset(questions);
  gotoxy(1,23);
  write('Vas a adicionar un nuevo concepto (s/n) ?');
  lee(45,23,op);
  raya(13);
  raya(20);
  i:=0;
  seek(objetos,filesize(objetos));
  seek(questions,filesize(questions));
  while (op='s')or(op='S') do
    begin
      i:=i+1;
      gotoxy(1,15);
      write('Concepto : .....');
      gotoxy(12,15);
      readln(conce.nombre);
      gotoxy(1,17);
      write('Su interpretacion es : .....');
      write('.....');
      gotoxy(24,17);
      readln(conce.interpret);
      gotoxy(1,19);
      write('Pregunta asociada: .....');
      write('.....');
      gotoxy(20,19);
      readln(pregunta);
      write(objetos,conce);
      write(questions,pregunta);
      gotoxy(1,23);
      write('Vas a darme mas conceptos nuevos (s/n) ? ');
      lee(42,23,op);
    end;
  (*end while*);
  close(objetos);
  close(questions);
end;
(*end tomatributos*)

(*****

```

```

procedure leeatri (var numattr:integer;var attr:arreglo;
                  var inte:vector);
var
  aux:conceptos;
begin
  reset (objetos);
  numattr:=1;
  read (objetos,aux);
  attr[1].nom:=aux.nombre;
  attr[1].refer:=1;
  inte[1]:=aux.interpret;
  while not eof(objetos) do
    begin
      numattr:=numattr+1;
      read(objetos,aux);
      attr[numattr].nom:=aux.nombre;
      attr[numattr].refer:=numattr;
      inte[numattr]:=aux.interpret;
    end;
  (*end while*)
  close(objetos);
end;
(*end leeatri*)

(*****)
procedure sort(izq,der:integer;var a:arreglo);
var
  aux1,aux2:concep;
  i,j:integer;
begin
  i:=izq;
  j:=der;
  aux1:=a[(izq+der) div 2];
  repeat
    while a[i].nom < aux1.nom do
      i:=i+1;
    while aux1.nom < a[j].nom do
      j:=j-1;
    if i <= j then
      begin
        aux2:=a[i];
        a[i]:=a[j];
        a[j]:=aux2;
        i:=i+1;
        j:=j-1;
      end;
  until i > j;
  if izq < j then
    sort (izq,j,a);
  if i < der then
    sort (i,der,a);
end;

```

```
(*end sort*)
```

```
(*****)  
procedure tomareglas(natributos:integer;atributos:arreglo;  
                  inte:vector);
```

```
var  
  efecto:registro;  
  causa:cond;  
  name:string[8];  
  causalidad:archi;  
  contador,posicion:integer;  
  op:char;  
  cade:string[4];  
  nombre :cadena;  
  esta:boolean;
```

```
(*****)  
procedure limpantalla(ri,rf:integer);
```

```
var  
  r:integer;  
begin  
  gotoxy(1,ri);  
  for r:=ri to rf do  
    write(' ':80);  
  (*end for*)  
end;  
(*end limpantalla*)
```

```
(*****)  
procedure escribe(natributos:integer;a:arreglo;inte:vector);  
var
```

```
  ya:char;  
  i:integer;
```

```
(*****)  
procedure imprime(var i:integer;lim:integer;atributos:arreglo;  
                  inte:vector);
```

```
var  
  lf,r:integer;
```

```
begin  
  limpantalla(3,18);  
  if (i+15) > lim then  
    lf:=lim-15  
  else  
    lf:=15;  
  (*end if*)
```

```

for r:=1 to lf do
  begin
    gotoxy(1,r+3);
    i:=i+1;
    write(a[i].nom:20,' ... ',inte[a[i].refer]);
  end;
(*end for*)
end;
(*end imprime*)

(* cuerpo de escribe *)
begin
  clrscr;
  raya(2);
  raya(19);
  i:=0;
  repeat
    imprime(i,natributos,atributos,inte);
    gotoxy(1,21);
    write('Deseas ver mas conceptos (s/n) ?');
    lee(34,21,ya);
  until (i >= natributos) or ( (ya='n') or (ya='N')));
end;
(*end escribe*)

(*****)
procedure borra(c,r:integer);
begin
  gotoxy(c,r);
  write(' ':15);
  gotoxy(c,r);
end;
(*end borra*)

(*****)
procedure busca(nombre:cadena;n:integer;a:arreglo;var esta:boolean;
var i:integer);
var
  j,k:integer;
begin
  esta:=false;
  k:=1;
  j:=n;
  repeat
    i:=(k+j) div 2;
    if nombre > a[i].nom then
      k:=i+1
    else
      j:=i-1;
  until ( a[i].nom=nombre ) or (k>j);

```

```

    if a[i].nom=nombre then
        esta:=true;
end;
(*end busca*)

(* cuerpo de tomareglas *)
begin
    assign(rules,'b:base.dat');
    reset(rules);
    contador:=filesize(rules);
    gotoxy(1,23);
    seek(rules,filesize(rules));
    write('Deseas adicionar una regla (s/n)? ');
    lee(35,23,op);
    escribe(natributos,atributos,inte);
    while (op='s') or (op='S') do
        begin
            contador :=contador +1;
            str(contador,cade);
            name:=concat('b:ante',cade);
            assign(causalidad,name);
            rewrite(causalidad);
            efecto.condiciones:=name;
            gotoxy(1,20);
            writeln('Condiciones : ');
            gotoxy(1,22);
            write('SI concepto = ');
            gotoxy(40,22);
            write(' ES : ');
            repeat
                repeat
                    borra(15,22);
                    readln(nombre);
                    busca(nombre,natributos,atributos,esta,posicion);
                until esta;
                causa.atr:=atributos[posicion].refer;
                borra(46,22);
                readln(causa.val);
                write(causalidad,causa);
                gotoxy(1,24);
                write('Otra condicion adicional en la regla (s/n) ?');
                lee(45,24,op);
            until (op='n') or (op='N');
            close(causalidad);
            gotoxy(1,20);
            writeln('Conclusion : ');
            gotoxy(1,22);
            write('ENTONCES concepto = ');
            gotoxy(40,22);
            write(' ES : ');
            repeat
                borra(21,22);

```

```

        readln(nombre);
        busca(nombre,natributos,atributos,esta,posicion);
until esta;
efecto.numconc:=atributos[posicion].refer;
borra(46,22);
readln(efecto.valconc);
write(rules,efecto);
gotoxy(1,24);
write('Deseas introducir otra regla (s/n) ? ');
lee(40,24,op);
limpantalla(20,24);
end;
(*end while*)
close(rules);
end;
(*end tomareglas*)

```

```

(*****
***** programa principal *****)
begin
repeat
menu(op);
case op of
'c','C': tomatributos;
'r','R': begin
assign(objetos,'b:atributo.dat');
assign(questions,'b:pregunta.dat');
leeatri(natributos,atributos,inte);
sort(1,natributos,atributos);
tomareglas(natributos,atributos,inte);
close(objetos);
close(questions);
end;
end;
(*end case*)
until (op='F') or (op='f');
end.

```

ANEXO 2

=====

LISTADO DEL PROGRAMA DE INFERENCIA, EXPLICACION Y  
-----  
APRENDIZAJE.  
-----

```
program experto(input,output);

(**** Seccion de declaraciones ****)

const
  limite=100;
  lim=100;
  limi=50;
type
  cadena=string[15];
  pointer=^item;
  item=packed record
    atr:integer;
    value:cadena;
    liga:pointer;
  end;
  (*end record*)
  bc=packed array[1..lim] of packed record
    antece,
    conse:pointer;
    valida:boolean;
  end;
  (*end record*)
  condic=packed record
    atr:integer;
    val:cadena;
    inf:boolean;
  end;
  (*end record*)
  cond=packed record
    atr:integer;
    val:cadena;
  end;
  (*end record*)
  stkcon=packed array[1..limi] of condic;
  stkmata=packed array[1..limi] of pointer;
  evaluacion=(cierta,falsa,indeterminada);
  palabras=string[40];
  archivo=file of palabras;
```

```

archi=file of cond;
registro=packed record
    numconc:integer;
    valconc:cadena;
    condiciones:string[8];
end;
(*end record*)
basecon=file of registro;
conceptos=packed record
    nombre:cadena;
    interpret:palabras;
end;
(*end record*)
archiatr=file of conceptos;
concep=packed record
    nom:cadena;
    refer:integer;
    escon:boolean;
end;
(*end record*)
arreglo=packed array[1..limite] of concep;

(*****

var
    rules;basecon;
    questions:archivo;
    objetos:archiatr;
    nreglas,natributos,asc,asm:integer;
    atributos:arreglo;
    reglas:bc;
    causalidad:archi;
    contex:stkcon;
    metas:stkmeta;
    objetivos:pointer;

(*****
procedure leeatri(var numattr:integer;var attr:arreglo);
var
    aux:conceptos;
begin
    numattr:=1;
    read(objetos,aux);
    attr[1].nom:=aux.nombre;
    attr[1].refer:=1;
    attr[1].escon:=false;
    while not eof(objetos) do
        begin
            numattr:=numattr+1;
            read(objetos,aux);
            attr[numattr].nom:=aux.nombre;

```

```

        attr[numattr].refer:=numattr;
        attr[numattr].escon:=false;
    end;
    (*end while*)
end;
(*end leeatri*)

(*****)
procedure leeregias(var attr:arreglo;var ntotreg:integer;
                    var reglas:bc);
var
    efecto:registro;

    (*****)
    procedure generareglas(var reglas:bc;ntotreg:integer;
                           var attr:arreglo);
    var
        aux:pointer;
        causa:cond;
        name:string[8];

        (*****)
        procedure llenapu(var aux:pointer);
        begin
            aux^.atri:=causa.atr;
            aux^.value:=causa.val;
            aux^.liga:=nil;
        end;
        (*end llenapu*)

        (** cuerpo de generareglas **)
    begin
        with reglas[ntotreg] do
            begin
                valida := true;
                new(conse);
                conse^.atri:=efecto.numconcl;
                conse^.value:=efecto.valconcl;
                conse^.liga:=nil;
                attr[conse^.atri].escon:=true;
                name:=efecto.condiciones;
                assign(causalidad,name);
                reset(causalidad);
                read(causalidad,causa);
                new(antece);
                aux:=antece;
                llenapu(aux);
                while not eof(causalidad) do
                    begin
                        read(causalidad,causa);
                        new(aux^.liga);
                    end;
                end;
            end;
        end;
    end;
end;

```

```

        aux:=aux^.liga;
        llenapu(aux);
    end;
    (*end while*)
    close(causalidad);

    end;
    (*end with*)
end;
(*end generareglas*)

(**** cuerpo leereglas ****)
begin
    ntotreg:=1;
    read(rules,efecto);
    generareglas(reglas,ntotreg,attr);
    while not eof(rules) do
        begin
            read(rules,efecto);
            ntotreg:=ntotreg+1;
            generareglas(reglas,ntotreg,attr);
        end;
    (*end while*)
end;
(*end leereglas*)

(*****
procedure sort(izq,der:integer;var a:arreglo);
var
    aux1,aux2:concep;
    i,j:integer;
begin
    i:=izq;
    j:=der;
    aux1:=a[(izq+der) div 2];
    repeat
        while a[i].nom < aux1.nom do
            i:=i+1;
        while aux1.nom < a[j].nom do
            j:=j-1;
        if i <= j then
            begin
                aux2:=a[i];
                a[i]:=a[j];
                a[j]:=aux2;
                i:=i+1;
                j:=j-1;
            end;
    until i > j;
    if izq < j then
        sort (izq,j,a);

```

```

    if i < der then
        sort (i,der,a);
    end;
    (*end sort*)

```

```

    (*****

```

```

function hayconsulta : boolean;
var
    op:char;

```

```

    (*****

```

```

procedure lee(c,r:integer;var op:char);
begin
    repeat
        gotoxy(c,r);
        read(kbd,op);
    until (op='s') or (op='S') or (op='N') or (op='n');
end;
(*end lee*)

```

```

(* cuerpo de hayconsulta *)

```

```

begin
    clrscr;
    hayconsulta:=false;
    gotoxy(1,1);
    writeln('Quieres consultar al sistema (S/N) ?');
    lee(40,1,op);
    if (op='s') or (op='S') then
        hayconsulta:=true;
    end;
    (*end hayconsulta*)

```

```

    (*****

```

```

procedure leemeta(nreglas,natributos:integer;atributos:arreglo;
    var objetivo:pointer;reglas:bc);

```

```

var
    n,i : integer;

```

```

    (*****

```

```

procedure escribeylee(n:integer;a:arreglo;var numero:integer);
var

```

```

    i:integer;
    encontrado:boolean;
    concepto:cadena;

```

```

    (*****

```

```

procedure raya(n:integer);

```

```

var
    i:integer;
begin

```

```

    gotoxy(1,n);
    for i:=1 to 80 do
        write('*');
    (*end for*)
end;
(*end raya*)

(*****)
procedure imprime(var i,lim:integer;a:arreglo);
var
    r:integer;
    obj:conceptos;
begin
    gotoxy(1,1);
    write('Lista de conceptos :');
    r:=2;
    while (i<lim) and (r<17) do
        begin
            i:=i+1;
            if a[i].escon then
                begin
                    seek(objetos,a[i].refer-1);
                    read(objetos,obj);
                    r:=r+1;
                    gotoxy(5,r+2);
                    write(a[i].nom:15,' ... ',obj.interpret);
                end;
            (*end if*)
        end;
    (*end while*)
end;
(*end imprime*)

(*****)
procedure busca(nombre:cadena;n:integer;a:arreglo;var esta:boolean;
    var i:integer);
var
    j,k:integer;
begin
    esta:=false;
    k:=1;
    j:=n;
    repeat
        i:=(k+j) div 2;
        if nombre > a[i].nom then
            k:=i+1
        else
            j:=i-1;
    until (( a[i].nom=nombre )and a[i].escon) or (k>j);
    if (a[i].nom=nombre) and (a[i].escon) then
        esta:=true
    else

```

```

        i:=0;
    (*end if*)
end;
(*end busca*)

```

```

(*** cuerpo de escribeylee ***)

```

```

begin
    clrscr;
    raya(2);
    raya(20);
    i:=0;
    encontrado:=false;
    repeat
        imprime(i,natributos,atributos);
        repeat
            gotoxy(1,22);
            write('Dime el concepto que deseas conocer o ense#arme : ');
            if i<natributos then
                write(' o RETURN para el siguiente menu ');
            (*end if*)
            readln(concepto);
            if length(concepto) > 0 then
                busca(concepto,natributos,atributos,encontrado,numero);
            (*end if*)
            until (encontrado) or (length(concepto)=0) ;
            until (i >= natributos);
            clrscr;
        end;
    (*end escribe*)
end;

```

```

(*** cuerpo de leemeta ***)

```

```

begin
    escribeylee(natributos,atributos,n);
    if n<>0 then
        begin
            i:=nreglas + 1;
            repeat
                i:=i-1;
                until ( reglas[i].conse^.atri=atributos[n].refer);
                objetivo :=reglas[i].conse;
            end;
        (*end if*)
    end;
    (*end leemeta*)
end;

```

```

(*****)

```

```

procedure pushconex (numatri:integer;valatri:cadena;var asc:integer;
                    var contex:stkcon;deduc:boolean);

```

```

begin
    asc:=asc+1;
    with contex[asc] do

```

```

begin
  atr:=numatri;
  val:=valatri;
  inf:=deduc;
end;
(*end with*)
end;
(*end pushcontext*)

(*****)
procedure preguntar (apu:pointer;var asm,asc:integer;
  var contex:stkcon);
var
  frase:conceptos;
  respuesta:cadena;
  quest:palabras;
  op:char;
begin
  op:='N';
  with apu^ do
    begin
      reset(objetos);
      seek(objetos,atri-1);
      read(objetos,frase);
      writeln;
      seek(questions,atri-1);
      read(questions,quest);
      write(quest,' ',frase.interpret,' ? ');
      readln(respuesta);
      if length(respuesta) = 0 then
        respuesta:='indeterminado';
      (*end if*)
      pushcontext (atri,respuesta,asc,contex,false);
    end;
  (*end with*)
  asm:=asm-1;
end;
(*end preguntar*)

(*****)
procedure inferencia(natributos:integer;atributos:arreglo;
  var totreglas:integer;var reglas:bc;
  var asc,asm:integer;var contex:stkcon;
  var metas:stkmeta;objetivo:pointer);
var
  fin:boolean;
  nregla:integer;
  apu:pointer;

(*****)
procedure pushmeta(objetivo:pointer;var asm:integer;

```

```

                                var metas:stkmeta);
begin
  asm:=asm+1;
  metas[asm]:=objetivo;
end;
(*end pushmeta*)

(*****
function encontrar(lim,asm:integer;metas:stkmeta;reglas:bc;
                   var nregla:integer) : boolean;
begin
  encontrar:=false;
  nregla:=lim+1;
  repeat;
    nregla:=nregla-1;
    write(' ');
  until ( ( nregla=1 ) or
          ( reglas[nregla].conse^.atri=metas[asm]^.atri )
          and ( reglas[nregla].valida ) );

  if ( reglas[nregla].conse^.atri=metas[asm]^.atri ) and
     ( reglas[nregla].valida ) then
    encontrar := true;
  (*endif*)
end;
(*end encontrar*)

(*****
function valor(var apu:pointer;context:stkcon;reglas:bc;
              asc,nregla:integer) : evaluation;
var
  i:integer;
  aux:pointer;
  decidi :boolean;

begin
  aux :=reglas[nregla].antece;
  if asc<>0 then
    begin
      decidi:=false;
      while (aux<>nil) and (not decidi) do
        begin
          i:=0;
          repeat
            i:=i+1;
          until (context[i].atr=aux^.atri) or (i=asc);
          apu := aux;
          if context[i].atr = aux^.atri then
            if context[i].val=aux^.value then
              aux:=aux^.liga
        end;
      end;
    end;
end;

```

```

        else
            begin
                valor :=falsa;
                decidi := true;
            end
        (*end if*)
    else
        begin
            valor := indeterminada;
            decidi := true;
        end;
    (*end if*)
end;
(*end while*)
if not decidi then
    valor := cierta;
(*end if*)
end
else
    begin
        valor := indeterminada;
        apu := aux;
    end;
(*end if*)
end;
(*end valor*)

```

```

(*****)
function nohaymetas(asm:integer) ; boolean;
begin
    nohaymetas:=false;
    if asm=0 then
        nohaymetas:=true;
    (*end if*)
end;
(*end nohaymetas*)

```

```

(*****)
procedure descarte(nregla:integer;var reglas:bc);
begin
    reglas[nregla].valida:=false;
end;
(*end descarte*)

```

```

(*****)
procedure justifica(tope,asm:integer;context:stkcon;metas:stkmeta);
var
    obj1,obj2:conceptos;
    quest:palabras;

```

```

begin
  with contex[tope] do
    begin
      if val <> 'indeterminado' then
        begin
          writeln;
          write('Por ahora sabemos que ');
          if (val='si') or (val='SI') or
            (val='no') or (val='NO') then
            begin
              seek(objetos,atr-1);
              read(objetos,obj1);
              write(val);
              writeln(' ',obj1.interpret);
            end
          else
            begin
              seek(objetos,atr-1);
              read(objetos,obj1);
              write(obj1.interpret);
              writeln(' es ',val);
            end;
          (*end if*)
        end;
      (*end if*)
    end;
  (*end with*)
  if asm > 1 then
    begin
      seek(questions,metas[asm]^ .atri-1);
      read(questions,quest);
      seek(objetos,metas[asm]^ .atri-1);
      read(objetos,obj1);
      seek(objetos,metas[asm-1]^ .atri-1);
      read(objetos,obj2);
      writeln('si me dices ',quest,' ',obj1.interpret);
      if asm > 2 then
        writeln('podremos saber ',obj2.interpret)
      else
        writeln('conoceremos ',obj2.interpret);
      (*end if*)
    end;
  end;
(*end;
(*end justifica*)

(**** cuerpo de inferencia ****)
begin
  asm:=0;
  asc:=0;
  fin:=false;
  pushmeta(objetivo,asm,metas);

```

```

while not fin do
  if encontrar (totreglas,asm,metas,reglas,nregla) then
    case valor (apu,context,reglas,asc,nregla) of
      cierta: begin
        pushcontext (reglas[nregla].conse^.atri,
                    reglas[nregla].conse^.value,
                    asc,context,true);
        asm := asm-1;
        if nohaymetas(asm) then
          fin:=true;
          (*endif*)
        end;
      falsa: descarte(nregla,reglas);
      indeterminada: pushmeta(apu,asm,metas);
    end
    (*end case*)
  else
    begin
      writeln;
      if asm > 1 then
        begin
          if asc > 0 then
            justifica(asc,asm,context,metas);
            (*end if*)
          preguntar (metas[asm],asm,asc,context);
        end
      else
        fin:=true;
        (*end if*)
      end;
    end
    (*endif*)
  end
  (*end while*)
end;
(*end inferencia*)

(*****)
function haymeta(asc:integer;context:stkcon) : boolean;
var
  i:integer;
begin
  haymeta:=false;
  i := asc;
  while ( i>=1 ) and ( context[i].atr <> metas[i]^.atri ) do
    i := i-1;
  (*end while*)
  if context[i].atr = metas[i]^.atri then
    haymeta :=true;
  (*end if*)
end;
(*end haymeta*)

```

```

(*****)
procedure explica(tope:integer;context:stkcon);
var
  infersis:boolean;
  obj:conceptos;
begin
  clrscr;
  infersis:=true;
  writeln('Parece ser que ....');
  while tope <> 0 do
    begin
      if (context[tope].val='si') or
        (context[tope].val='SI') or
        (context[tope].val='no') or
        (context[tope].val='NO') then
        begin
          write(context[tope].val);
          seek(objetos,context[tope].atr-1);
          read(objetos,obj);
          write(' ',obj.interpret);
        end
      else
        begin
          seek(objetos,context[tope].atr-1);
          read(objetos,obj);
          write(obj.interpret);
          write(' es ',context[tope].val);
        end;
      (*end if*)
      tope := tope-1;
      if tope <> 0 then
        if not(context[tope].inf) then
          begin
            if not infersis then
              writeln(' y que')
            else
              writeln(', porque me has dicho que');
            (*end if*)
            infersis := false;
          end
        else
          begin
            writeln(', ademas he inferido que');
            infersis := true;
          end;
        (*end if*);
      (*end if*)
    end;
  (*end while*)
  readln;

```

```

end;
(*end explica*)

(*****)
procedure aprende(var reglas:bc;var totreglas,asm,asc:integer;
                 var contex:stkcon;metas:stkmeta);
var
  i,asmante:integer;
  aux:pointer;
  efecto:registro;
  causa:cond;
  nombre:string[8];
  cade:string[4];

(*****)
procedure creantece(var aux:pointer;var causa:cond;
                  contex:stkcon;i:integer);
begin
  aux^.atri:=contex[i].atr;
  aux^.value:=contex[i].val;
  aux^.liga:=nil;
  causa.atri:=aux^.atri;
  causa.val:=aux^.value;
  write(causalidad,causa);
end;
(*end creantece*)

(*****)
procedure otrositems(metas:stkmeta;asm:integer;var contex:stkcon;
                   var asc:integer);
var
  obj:conceptos;
  op:char;
  apu:pointer;
  auxiliar:condic;
  indicador:integer;
begin
  indicador := asm+1;
  apu:=metas[indicador]^liga;
  auxiliar:=contex[asc];
  asc:=asc-1;
  reset(objetos);
  if apu <> nil then
    begin
      writeln;
      write('Dime, para este caso, (s/n) es necesario ');
      writeln('que yo conozca');
    end;
  (*end if*)
  while apu <> nil do
    begin
      seek(objetos,apu^.atri-1);

```

```

read(objetos,obj);
write(obj.interpret,' ? ');
repeat
  read(kbd,op);
until (op='s') or (op='S') or (op='n') or (op='N');
writeln;
if (op='s') or (op='S') then
  preguntar(apu,indicador,asc,context);
(*end if*)
apu:=apu^.liga;
if apu<>nil then
  writeln(' tambien debo conocer ');
(*end if*)
end;
(*end while*)
asc:=asc+1;
context[asc] := auxiliar;
end;
(*end otrositems*)

```

(\*\*\*\*\* cuerpo de aprende \*\*\*\*\*)

```

begin
  writeln('No puedo llegar a ninguna conclusion, lo siento. ');
  writeln('Necesito aprender mas y quisiera que me dijeras ');
  asmante:=asm;
  asm:=i;
  preguntar(metas[asm],asm,asc,context);
  otrositems(metas,asmante,context,asc);
  i:=asc+1;
  repeat
    i:=i-1;
  until ( i=1 ) or (context[i].inf);
  totreglas:=totreglas + i;
  with reglas[totreglas] do
    begin
      valida := true;
      new(conse);
      conse^.atri:=context[asc].atr;
      conse^.value:=context[asc].val;
      conse^.liga:=nil;
      str (totreglas,cade);
      nombre:=concat('b:ante',cade);
      assign (causalidad,nombre);
      rewrite(causalidad);
      new(antece);
      aux:=antece;
      createce(aux,causa,context,i);
      aux^.liga:=nil;
      while i<asc-1 do
        begin
          i:=i+1;

```

```

        new(aux^.liga);
        aux:=aux^.liga;
        creantece(aux,causa,contex,i);
    end;
    (*end while*)
    close(causalidad);
    seek(rules,filesize(rules));
    efecto.numconc:=conse^.atri;
    efecto.valconc:=conse^.value;
    efecto.condiciones:=nombre;
    write(rules,efecto);
end;
(*end with*)
end;
(*end aprende*)

(*****
procedure renueva(totreglas:integer;var reglas:bc);
var
    i:integer;
begin
    for i:=1 to totreglas do
        if not reglas[i].valida then
            reglas[i].valida:=true;
        (*endif*)
    end;
(*end renueva*)

(*****
procedure presenta(totreglas:integer;reglas:bc);
var
    i:integer;
    aux:pointer;

(*****
procedure pon(apu:pointer);
var
    obj:conceptos;
begin
    read(objetos,obj);
    with apu^ do
        begin
            if (value='si') or (value='SI') or
                (value='no') or (value='NO') then
                writeln(value,' ',obj.interpret)
            else
                writeln(obj.interpret,' es ',value);
            (*end if*)
        end;
    (*end with*)

```

```

end;
(*end pon*)

(* cuerpo de presenta *)
begin
  clrscr;
  for i:=1 to totreglas do
    begin
      writeln('***** regla # ',i:3,'*****');
      write('SI ');
      seek(objetos,reglas[i].antece^.atri-1);
      aux:=reglas[i].antece;
      pon(aux);
      aux:=reglas[i].antece^.liga;
      while aux<>nil do
        begin
          write(' y ');
          seek(objetos,aux^.atri-1);
          pon(aux);
          aux:=aux^.liga;
        end;
      (*end while*)
      writeln;
      write('ENTONCES ');
      seek(objetos,reglas[i].conse^.atri-1);
      aux:=reglas[i].conse;
      pon(aux);
      readln;
    end;
  (*end with*)
end;
(*end presenta*)

```

```

(*****
***** programa principal *****)
begin
  assign(rules,'b;base.dat');
  reset(rules);
  assign(objetos,'b;atributo.dat');
  assign(questions,'b;pregunta.dat');
  reset(objetos);
  reset(questions);
  leeatri(natributos,atributos);
  leereglas(atributos,nreglas,reglas);
  sort(1,natributos,atributos);
  while hayconsulta do
    begin
      leemeta(nreglas,natributos,atributos,objetivo,reglas);
      inferencia(natributos,atributos,nreglas,reglas,asc,asm,

```

```

                                contex,metas,objetivo);
if haymeta(asc,contex) then
    explica(asc,contex)
else
    aprende(reglas,nreglas,asm,asc,contex,metas);
    (*endif*)
    renueva(nreglas,reglas);
end;
(*end while*)
readln;
presenta(nreglas,reglas);
close(rules);
close(objetos);
close(questions);
end.

```

**BIBLIOGRAFIA.**  
=====

**a. - MODELOS DE APRENDIZAJE.**  
-----

Anderson J.  
LANGUAGE, MEMORY AND THOUGHT.  
Hillsdale, N.J. Lawrence Erlbaum associates, 1983

Anderson J.  
"A general learning theory and its application to the acquisition  
of proof skills in geometry".  
MACHINE LEARNING: AN ARTIFICIAL INTELLIGENCE APPROACH.  
Tioga Press, 1983

Anderson J y Bower G.H.  
HUMAN ASSOCIATIVE MEMORY.  
Winston-Wiley, Washington D.C, 1973.

Badre N.A.  
"CLET: A computer program that learns arithmetic from an elementary  
textbook".  
IBM RESEARCH Rept. RC 4235, 1973

Bobrow D.G y Collins A.  
REPRESENTATION AND UNDERSTANDING: STUDIES IN COGNITIVE SCIENCE.  
Academic Press, NY, 1975

Buchanan B.G.  
"Model directed learning of production rules".  
PATTERN-DIRECTED INFERENCE SYSTEMS  
Academic Press, 1978, pages:297-312

Feigenbaum E.A. y Feldman J.  
COMPUTERS AND THOUGHT  
McGraw-Hill, 1963.

Polya G.  
MATHEMATICS AND PLAUSIBLE REASONING, Vol.2  
Jhon Wiley and Sons, NY, 1954.

Schank R.C y Colby K.M.  
COMPUTER MODELS OF THOUGHT AND LANGUAGE.  
W.H. Freeman, San Francisco, 1973

b.- TOPICOS EN INTELIGENCIA ARTIFICIAL.

---

Barr A. y Feigenbaum E.A.  
THE HANDBOOK OF ARTIFICIAL INTELLIGENCE.  
Los Altos, Cal. 1981 (varios volúmenes)

Dreyfus H.  
WHAT COMPUTERS CAN'T DO: A CRITIQUE OF ARTIFICIAL REASON.  
Harper & Row, N.Y., 1972.

Erman L.D y Lesser V.R.  
"System engineering techniques for artificial intelligence systems".  
COMPUTER VISION SYSTEMS, Academic Press, 1978, Pags:37-45

Goldstein I y Papert S.  
"Artificial intelligence, language and the study of knowledge".  
COGNITIVE SCIENCE, Vol.1, No.1, 1977, pags:84-123

James M.  
ARTIFICIAL INTELLIGENCE IN BASIC.  
Newnes Technical Books, 1984.

Kowalsky R.  
"AI and software engineering".  
DATAMATION, noviembre 1984.

Lenat D.B.  
"The nature of heuristics".  
ARTIFICIAL INTELLIGENCE. No.19, 1982, pags:189-249

McCorduck P.  
MACHINES WHO THINK: A PERSONAL INQUIRY INTO THE HISTORY AND PROSPECTS OF ARTIFICIAL INTELLIGENCE.  
W.H. Freeman, San Francisco, 1979.

Minsky M.  
SEMANTIC INFORMATION PROCESSING.  
MIT Press, Cambridge, Mass., 1968.

Stallman R y Sussman G.J.  
"Forward reasoning and dependency-directed backtracking in a system for a computed-aided circuit analysis".  
ARTIFICIAL INTELLIGENCE, No.9, 1977, pags:135-196

Winograd T.  
"Extended inference modes in reasoning by computer systems".  
ARTIFICIAL INTELLIGENCE, No.13, 1980, pags:5-26

Winston P.  
THE PSYCHOLOGY OF COMPUTER VISION.  
McGraw-Hill, 1975.

Yasaki E.K.  
"AI: More than science".  
DATAMATION, septiembre 1981, pags:63-64

c.- EDUCACION ASISTIDA POR COMPUTADORA EN BASE A INTELIGENCIA  
-----  
ARTIFICIAL.  
-----

Blaine L. y Smith R.L.  
"Intelligent CAI: The role of the curriculum in suggesting  
computational models of reasoning".  
Proceedings of the 1977 ACM annual conference, octubre 1977,  
en Seattle, pags:241-246

Brown, J.S. y Burton R.R.  
"Multiple representations of knowledge for tutorial reasoning".  
REPRESENTATION AND UNDERSTANDING: STUDIES IN COGNITIVE SCIENCE.  
Academic Press, 1975, pags:311-349

Brown J.S y Burton R.R.  
"Diagnosing models for procedural bugs in basic mathematical  
skills".  
COGNITIVE SCIENCE, Vol.2, 1978, pags:155-192

Brown J.S, Burton R.R y Bell A.  
"SOPHIE: a sophisticated instructional environment for teaching  
electronic troubleshooting (an example of AI in CAI)".  
International Journal in Man-machine studies, Vol.7, 1977.

Carbonell, J.R.  
"AI in CAI: An artificial intelligence approach to computer  
assisted instruction".  
IEEE Transactions on Man-machine Systems, 1970, MMS-11190-202.

Clancey W.J, Shortliffe y Buchanan B.G.  
"Intelligent computer-aided instruction for medical diagnosis".  
PROCEEDINGS of the third annual symposium of computer applica-  
tions in medical care., 1979, pags: 175-183

Clancy, William, Bennett, James y Cohen.  
"Applications-oriented AI research: education".  
HANDBOOK OF ARTIFICIAL INTELLIGENCE, (Barr, Avron y Feigenbaum,  
editores), 1979.

Gable A. y Page C.V.  
"The use of AI techniques in computer-assisted instruction: an  
overview".  
International Journal in Man-machine studies, 1980, No.12,  
pags:259-282

Minamikawa T, Oyanagi S y Takei K.  
"Expert systems for education and other domains".  
J. Inst. Electron. and Commun. JPN.

Mora J.L. y Negrete J.  
"Can a learning expertise system become a tutor?"  
Proceedings of the 6th European Conf. in A.I., septiembre 1984.

Sleeman D. y Brown J.S.  
INTELLIGENT TUTORING SYSTEMS.  
Academic Press, Cambridge, Mass., 1982

varios autores  
THE ICAEW ANNUAL CONFERENCE - THE TECHNOLOGY REVOLUTION.  
ACCOUNTANT, Gran Breta#a, No.12, pags:7-9

#### d.- SISTEMAS EXPERTOS.

Buchanan, B.G.  
"New research on expert systems".  
MACHINE INTELLIGENCE, Edinburg University Press, Vol.10,  
1982, pags: 269-299

Buchanan B.G y Feingenbaum E.A.  
"DENDRAL and METADENDRAL: Their applications dimension".  
ARTIFICIAL INTELLIGENCE, No. 11, 1978, pags:5-24

Davies R.  
"Expert systems: Where are we? and where do we go from here?"  
AI MAGAZINE 3, 1982, No.2, pags:3-22

Davies R.  
"Interactive transfer of expertise: acquisition of new inference  
rules".  
Proceedings of the international joint conference on  
artificial intelligence 5, 1977, pags:409-415.

Duda R.O y Gaschnig J.G

"Knowledge-based expert systems coming of age".  
BYTE Vol.6, No.9, pags:238-281

Edelson E.

"Expert systems - computers that think like people".  
POPULAR SCIENCE, septiembre 1982, pags:48-52

Erman L.D, London P.E y Fickas S.F.

"The design and an example use of HERSAY-III".  
Proceedings of the international joint conference on  
artificial intelligence 7, 1979, pags:409-415.

Gaschnig J.

"PROSPECTOR. An expert system for mineral exploration".  
MACHINE INTELLIGENCE, Infotech state of the art report 9, No.3,  
1981.

Gorry G.A.

"Research on expert systems".  
MAC technical memorandum 56, project MAC, MIT, 1974.

Hayes-Roth F.

"Building expert systems: putting knowledge to work".  
Proceedings of the international joint conference on  
artificial intelligence 7, 1979, invited tutorial.

Larkin J, McDermott J, Simon D.P y Simon H.A.

"Expert and novice performance in solving physics problems".  
SCIENCE, No. 20, 1980, pags:208

Martins G.R.

"The overselling of expert systems".  
DATAMATION, noviembre 1984.

McDermott J.

"R1: An expert in the computer systems domain".  
Proceedings, first annual Nat. Conf. Artificial Intelligence,  
1980, pags:269-271

Michaelson R.

"An expert system for federal tax planning".  
EXPERT SYSTEMS: THE INTERNATIONAL JOURNAL OF KNOWLEDGE ENGINEERING,  
Octubre 1984, pag:149

Michaelson R. y Michie D.

"Expert systems in bussines".  
DATAMATION, noviembre 1983.

Michie D.  
EXPERT SYSTEMS IN THE MICRO-ELECTRONIC AGE  
Edinburg University Press, 1979.

Michie D.  
INTRODUCTORY READINGS IN EXPERT SYSTEMS.  
Gordon & Breach, London and NY, 1982

Naylor C.  
BUILD YOUR OWN EXPERT SYSTEM.  
Sigma Technical Press, 1983

Shortliffe E.H.  
COMPUTER-BASED MEDICAL CONSULTATION: MYCIN.  
American Elsevier, N.Y, 1976

Stefik M, Aikins J, Balzer R, Benoit J, Birnbaum L, Hayes-Roth F y  
Sacerdoti E.  
"The organization of expert systems: A tutorial".  
ARTIFICIAL INTELLIGENCE, No.18, 1982, pags:135-173

Waterman D.A.  
"Rule-based expert systems".  
MACHINE INTELLIGENCE, Infotech state of the art report 9, No.3,  
1981.

#### e.- LENGUAJE NATURAL.

-----

Burton R.R.  
"Semantic grammar: an engineering technique for constructing  
natural language understanding systems".  
BBN Rept 3453, Cambridge, Mass, 1976.

Hendrix G.G y varios.  
"Developing a natural language interface to complex data".  
ACM Trans. Database systems, Vol. 3, 1978, pags:105-147

Sager N.  
NATURAL LANGUAGE INFORMATION PROCESSING.  
Addison-Wesley, reading, Mass., 1981.

Shapiro S.C. y Kwasny S.C.  
"Interactive consulting via natural language".  
COMM. ACM, Vol.18, No.8, agosto 1975.

Shneiderman B.  
SOFTWARE PSYCHOLOGY: HUMAN FACTORS IN COMPUTER AND INFORMATION  
SYSTEMS.  
Winthrop, Cambridge, Mass, 1980.

Thomas J y Carrol J.  
"Human factors in communication".  
IBM SYSTEMS JOURNAL, No.20, pags:237-263

Weinzenbaum J.  
"ELIZA: A computer program for the study of natural language  
communication between man and machine".  
COMM. ACM, Vol.9, No.1, enero 1966, pags:36-44.

Wilensky R.  
"Meta-planning: representing and using knowledge about planning  
in problem solving and natural language understanding".  
COGNITIVE SCIENCE, No.5, 1981, pags:197-233

Winograd T.  
UNDERSTANDING NATURAL LANGUAGE.  
Academic Press, NY, 1972

f.- REPRESENTACION DE CONOCIMIENTO.  
-----

Davies R.  
"Knowledge acquisition in rule-based systems: knowledge about  
representation as a basis for system construction and maintenance".  
PATTERN-DIRECTED INFERENCE SYSTEMS. Academic Press, 1978,  
pags:99-134

Davies R. y Buchanan B.G.  
"Meta level knowledge: overview and applications".  
Proceedings of the international joint conference on  
artificial intelligence 5, 1977, pags:920-928

Davies R. y Lenat D.  
KNOWLEDGE-BASED SYSTEMS IN ARTIFICIAL INTELLIGENCE  
McGraw-Hill, 1980

Feingenbaum E.A.  
"The art of artificial intelligence: Themes and case studies of  
knowledge engineering".  
Proceedings of the international joint conference on  
artificial intelligence 5, 1977, pags:1014-1029

Ferrater J y Hugues L.  
LOGICA MATEMATICA, (1937).  
Fondo de cultura economica, ed. de 1983.

Hayes-Roth F, Klahr P y Mostow D.J.  
"Knowledge acquisition, knowledge programming and knowledge  
refinement".  
Rand Paper R-2540-NSF, Rand Corp. 1980.

Kaufmann A.  
AN INTRODUCTION TO THE THEORY OF FUZZY SETS.  
Academic, NY, 1975, vol 1.

Shubert L.K.  
"Extending the expressive power of semantic nets".  
ARTIFICIAL INTELLIGENCE, Vol.7, No.2, 1976, pags:163-198

Zadeh L.A.  
"Fuzzy Logic and approximate reasoning".  
SYNTHESE, Vol.30, 1975.

Zadeh L.A.  
"Fuzzy sets".  
INFORMATION AND CONTROL, No.8, 1965, pags:338-353

g.- SOFTWARE PARA LA CONSTRUCCION DE SISTEMAS BASADOS EN  
-----  
CONOCIMIENTO.  
-----

Bobrow D.G. y Raphael B.  
"New programming languages for AI research".  
COMPUTER SURVEYS, Vol.6, No.3, septiembre 1974.

Bobrow, D.G. y Winograd T.  
"An overview of KRL, a knowledge representation language".  
COGNITIVE SCIENCE, 1977, No. 1, pags:3-46

Campbell J.A.  
IMPLEMENTATIONS OF PROLOG.  
Jhon Wiley and Sons, NY, 1984.

Clark K.L y McCabe F.G.  
MICRO-PROLOG: PROGRAMMING IN LOGIC.  
Prentice Hall, 1984.

McDermott D.

"The PROLOG phenomenon".

SIGART NEWSLETTER, No. 72, Julio 1980, pags:16-20

Hendrix G.G.

"LIFER: a natural language interface facility".

SIGART Newsletter, Vol.16, 1977

Murphy T.

"Artificial intelligence topics at IBM".

SIMULATION, enero 1985, pags: 33-37.

Nii H.P y Aiello N.

"AGE: A knowledge-base program for building knowledge-based programs".

Proceedings of the international joint conference on artificial intelligence 6, 1979, pags:645-655

Roussel P.

"PROLOG: manuel de reference et d' utilisation".

Aix-Marseille University, 1975.

Schmolze J.G y Brachman R.J.

"Summary of the KL-ONE language".

Proceedings 1981 KL-ONE workshop, FLAIR tech, report 4, Fairchild laboratory for artificial intelligence research, Palo Alto, Cal. 1982, pags:231-257

Sussman G.J, Winograd T y Charniak E.

"MICROPLANNER Reference manual".

AI memo 203, Cambridge,Mass.,MIT artificial intelligence laboratory, 1970.

Ueno H.

"An end-user oriented language to develop knowledge-based expert systems".

Proceedings COMPCON 83 Fall : Delivering computer power to end users. 27th IEEE computer soc. inter. conf., Arlington, USA. IEEE Comp. Soc. Press., septiembre 1983, pags: 25-29

Van Emden y Kowalsky R.A.

"The semantics of predicate logic as a programming language".

J. ACM, Vol.23, No.4, 1976.

Van Melle W, Shortliffe E.H. y Buchanan B.G.

"EMYCIN, A domain-independent system that aids in constructing knowledge-based consultation programs".

MACHINE INTELLIGENCE, Infotech state of the art report 9, No.3, 1981.

Weiss S.M y Kulikowsky C.A.  
"EXPERT: A system for developing consultation models".  
Proceedings of the international joint conference on  
artificial intelligence 6, 1979, pags:942-947

Winston P y Berthold K.P.  
LISP, segunda edicion.  
Addison-Wesley, 1984.

h.- EDUCACION ASISTIDA POR COMPUTADORA.  
-----

Bengelsdorf I.S.  
"University's role is education, not programming".  
COMPUTERWORLD, Vol.17, No. 33, 1983, pags:11-19

Bloom B.S.  
"The 2 Sigma problem: the search for methods of group instruction  
as effective as one-to-one tutoring".  
EDUCATIONAL RESEARCHER, No.13, 1984, pag:3

Couffignal L.  
"La pedagogie cybernetique".  
L'EDUCATION NATIONALE, No.15-16, abril 1965.

Coulson J.E.  
"Utilisation des ordinateurs pour la recherche et le developpement  
de l'instruction programme".  
LE COURRIER DE LA RECHERCHE PEDAGOGIQUE, enero 1965.

Hirschbuhl K.  
"The need for computer literacy and computer applications in the  
nation's classrooms".  
J. EDUC. TECHNOL. SYST., USA, 1983, Vol. 19, No.3, pags:183-191

Landa L.N.  
CIBERNETICA Y PEDAGOGIA.  
Editorial Labor, Barcelona, 1972.

Mayer R.E.  
"Some conditions of meaningful learning for computer programming".  
Journal of Educational Psychology, 1982, Vol.68, pags:143-150

Novacki T.  
"Les postulats de recherches de la pedagogie cybernetique".  
Actes du VI congr. intern. des Sciences de l'Education, II, Paris,  
1976, pags:11-15

Rushby, N.J.  
COMPUTER-BASED LEARNING. STATE OF THE ART REPORT.  
Pergamon Infotech, Inglaterra, 1983

Taylor R.P.  
THE COMPUTER IN THE SCHOOL: TUTOR, TOOL, TUTEE.  
Columbia University, Teachers College Press, NY, 1980.

Watt D.  
LEARNING WITH LOGO  
Byte Books, McGraw-Hill, NY, 1983

varios autores.  
SIMPOSIUM INTERNACIONAL LA COMPUTACION EN LA EDUCACION INFANTIL.  
Memoria, 24-26 octubre 1984, Programa Universitario de Cómputo,  
UNAM, México.

#### 1.- PEDAGOGIA.

-----

Not L.  
LAS PEDAGOGIAS DEL CONOCIMIENTO, (1979).  
Fondo de cultura económica, ed. de 1983.

Piaget J.  
LA ENSEÑANZA DE LAS MATEMATICAS  
Ed. Aguilar, Madrid, 1963.

Piaget J.  
LA PSICOLOGIA DE LA INTELIGENCIA  
Ed. Psique, Buenos Aires, 1960

Piaget J.  
EL NACIMIENTO DE LA INTELIGENCIA EN EL NIÑO.  
Ed. Aguilar, Madrid, 1963.

Piaget J.  
LA FORMACION DEL SIMBOLO EN EL NIÑO.  
Fondo de cultura económica, Mexico, 1961

Piaget J. y Beth E.W.  
RELACIONES ENTRE LA LOGICA FORMAL Y EL PENSAMIENTO REAL.  
Ed. Ciencia nueva, Madrid, 1968

Piaget J. y Inhelder B.  
EL DESARROLLO DE LAS CANTIDADES EN EL NIÑO.  
Ed. Nova Terra, Barcelona, 1971

Piaget J, Wertheimer M, Hendle M, Woodworth R.S y otros.  
INVESTIGACIONES SOBRE LOGICA Y PSICOLOGIA.  
Alianza editorial, 1977.

Sastre G. y Moreno M.  
DESCUBRIMIENTO Y CONSTRUCCION DE CONOCIMIENTOS: UNA EXPERIENCIA  
DE LA PEDAGOGIA OPERATORIA.  
Ed. Gedisa, Barcelona, 1980

**REFERENCIAS.**

=====

Anderson J.R y Reiser B.J.

"The LISP Tutor".

BYTE, Vol.10, No.4, abril 1985, p: 159-175

Ayala G.

"Representacion de conocimiento".

Memorias de la Primera Conf. Intern. Las computadoras en instituciones de educación superior, México, Mex., marzo 20-23, 1985, Burroughs y PUC-UNAM.

Barstow, Aiello, Duda, Erman, Forgy, Gorlin, Greiner, Lenat, London, McDermott, Nii, Politakis, Reboh, Roseschain, Scott, Van Melle y Weiss.

"Languages and tools for knowledge engineering".

BUILDING EXPERT SYSTEMS, Addison-Wesley, 1983, p: 283-345

Brachman R.J.

"On the epistemological status of semantic networks".

ASSOCIATIVE NETWORKS; REPRESENTATION AND USE OF KNOWLEDGE BY COMPUTERS, Academic Press, 1979, p: 3-50

Brachman R.J.

"What IS-A is and Isn't".

COMPUTER, Vol. 16, No. 10, octubre 1983, p: 30-36

Brachman R.J, Amarel, Engelman, Engelmores, Feingenbaum y Wilkins.

"What are expert systems ?".

BUILDING EXPERT SYSTEMS, Addison-Wesley, 1983, p: 31-57

Brachman R.J, Fikes R.E y Levesque H.J.

"Krypton : A functional approach to knowledge representation".

COMPUTER, vol.16, No. 10, octubre 1983, p: 67-73

Bonner P.

"Computers in education : promise and reality.

PERSONAL COMPUTING, Vol. 8, No. 9, septiembre 1984, p : 64-77

Boyer E.L.

"Education's new challenge".

PERSONAL COMPUTING, Vol. 8, No. 9, septiembre 1984, p : 81-85

Buchanan B.G, Barstow D, Rechtal R, Bennett J, Clancey W, Kulikowski C, Mitchell T and Waterman D.A.

"Constructing an Expert System".

BUILDING EXPERT SYSTEMS, Addison-Wesley, 1983, p :127-167

Chorover S,L.

"Cautions on computers in education, Effects on the student-teacher relationship".

BYTE, Vol. 8, No. 6, junio 1984, p: 223-226

Dhal V.

"Logic Programming as a representation of knowledge".

COMPUTER, vol.16, No.10, octubre 1983, p: 106-111

Feingenbaum y Mc.Corduck

THE FIFTH GENERATION

Addison-Wesley, 1983, 275 pags.

Findler N.V.

"Artificial Intelligence".

ENCYCLOPEDIA OF COMPUTER SCIENCE.

Van Nostrand Reinhold, 1976.

Finzer W y Gould L.

"Programming by rehearsal, an environment for developing educational software".

BYTE, Vol. 8, No. 6, junio 1984, p: 187-210

Frenzel L.

"The personal computer, last chance for CAI ?".

BYTE, Vol. 7, No. 5, julio 1980, p: 86-96

Gil V.

"Sistema Plato".

Primer simposio internacional "Las computadoras en la educación infantil". Programa Universitario de Cómputo, UNAM, México, octubre 1984.

Guzmán A.A.

"Sistemas expertos y sus aplicaciones".

INSTITUTO POLITECNICO NACIONAL, 1984.

Hasset J.

"Computers in the classroom".

PSYCHOLOGY TODAY, Vol.18, No.9, septiembre 1984

Hayes-Roth F.

"The knowledge based expert system : a tutorial".

COMPUTER, septiembre 1984, p: 11-28

Hayes-Roth F, Waterman D.A y Lenat D.B.

"An overview of expert systems".

BUILDING EXPERT SYSTEMS, Addison-Wesley, 1983, p: 3-29

Hendrix G.G.

"Encoding knowledge into partitioned networks".

ASSOCIATIVE NETWORKS; REPRESENTATION AND USE OF KNOWLEDGE BY COMPUTERS, Academic Press, 1979, p: 51-92

Kinnucan P.

"Computers that think like experts".

HIGH TECHNOLOGY, enero 1984, p: 30-38

Krivine F.

"Classroom computing : an european perspective".

PERSONAL COMPUTING, Vol. 8, No. 9, septiembre 1984, p : 70

Kobsa A.

"Knowledge representation : a survey of its mechanisms, a sketch of its semantics".

CYBERNETICS AND SYSTEMS, No. 15, 1984, p: 41-89

Lenat D, Davis R, Doyle J, Genesereth M, Goldstein I y Schrobe H.

"Reasoning about reasoning".

BUILDING EXPERT SYSTEMS, Addison-Wesley, 1983, p: 3-29

Levesque H y Mylopuolos J.

"A procedural semantics for semantic networks".

ASSOCIATIVE NETWORKS; REPRESENTATION AND USE OF KNOWLEDGE BY COMPUTERS, Academic Press, 1979, p: 121-175

Michaelson R.H, Michie D and Boulanger A.

"The technology of Expert Systems".

BYTE, Vol. 10, No. 4, abril 1985, p: 303-312

Mylopuolos J, Shibahara T y Tsotsos J,K.

"Building Knowledge-based systems: the PSN experience".

COMPUTER, Vol. 16, No. 10, octubre 1983, p: 83-89

Nau D.S.

"Expert computer systems".

COMPUTER, Vol. 16, No. 2, febrero 1983, p: 63-85

O'Malley C.

"A buyer's guide to educational science software".

PERSONAL COMPUTING, Vol. 8, No. 9, septiembre 1984, p : 94-115

Papert S.

MINDSTORMS. CHILDREN, COMPUTERS AND POWERFUL IDEAS.

Basic Books, Nueva York, 1980.

Pérez C.C y Pérez T.D.  
"El transplante de software educacional y el diseño de productos nacionales".

Primer simposio internacional "Las computadoras en la educación infantil". Programa Universitario de Cómputo, UNAM, México, octubre 1984.

Piaget J.  
PSICOLOGIA Y PEDAGOGIA. (1969)  
Sarpe, edición de 1983.

Piaget J.  
SEIS ESTUDIOS DE PSICOLOGIA. (1971)  
Origen/Planeta, edición de 1985.

Piestrup A.  
"Game sets and builders, graphics-based learning software".  
BYTE, Vol. 8, No. 6, junio 1984, p: 215-219

Rich E.  
ARTIFICIAL INTELLIGENCE.  
McGraw-Hill, international student edition, 1983, 436 pags.

Rich E.  
"Natural-language interfaces".  
COMPUTER, septiembre 1984.

Roberts F.C y Park D.  
Intelligent computer-assisted instructions: an explanation and overview".  
EDUCATIONAL TECHNOLOGY, diciembre de 1983, p:7-12

Schubert L.K, Goebel R.C y Cercone N.J.  
"The structure and organization of a semantic net for comprehension and inference".  
ASSOCIATIVE NETWORKS; REPRESENTATION AND USE OF KNOWLEDGE BY COMPUTER, Academic Press, 1979, p: 121-175

Suzuky J.D.  
"Formación de recursos humanos en las ciencias de la computación, para la enseñanza a niveles básicos medios".  
CERO UNO CERO, Vol. 4, No. 5, junio 1984, p: 11-16

Thompson B.A and Thompson W.A.  
"Inside an Expert System".  
BYTE, Vol. 10, No. 4, abril 1985, p: 315-330

Weinzenbaum J.  
"Another view from MIT".  
BYTE, Vol. 8, No. 6, junio 1984, p: 225

Weiss S.M. and Kulikowski C.A.  
A PRACTICAL GUIDE TO DESIGNING EXPERT SYSTEMS.  
Rowman and allenheld publishers, 1984, 174 pags.

Winston P,H.  
ARTIFICIAL INTELLIGENCE, Addison-Wesley, 1977.

Woods W.A.  
"What's important about knowledge representation ?".  
COMPUTER, Vol. 16, No. 10, octubre 1983, p: 22-27

Woolf B y McDonald D.  
Building a computer tutor: design issues.  
COMPUTER, septiembre 1984, p:61-73

Wyer J.  
"New bird on the branch: Artificial intelligence and Computer-  
Assisted Instruction".  
INFORMATION SYSTEMS, Vol.3, No.21, Digital Equipment Corporation,  
Massachusetts, 1984, p:185-191