16
24

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON

"PROYECTO DE UN LABORATORIO DE DESARROLLO DE SISTEMAS EXPERTOS"

T E S I S

Que para obtener el título de

INGENIERO EN COMPUTACION

P r e s e n t a n

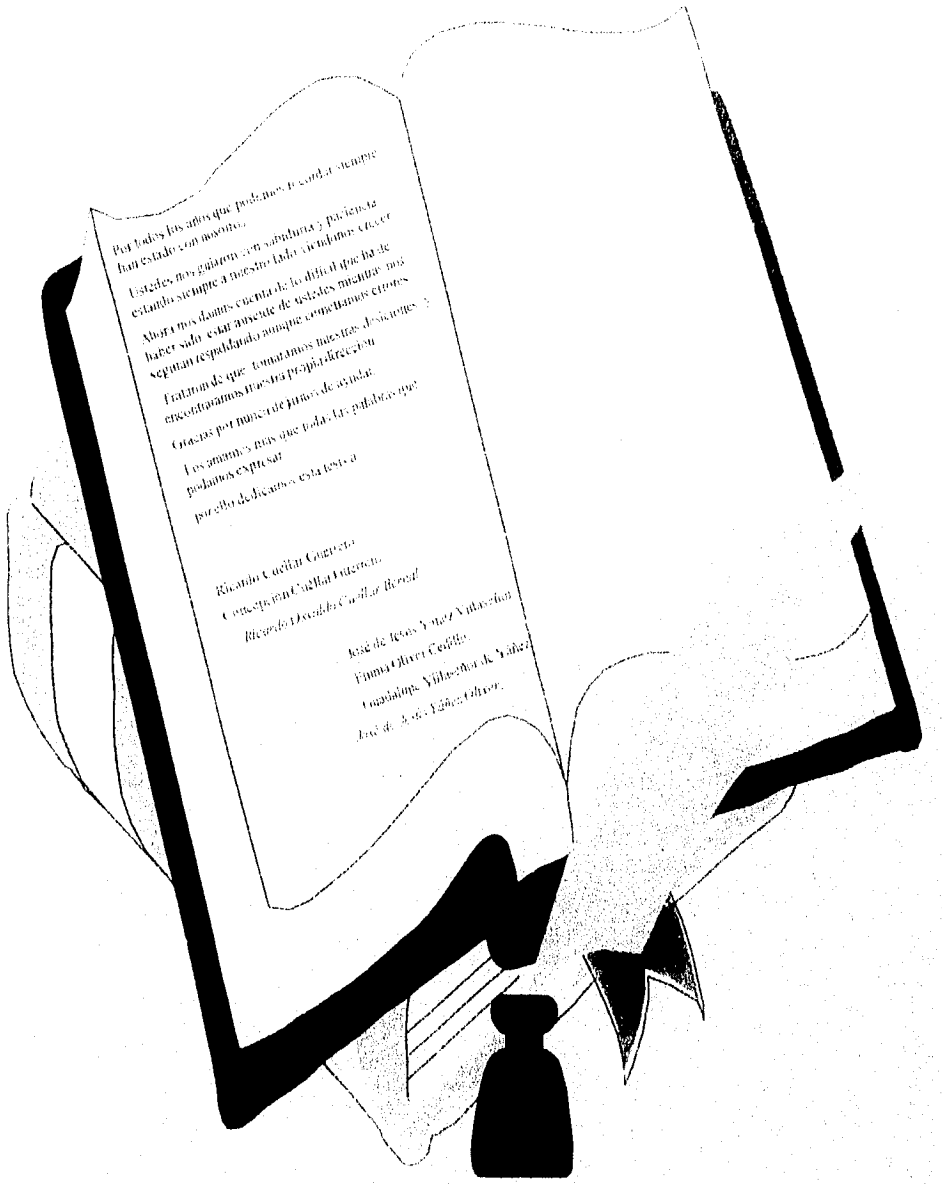
RICARDO OSVALDO CUELLAR BERNAL

JOSE DE JESUS YAÑEZ OLIVER

Asesor Ing. Amilcar Monterrosa Escobar

San Juan de Aragón Edo. de Mex. Agosto 1996.

CONTENIDO


Por todos los años que podamos contar siempre
han estado con nosotros
Ustedes nos guiaron con sabiduría y paciencia
estando siempre a nuestro lado cuidando crecer
Ahora nos damos cuenta de lo difícil que ha de
haber sido estar ausente de ustedes mientras nos
seguían respaldando aunque comentamos errores
Tratamos de que tomaran nuestras decisiones
encontráramos nuestra propia dirección
Gracias por nunca dejarnos de ayudar
Les amamos más que todas las palabras que
podamos expresar
por ello dedicamos esta tesis a
Ricardo Cuellar Cisneros
Concepción Cuellar Guerrero
Ricardo Osvaldo Cuellar Bernal
José de Jesús Yáñez Villaseñor
Emma Olivera Castillo
Guadalupe Villaseñor de Yáñez
José de Jesús Yáñez Olivera

ANTES QUE NADA QUEREMOS AGRADECER A DIOS, POR QUE SIN SU AYUDA NO HUBIERA SIDO POSIBLE REALIZAR ESTE SUEÑO .

Deseamos agradecer especialmente:

A mis hermanos porque con ellos he compartido la fortuna de vivir.

Agradezco a Diana por todos esos momentos que compartimos, por el impulso y apoyo que me brindo cuando lo necesite en momemos dificiles.

A mis compañeros de trabajo y jefes de la SCT por su apoyo en la realización de este trabajo asi como durante mis años en la universidad

JOSE DE JESUS YAÑEZ OLIVER

A Carlos y Verónica por el estímulo y Cariño que me han brindado.

RICARDO OSVALDO CUELLAR BERNAL.

A Nuestro Asesor Ing. Amilcar Monterrosa Escobar por habernos brindado su tiempo, conocimientos y apoyo para la realización de este trabajo.

A Nuestros grandes amigos de la universidad por serlo, Raúl, Rodolfo y Gladis (ZH).

Por nuestra amistad....

Tener un amigo es un don divino. La amistad leal, sincera, desinteresada es la verdadera comunión, la verdadera amistad perdura atraves del tiempo y la distancia...

JOSE DE JESUS YAÑEZ OLIVER RICARDO OSVALDO CUELLAR BERNAL

Somos una civilización científica - ha dicho Jacob Bronowski - . Eso significa una civilización en la que el saber y su integridad son factores cruciales. Ciencia no es mas que una palabra latina que significa conocimiento... Nuestro destino es el conocimiento.

Carl Sagan, Los dragones del Edén.

PROLOGO

" Llegará una época en la que una investigación diligente y prolongada sacará a la luz cosas que hoy están ocultas. La vida de una sola persona, aunque estuviera toda ella dedicada al ciclo, sería insuficiente para investigar una materia tan vasta. Por lo tanto este conocimiento sólo se podrá desarrollará lo larga de sucesivas edades. Llegará una época en la que nuestros descendientes se asombrarán de que ignoremos cosas que para ellos son tan claras... Muchos son los descubrimientos reservados para las épocas futuras, cuando se haya borrado el recuerdo de nosotros. Nuestro universo sería una cosa muy limitada sino ofreciera a cada época algo que investigar... La naturaleza no revela sus misterios de una vez para siempre".

Séneca, Cuestiones naturales libro ", siglo primero

El presente trabajo es una investigación que propone incorparar a la Universidad Nacional Autonoma de México en su Campus "ARAGON" a formar parte del desarrollo de la tecnología de Sistemas Expertos. Para ello primeramente se muestran los conceptos teóricos, sobre los que se sustenta el desarrollo de SE's, posteriormente se hace una presentación de la situación de los SE's en México y finalmente se propone la infraestructura y elementos auxiliares del Laboratorio de Desarrollo de Sistemas Expertos. A continuación, se hace una descripción de los capítulos que se encuentran en el presente trabajo.

En el primer capítulo se da una introducción a la Inteligencia Artificial y a los Sistemas Expertos, para que el lector comprenda a grandes rasgos las principales características de estas areas de estudio.

El capítulo 2 trata los conceptos teóricos en los que se fundamenta la tecnología de SE's, partiendo de su definición, la importancia del conocimiento, y a manera de antología, los esquemas de representación de conocimiento más comunes. Así mismo se incluye la descripción de la arquitectura de un SE y de los módulos que lo integran.

En el capítulo 3 se presenta un panorama de la situación mexicana en torno a los SE's. Al principio se describe brevemente, en forma de anecdotario, la historia y evolución dela IA en México. Posteriormente se muestra la problemática de integración de la tecnología al proceso productivo y las diversas alternativas que han sido presentadas por las empresas y los grupos de investigación. Dentro de éste panorama, se analizan los planes de estudio de algunas instituciones de enseñanza superior, interesadas en formar recursos humanos en esta tecnología. Finalmente se presentan los pronósticos de algunos expertos y profesionales del área, sobre lo que ocurrirá en nuestro país con los SE's.

En el capítulo 4 se hace un análisis de las herramientas de construcción de SE's, disponibles en el mercado. El capítulo se divide en herramientas de software y de hardware. Para las primeras se analizan los lenguajes de programación más utilizados en IA, así como las utilerías escritas a partir de ellos. Posteriormente se mencionan algunos criterios de selección de herramientas y se presenta un catálogo de las más comúnes en el mercado, describiendo sus atributos más significativos.

En el capítulo 5 se proponen las características que el Laboratorio de Desarrollo de SE's, del Depto de Ingeniería en Computación, de la Universidad Nacional Autonoma de México en su Campus "ARAGON" debe tener de acuerdo a los objetivos postulados. Posteriormente se hace una breve descripción de la metodología dediseño y los elementos que interviene, para finalizar con la elección de heramientas y actividades recomendadas.

En el capítulo 6 se dan las conclusiones finales de ltrabajo, mostrando cuales han sido las contibuciones, cuales las limitaciones y cuales podrían ser las posibles extensiones. Al final se encuentran los comentarios finales, donde se mencionan opiniones y perspectivas personales sobre el presente texto.

CAPITULO I
INTRODUCCION

1.1 I.A.

Dar un concepto de I.A. es muy difícil, puesto que varios estudiosos de la materia han expresado sus definiciones, las cuales no son aceptadas del todo. Para poder conceptualizar lo qué es I.A., es importante saber qué es la inteligencia y cual es su función como tal. La inteligencia es el conocimiento que se adquiere por las actividades practicas y mentales que realizan los seres humanos relacionándose con todo lo que existe: otras personas, seres vivos y las cosas.[1]

Según Douglas Hofstadter, quién no tiene una definición de inteligencia, pero si plantea características como "actitudes esenciales de la inteligencia", contenidas en su libro Gödel, Escher Bach: Un eterno y orden y grácil bucle [2], y son las que se dan a continuación:

- "Responder de una manera muy flexible a las situaciones". No se responde necesariamente de las misma forma siempre que uno se enfrenta a un problema idéntico. Si se hiciera así se tendría una conducta mecánica en lugar de inteligente.

- "Sacarle el sentido a los mensajes contradictorios o ambiguos". Se entienden muchas declaraciones que aparecen como ambiguas o contradictorias, ya que somos capaces de colocarlos en un contexto, debido a nuestro conocimiento y experiencia.

- "Reconocer la importancia relativa de los diferentes elementos de una situación". Aunque se nos bombardea cada día con una cantidad ingente de información, se "da sentido" al mundo que nos rodea asignando diferentes niveles de importancia a los diferentes acontecimientos

---

[1] Lara y Velazquez, Etica y conocimiento del hombre, ed. Publicaciones culturales, México 1986, p. 13

[2] Citado en: Mishkoff C., Henry, A fondo: Inteligencia Artificial, Ediciones Anaya Multimedia, Madrid 1988, pp. 15 16

- "Encontrar semejanzas entre las situaciones, a pesar de las diferencias que puede haber entre ellas". Al reconocer semejanzas se puede basar las acciones futuras en lo que se ha aprendido en el pasado. Dos situaciones no tienen por que ser idénticas para que les apliquemos nuestra experiencia.

- "Extraer diferencias entre situaciones a pesar de las similitudes que pueda haber entre ellas". Aunque superficialmente dos situaciones puedan parecer similares somos capaces de notar diferencias y de actuar de acuerdo con ellas.

Todas las características mencionadas son fáciles o comunes de realizar para los seres humanos; y las podemos reconocer como sentido común de cada persona.
Según Cuena, "Un agente, en general, se considera inteligente en un tema (*entiende* de ese tema) cuando incorpora conocimientos sobre de él, da forma que al planteársele un problema de ese ámbito, encuentre fácilmente la solución, sin requerir (o requiriendo en menor medida) el esfuerzo de pruebas y ensayos estériles de otros agentes que "no entienden del tema",(1985.p.9).

Con base a lo revisado anteriormente, podemos decir que todas las actividades del hombre, tales como cocinar , razonar con base al sentido común, entender un lenguaje e incluso conducir un automóvil, se requiere necesariamente inteligencia.

En la actualidad las computadoras pueden realizar muchas actividades o habilidades humanas, como calcular, hablar, comparar números, dibujar, etc. Los investigadores de I.A. han estado en constante búsqueda de máximas aplicaciones para una mejor explotación de la computadora, a fin de que logren tener la capacidad de razonar, aprender o "acumular conocimientos". De todo lo anterior y más se encarga la I.A., que como ya se dijo es muy difícil de definir, pero se dan las siguientes definiciones de especialistas en esta área:

Rich Elaine (1983).- La I.A. es el estudio de cómo hacer que los ordenadores hagan las cosas que por el momento las personas realizan de una forma más perfecta.[3]

Larry Long (1990).- La I.A. se define como una expansión de las capacidades de las computadoras, que incluyen la capacidad de razonar , aprender, esforzarse para lograr la autoprotección e imitar habilidades sensitivas.[4]

José Cuena (1985).- La I.A. es un tipo de agente muy concreto: Es aquel sistema que incorpora, en forma manipulable por computador, el conocimiento sobre un tema y los procedimientos generales para encontrar respuestas, dentro de esa representación, a los problemas sobre un determinado tema.[5]

Avron Barr y Edward A. Feigenbaum (1982).- La I.A. es la parte de la informática relacionada con el diseño de sistemas de ordenadores inteligentes, es decir, sistemas que exhiben las características que asociamos con la inteligencia en la conducta humana.[6]

Bruce G. Buchanan (1984).- La I.A. es una rama de la informática que trata de la resolución de problemas por métodos simbólicos y no algorítmicos.[7]

Bruce G. Buchanan (1985).- La I.A. es una rama de la informática, que tiene como métodos para procesar la información la representación del conocimiento usando símbolos en lugar de números y la heurística o reglas basadas en la experiencia. [8]

---

[3] Kich, E., Inteligencia Artificial,ed. Gustavo Gili, Barcelona 1983, p9

[4] Long, L., introducción a la computación y al procesamiento de información, ed. Prentice Hall, México 1999, p. 252

[5] Cuena, José, Inteligencia Artificial: Sistemas expertos, ed. Alianza, Madrid 1985, pp. 9-10

[6] Citado en: Mishkoff, Henry C., Op cit, p. 14

[7] ibid p. 20

[8] ibidem p. 21

Brattle Research Corporation.- La I.A. trabaja con métodos de comparación de patrones para tratar de describir los objetos, acontecimientos o procesos en función de sus características cualitativas y lógicas y de sus relaciones.[9]

Así, la I.A. ha desarrollado a través del tiempo distintas técnicas informáticas, para valerse de ellas para la resolución de problemas. En otras palabras, los principales objetivos de la I.A. son:

1.- Simplificar actividades que generalmente realiza el hombre. Resolver problemas muy difíciles tan bien o mejor que un experto humano.

2.- Simular el proceso del pensamiento.. Razonar heurísticamente, utilizando reglas que los expertos humanos consideran eficaces..

3.- Crear nuevos métodos de uso amistoso para interactuar con los distintos mecanismos, como son: reconocimiento de la voz, reconocimiento de patrones, sintetizadores de voz y procedimientos de lenguajes naturales.

4.- Realizar tareas en forma automática. Funcionar con datos erróneos y reglas imprecisas.

5.- "Toma de Decisiones", según los mejores medios o caminos que se apeguen a la solución del problema. Contemplar simultáneamente múltiples hipótesis y alternativas.

Cada tecnología que surge, es necesario que se apoye en herramientas o técnicas para poder brindar medios al ser humano y pueda cada quien satisfacer sus necesidades o encontrar una alternativa de solución a un problema determinado.

La I.A. cuenta con herramientas muy particulares para aplicar su método de resolución que propone, así podemos encontrar técnicas o herramientas que son las siguientes:

---

[9] ibidem p. 21

a) Ordenadores

b) Procesamiento simbólico

c) Heurística

d) Comparación de patrones

a) Ordenadores. El ordenador, es la herramienta principal de la I.A., la cual ha aumentado su aplicación a menores costos. Las actividades que desempeñan los ordenadores son de mejor forma, de como las realizan las personas, dichas actividades son conocidas como mecánicas. Esto tuvo una notable consideración en la Revolución Industrial, donde se aceptaron las máquinas para eficientar y mejorar las labores de los seres humanos. En ocasiones, se dice que los ordenadores pueden superarnos en las tareas mentales que sean auténticamente mecánicas. Sin embargo este tipo de eventos ha llegado a dañar la autoestima de las personas y da lugar a que demuestren que pueden llevar a cabo las actividades de una mejor forma que un ordenador.

Pero en realidad no se pretende hacer superiores los ordenadores, sino que sean una herramienta muy útil, ya que los seres humanos no solamente procesamos información, la entendemos, la analizamos y la utilizamos como apoyo a la toma de decisiones y así aplicar nuevas ideas para mejorar en nuestras tareas.

Según Barr y Feigenbaum definen un ordenador inteligente, como uno que emula la conducta inteligente de los humanos, teniendo como objetivo principal de la I.A. el de desarrollar ordenadores inteligentes.[10]

Es muy difícil lograr desarrollar un ordenador inteligente, puesto que a través del tiempo no ha habido un acuerdo de lo que representa la inteligencia humana para ser simulada por un ordenador.

Como ejemplos de ordenadores tenemos:

1) Calculadora manual. Pueden realizar cálculos matemáticos rápidos y precisos, que para una persona resultaría muy difícil realizarlo.

---

[10] Citado en: Mishkoff, Henry C., Op cit, p. 15

2) Almacenamiento de información. Almacena grandes cantidades de datos como nombres, direcciones de miles de clientes, además de que puede llevar a cabo una cantidad considerable de transacciones.

3) Operaciones repetitivas. Son capaces de procesar o imprimir suficientes copias de informes, facturas, con la ventaja de que a pesar de que los ordenadores realizan las actividades una y otra vez, no llegan a aburrirse.

b) Procesadores simbólicos. La mayoría de las personas tiende a pensar simbólicamente en lugar de que sea numéricamente, es por ello que la inteligencia parece estar basada para manipular símbolos en lugar de números.

c) Comparación de patrones. Cuando ya se tiene cierta información de objetos, acontecimientos o procesos, podemos descubrir relaciones entre dichos aspectos y de este modo explicar como afectan estas relaciones entre si.

De esta forma para poder aplicar las técnicas de la I.A. se requiere conocimiento. El conocimiento debe ser al menos: ser voluminoso, ser complicado de caracterizar con precisión y estar cambiando constantemente. Así la I.A. se le es indispensable explotar el conocimiento a través de un método que debe de cubrir lo siguiente:

- Que capte generalizaciones. Las situaciones que comparten propiedades importantes se agrupan.

- Pueda ser comprendido por la gente que deba proporcionarlo. Los términos deben ser proporcionados o expuestos de manera que se entiendan o se comprendan.

- Puede ser modificado para corregir errores y para modificar cambios al mundo.

- Puede ser usado en muchas situaciones, incluso si no es totalmente preciso o completo.

- Puede ser usado para ayudar a superar su propia extensión absoluta, ayudando a incrementar el rango de posibilidades que deban considerarse.

Finalmente las técnicas de la I.A., deben diseñarse de acuerdo a las restricciones de los problemas de la I.A.

Un programa es un conjunto de instrucciones en secuencia lógica, que realiza el ordenador para la resolución de un problema. Para la programación se han utilizado o se siguen manejando los distintos niveles de programación que son generalmente tres:

- Lenguaje máquina. Es el nivel más básico de la operación de la computadora. Las instrucciones son codificadas como una serie de ceros y unos.

- Lenguaje ensamblador. Usa palabras cortas o utilizan símbolos de fácil reconocimiento llamados mnemónicos para representar instrucciones.

- Lenguaje de alto nivel. Es parecido al lenguaje común o natural, haciendo más fácil lo que se desea que haga el ordenador.

Para la I.A., los programadores de esta área han tenido que desarrollar sus propios lenguajes de programación para manejar símbolos más eficientemente. A estos lenguajes especializados de I.A. se les llama "Lenguajes de Procesamiento Lógico, o conocido también como: "Lenguaje de Procesamiento de Símbolos" o simplemente "Lenguajes de Programación de I.A.".

El primer lenguaje desarrollado para la I.A. fue el lenguaje IPL (Lenguaje de Procesamiento de Información), creado por Allen Newell, J.C. Shaw y Herbert Simon en 1965. Sin embargo, debido a su complejidad de manejo del IPL, por lo que se ha orientado a otro lenguaje que resulta más adecuado a la I.A., siendo el lenguaje LISP (Procesador de Listas), el más utilizado para las aplicaciones de I.A.. Este lenguaje fue desarrollado por

John McCarthy, debido a que anteriormente se había diseñado un programa de I.A. que se llamó Advice Taker. Este programa pretendía usar las técnicas de la lógica matemática formal para hacer deducciones y para que presentara "sentido común". Por lo consiguiente McCarthy diseño su propio lenguaje de programación, ya que no podía desarrollar este programa en ninguno de los lenguajes de programación existentes. El programa Advice Taker de McCarthy, sirvió como guía a muchos investigadores de I.A.; considerando a LISP como el primer lenguaje de procesamiento lógico mas usado.

Además de crear LISP, se implementaron nuevos lenguajes de programación para resolución de problemas específicos en I.A.. En respuesta surgen los siguientes lenguajes de programación:

- POP-2 de la Universidad de Edimburgo.
- PROLOG (PROgramación LOGica), de la Universidad de Marsella.
- SAIL, de la Universidad de Stanford.
- SmalTalk tm, de la Xerox PARC.

Cabe destacar la aplicación del lenguaje de programación PROLOG, que fue desarrollado en Francia en 1973, se le conoce como un "sistema que demuestra teoremas" y utiliza la técnica lógica formal llamada "cálculo de predicados". En los E.U. PROLOG se usa como complemento a LISP, añadiendo algunas características a LISP, para que resuelva ciertos problemas.

La I.A. se va a encargar de resolver ciertos problemas que están a su alcance, por lo que se hace necesario conocer que tipo de problemas caen en el campo de acción de la I.A .

- Juegos.
- Resolución general de problemas.
- Visión
- Comprensión del lenguaje natural.
- Matemática simbólica.
- Análisis de química.
- Demostración de teoremas.
- Percepción
- Reconocimiento de voz
- Resolución de problemas expertos.
- Diagnósticos médicos.
- Diseño de ingeniería.

Es indudable que hoy en día las tecnologías de punta como los SE's y la I.A. se han convertido en herramientas poderosas de desarrollo y competitividad. En nuestro país existe un enorme mercado potencial para estos sistemas, aunque ciertamente nos encontramos en un momento en el que la industria aún se muestra escéptica al respecto. En otros países, los SE's han aportado notables beneficios en todos los órdenes. Esto implica que en muy pocos años las entidades productivas de nuestro país, necesitarán estar plenamente involucradas en el área y para tal efecto demandar profesionales en el campo de inmejorables cualidades. Conscientes de lo anterior dentro del Plan de Estudios de ICOM, se incluyen las materias de SE's e I.A. y apoyándonos en éstas pretendemos poder hacer que la Universidad Nacional Autónoma de México en su Campus "ARAGÓN" forme parte importante en el desarrollo de esta tecnología. En ésta investigación nos concretaremos a un proyecto que se concibe como un apoyo a dicho proceso y en el cual se propone un estudio integral de la infraestructura necesaria para el establecimiento de un Laboratorio de Desarrollo de Sistemas Expertos dentro del Campus "ARAGÓN" de la UNAM. Es evidente que la educación y la capacitación son los pilares de la tecnología, por lo tanto es necesario contemplar todo posible acercamiento a una tecnología que ha mostrado ya contundentes resultados en la industria, y que a la luz de los expertos ser la que determine la línea de desarrollo en los próximos años.

1.2 OBJETIVOS.

Los objetivos de este estudio son:

- Investigar cuáles son las posibilidades de la UNAM Campus "ARAGÓN" para establecer una infraestructura de desarrollo de sistemas expertos.

- Desarrollar una Guía donde se describan las actividades académicas, y los temas a desarrollar que se llevaran a cabo en dicho laboratorio.

Para cumplir el primer objetivo, se realizará (1) una cuantificación de los recursos de la Universidad que se podrían utilizar para desarrollar sistemas expertos; (2) un inventario de profesionistas dedicados a esta especialidad que podrían apoyar este proyecto, y (3) un análisis de la manera en la que otras instituciones educativas y de investigación han establecido laboratorios de este tipo.

El segundo objetivo se alcanzará mediante una revisión de la literatura de sistemas expertos, con explícita referencia a las herramientas de construcción de los mismos; además de un análisis de los planes de estudio y/o de trabajo de otros centros de desarrollo que ya operan en el país, tanto en los sectores académicos, como en los industriales y comerciales.

1.3 ALCANCES.

Se busca el establecimiento de un laboratorio de desarrollo de sistemas expertos ya que las ventajas de utilizar sistemas expertos ha sido comprobada, tanto en experiencias académicas, como en experiencias comerciales-industriales; es imperante la necesidad de que las instituciones académicas estrechen aún mas la brecha entre el desarrollo académico y el proceso productivo, en pro de la formación de profesionales cada vez más competitivos; cualquier institución que desee participar activamente en el desarrollo futuro de un país como el nuestro, debe involucrarse ampliamente en tecnologías de vanguardia tales como los sistemas expertos; el establecimiento de un laboratorio como éste permitirá a los estudiantes complementar su formación en una área de gran interés para la Ingeniería y para otras disciplinas profesionales.

Como consecuencia de la gran cantidad de información susceptible de ser recopilada se proponen los siguientes alcances y limitaciones al proyecto.

El laboratorio debe ser una propuesta formal, pero la virtual implementación de éste se restringe a los recursos y posibilidades que las autoridades universitarias decidan otorgar.

La instalación de paquetes de software y lenguajes de desarrollo se contempla sólo en caso que se encuentren al alcance de la Universidad y/o que se cuente con el apoyo del proveedor.

Asimismo, las pruebas realizadas en las distintas herramientas estarán restringidas a las facilidades que las distintas instituciones que cooperen con en el proyecto deseen otorgar.

El sondeo del mercado mexicano queda restringido a proyectos quede alguna forma hayan probado su efectividad o que se encuéntren lo suficientemente documentados como para obtener evaluaciones importantes de ellos y de la disponibilidad de información que los desarrolladores, investigadores y líderes de proyecto estén dispuestos a ofrecer.

Los proyectos de ingeniería realizados fuera del país podrán ser un sustento teórico importante del proyecto, pero su verificación práctica queda totalmente descartada, salvo en casos verdaderamente excepcionales.

Las experiencias recogidas en cursos, simposiums, reuniones y seminarios, nacionales e internacionales, también constituir importantes aportaciones al proyecto.

1.4 METODOLOGIA.

El problema de recopilación de información sobre tecnologías de punta, como los sistemas expertos, se caracteriza principalmente por:

La información que aparece es muy teórica y abstracta

No plantea casos de la vida real

No hay análisis de resultados prácticos

Y particularmente en México y Latinoamérica la mayoría de los trabajos versan sobre temas académicos e investigaciones en áreas muy especializadas, lo cual deja muy pocos trabajos acerca de SE's. Por lo tanto se plantearon una serie de actividades tendientes a dar dirección y ejecución al proyecto:

- Se realizarán visitas a centros de investigación dentro del país.

- Se solicitará información a bancos de datos de instituciones como CONACYT, INEGI y la Sociedad Mexicana de Inteligencia Artificial entre otras.

- Se establecerá contacto con proveedores de software y hardware.

- Se investigarán planes de estudio de distintas instituciones de enseñanza superior.

CAPITULO 2
TEORIA DE SISTEMAS EXPERTOS

2.1 INTRODUCCION

En este capítulo se hace una descripción a grandes rasgos de, ¿que es un sistema experto?, ¿cuáles son las partes que lo conforman?, y la importancia de cada uno de sus elementos. En la sección 2.2 se discuten las definiciones que algunos autores han dado sobre SE's. Posteriormente, en la sección 2.3 se menciona la importancia del conocimiento y el papel que juega en esta tecnología además se da el concepto de Ingeniería de Conocimiento y se menciona el lugar que ocupa en la construcción de SE's. En la sección 2.4 se hace una descripción a cierto detalle, de los diferentes esquemas de representación de conocimiento que existen. Con base en lo anterior, en la sección 2.5 se muestran los elementos que describen la estructura de un SE, y como los esquemas de representación influyen en ella. Así mismo se exponen otros módulos como la interface con el usuario y los mecanismos de explicación. En la sección 2.6 se describe el Ciclo de vida que sigue un SE y en la sección 2.7 hablamos de Arquitecturas de SE's más comunmente usadas.

2.2 DEFINICION DE UN SISTEMA EXPERTO

Dentro de las ramas que abarca la Inteligencia Artificial tenemos la de los Sistemas Expertos. Un Sistema Experto (SE) es una aplicación de computadora que resuelve problemas complicados que de otra forma requerirían intensamente de la participación de un experto humano para su solución. Para lograrlo, los Sistemas Expertos simulan el proceso humano de razonamiento aplicando conocimientos específicos e inferencias. Dichos conocimientos son provenientes de expertos en un dominio específico, y son alimentados al sistema previa ejecución del mismo. Una definición aceptada de un SE es la siguiente:

Un Sistema Experto es un programa que tiene un amplia base de conocimientos en un dominio restringido, y que utiliza razonamiento inferencial complejo para desempeñar tareas que un experto humano puede desarrollar [Winston 1987].

Para completar esta definición, algunos autores consideran que un SE debe poseer las siguientes características:

- Conocimientos específicos extensivos del dominio de aplicación.
- Aplicación de técnicas de búsqueda.
- Apoyo para análisis heurísticos.
- Capacidad de inferir nuevos conocimientos a partir de los existentes.
- Procesamiento simbólico.
- Habilidad de explicar su propio razonamiento.

Los SE's han sido entusiastamente aceptados por muchas compañías de diversos giros, como una forma de tener 'expertos' disponibles en cualquier lugar y tiempo que se requiera. Particularmente, estos sistemas han sido mejor aceptadas en aquellos campos en los que los expertos humanos existentes son caros, y no fáciles de consultar.

2.3 EL PAPEL DEL CONOCIMIENTO EN SISTEMAS EXPERTOS.

La parte quizá más importante dentro del diseño de SEs es la que se refiere a la adquisición y representación de los conocimientos que guiarán el compartamiento del sistema. Podemos decir que el poder de un experto estriba en la extensión de los conocimientos que posee sobre un campo dominio especifico. Existen varios componentes del conocimiento que son fuente del desempeño de un experto, tales como:

- *Hechos:* Afirmaciones que relacionan elementos de verdad dependientes del dominio del que se trate. Por ejemplo:
  El pasto es verde.
  Los perros son mamíferos.
  Ethernet es una red apegada al estándar IEEE 802.3.

- *Reglas de Procedimiento (Procedural Rules):* Reglas invariantes y bien definidas que describen secuencias fundamentales de eventos y relaciones relativas al dominio. Por ejemplo:

"SI existe una línea de alta tensión en la trayectoria de un cable coaxial para transmisión de datos ENTONCES habrá que evitar localizarlo cerca de esta línea."

- *Reglas Heurísticas.* Reglas generales en forma de "corazonada" que sugieren procedimientos a seguirse cuando la reglas procesales invariantes no se encuentran disponibles. Por ejemplo:

  "Es mejor tratar de ajustar el presupuesto que castigar el desempeño de una red de transmisión de datos."

  La presencia de heurísticas ha contribuido enormemente a la potencialidad y flexibilidad de los SEs y es el tipo de conocimiento que distingue a estos sistemas de los de programación tradicional.

- *Razones.* Son las explicaciones que se dan al consultante, del porque la elección de tal o cual solución. Por ejemplo:

  "la solución "x" fue elegida por ser la más optima en ese rango de evaluación"

Además de éstas formas específicas del conocimiento, un experto posee también un modelo conceptual general del dominio en el cual se desempeña y un esquema general de búsqueda de soluciones. Estas vistas globales forman la estructura básica de aplicación del conocimiento de un experto.

2.3.1 Ingeniería Del Conocimiento

Edward Figenbaum define a la Ingeniería del Conocimiento como sigue:

" *La Ingeniería del conocimiento practica el arte de traer los principios y herramientas de investigación en IA, para apoyar aplicaciones a problemas difíciles que requieren conocimiento experto para su solución. El objetivo técnica de adquirir éste conocimiento, representarlo, y usarlo apropiadamente para construir y explicar líneas de razonamiento, son problemas importantes en el diseño de sistemas basados en conocimiento ..... El arte de construir agentes inteligentes es una parte y una extensión del arte de programar. Es el*

arte de construir programas complejos de computadora que representen y razonen con el conocimiento del mundo". [Figenbaum, 1977]

La Ingeniería del Conocimiento, es el proceso a través del cual se adquiere conocimiento, se construye un esquema de representación, y se genera una base de conocimiento relativa a un dominio. A pesar de que el conocimiento puede ser extraído de diversas fuentes, incluyendo documentación, sistemas de información, etc., la mayor parte debe ser obtenida de expertos humanos. El ingeniero del conocimiento, (IC) es la persona que dirige este proceso de extracción y representación, de manera tal, que de lugar a una base de conocimiento útil al sistema.

Para adquirir el conocimiento necesario, el IC debe establecer primeramente un entendimiento general del dominio del experto, tanto de su vocabulario como de sus expresiones y fundamentalmente de los conceptos claves. La adquisición del conocimiento es frecuentemente la función más difícil de realizar en el desarrollo de SEs. Esto se debe primeramente, al hecho de que el proceso requiere de una extensa comunicación humana entre el experto y el IC y de ahí sus subsecuentes dificultades. Como resultado, la adquisición de conocimiento es un proceso no muy bien definido ni entendido.

2.3.1.1 Representación Del Conocimiento

Uno de los problemas inherentes a la ingeniería de conocimiento es la selección del esquema de representación adecuada al problema. Estos esquemas hoy en día son resultado de una investigación exhaustiva y de un detallado análisis empírico, más que de una consideración filosófica de alguna teoría de representación. No existe una representación universalmente aceptada como la mejor, y un esquema puede resultar más aplicable que otro cuando se evalúan en torno a un dominio específico. Sin embargo, se puede evaluar cualquier esquema de representación de acuerdo a una serie de importantes criterios generales:

Transparencia: Indica la facilidad con la que se puede identificar el conocimiento almacenado (sin ambigüedades).

Explicidad. La forma en la cual el conocimiento es explícitamente representado.

Naturalidad: La facilidad con la cual el conocimiento es representado en su forma nativa.

Eficiencia: La relativa facilidad con la que se puede acceder el conocimiento durante la ejecución.

Adecuación: El punto en el cual una estructura dada puede ser usada para representar todo el conocimiento requerido por el sistema.

Modularidad: La facilidad de almacenar fragmentos del conocimiento en forma independiente de los otros.

2.3.1.2 Nivel De Representación

El conocimiento involucrado en el proceso de razonamiento, puede encontrarse a diferentes niveles de detalle. Los elementos del nivel más bajo (conocimiento de bajo nivel), conocidos como primeros principios, forman los bloques fundamentales de construcción sobre los que se basa el dominio del experto.

De estos principios más específicos, se desarrollan teoremas y acciones de reglas (conocimiento de alto nivel). Estos a su vez, forman la base de derivaciones adicionales de conocimiento. Este es un proceso fundamentalmente de síntesis.

El conocimiento de bajo nivel es raramente de valor cuando se aplica directamente. Cualquier estudiante que ha tomado un examen final a libro abierto sin la suficiente preparación, es consiente de lo difícil que resulta leer conocimiento de bajo nivel, para contestar preguntas que requieren aplicación de conocimiento de alto nivel generado a partir de éste. La eventual derivación de una solución, con tiempo ilimitado, no se considera un comportamiento experto.

Desafortunadamente, la flexibilidad del conocimiento decrece conforme aumenta su nivel. Esto ocurre porque muchos elementos específicos son combinados en una forma muy particular para desarrollar conocimiento de alto nivel; y el conocimiento sólo es útil, cuando la presupuesta combinación de elementos está presente. Por otro lado, contado sólo con conocimiento de alto nivel, no existe forma de responder a una nueva e imprevista situación; tal respuesta requiere razonamiento originado en el conocimiento de bajo nivel.

2.4 ESQUEMAS DE REPRESENTACION

Los esquemas de representación del conocimiento pueden ser clasificados a grosso modo en dos diferentes tipos: declarativos y de procedimiento. Los esquemas declarativos representan al conocimiento como una acumulación de hechos estáticos junto con una limitada información que describe como ser utilizado.

2.4.1 Lógica Formal

La lógica formal, fué una de las primeras representaciones del conocimiento que se utilizaron en IA. La forma lógica más ampliamente usada es la llamada lógica de predicados de primer orden, la cual se forma de cuatro componentes básicos: un alfabeto, un lenguaje formal, un conjunto de preceptos básicos llamados axiomas, y un conjunto de reglas de deducción.

Cada axioma describe un fragmento de conocimiento y las reglas de deducción aplicadas a éstos derivan nuevos preceptos verdaderos.

El alfabeto consiste de símbolos a partir de los cuales, se construyen preceptos en lenguaje formal. El alfabeto de la lógica de predicados contiene elementos como: predicados, variables, funciones, constantes, conectivas, cuantificadores y delimitadores.

Las constantes se usan para representar un elemento específico del dominio. Se estila escribirlas con letras capitulares como distinción.

Las variable es utilizada para representar al miembro de un conjunto del dominio, sin especificar alguno. Se escriben con letras minúsculas para su identificación.

Las funciones describen elementos del dominio, identificándolos como resultados únicos de aplicar un mapeo entre otros elementos del dominio. Como un ejemplo, considérese el predicado:

perro (RUFO)

La función perro indica un único elemento del dominio que describe a RUFO como un perro. Nótese que la sintaxis válida es escribir el nombre de la función en minúsculas, su delimitadores y sus argumentos mismos que pueden ser constantes, variables y/o otras funciones.

Los predicados se usan para describir relaciones dentro del dominio, indicando la forma específica en cómo un elemento se relaciona con otro. Como ejemplo:

pertenece (perro (RUFO), DANIEL)

Podría indicar que el perro RUFO pertenece a DANIEL.

Utilizando fórmulas más complejas se pueden construir proposiciones compuestas usando las conectivas, mismas que se describen a continuación:

Y conjunción: Se usa para construir fórmulas que indiquen que todos sus componentes deben ser verdaderos para que toda la proposición sea verdadera.

v o disyunción: Se utiliza para construir fórmulas cuyo valor de verdad dependa solo de alguno de sus elementos.

→ *implicación:* Se utiliza para construir implicaciones SI-ENTONCES en los que si el antecedente es verdadero el consecuente también lo es.

↔ *equivalente:* Se utiliza para indicar equivalencia lógica entre dos fórmulas

¬ *negación:* Es usado para cambiar el valor de verdad de una formula de verdadero a falso y viceversa

Los cuantificadores complementan a los elementos anteriores para poder expresar las proposiciones de la lógica proposicional, un subconjunto de la lógica de predicados. Los calificadores son: el cuantificador universal ("para todo") y el cuantificador existencial (existe al menos uno).

2.4.2 Redes Semánticas

Las redes semánticas se concentran en la representación gráfica de las relaciones entre los elementos del dominio. Los componentes básicos de las mismas, son los nodos y las ligas (o arcos). Los nodos se usan para representar elementos del dominio y las ligas para describir relaciones entre éstos. Una liga pude visualizarse como algo que se afirma ser cierto acerca de un elemento relacionado con otro, mostrando una relación binaria

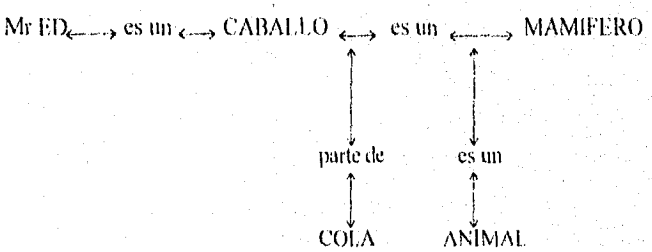
Ejemplo: El perro es un carnivoro se puede representar como:

PERRO ⟶ es un ⟶ CARNIVORO

Las relaciones binarias más comunes utilizadas en redes semánticas son:

es un y una parte de.

Cualquier caracteristica definitoria como color, tamaño, textura de un elemento del dominio, puede ser representada como una propiedad asociada con un nodo. De esta forma se puede crear una red que diga:



De donde podemos inferir que Mr ED es un caballo y por lo tanto un mamifero y por lo tanto es un animal y que además tiene cola. Estas relaciones describen una propiedad de

herencia que hace a las redes semánticas particularmente útiles para representar dominios que pueden ser estructurados como taxonomías.

Aunque la representación de redes semánticas es flexible su escasa formalidad puede ser una desventaja. Las ramas de redes más complejas, frecuentemente se pueden entrelazar y originar a confusiones. Así estos diagramas pueden ser difíciles de manejar y validar. La representación de Warnier/Orr [Carrico, 1989] fué propuesta para resolver estos problemas. Dicha representación consiste en convertir las redes semánticas en estructuras de datos jerárquicas, las cuales permiten normalizar los datos y son mucho más fáciles de validar.

El rearreglar las redes en una forma jerárquica es una técnica de normalización. Esto explica porque se prefiere esta representación para la codificación de sistemas basados en reglas. Por si mismo este esquema conduce a relaciones causa-efecto. Por su claridad estos diagramas son fáciles de construirse y de editarse. Algunas herramientas de alto rendimiento permiten este tipo de diagramación.

En las practicas convencionales de ingeniería de software, las redes semánticas son conocidas como modelos de datos. Su propósito es el de modelar objetos y relaciones al mismo tiempo que los normaliza para producir representaciones más claras. Estas mismas, frecuentemente se convierten en árboles de decisión para su codificación.

2.4.3 Marcos

Inicialmente introducidos por Minsky en 1957, Un marco, es una estructura para organizar el conocimiento y cuya fuerza radica en el manejo del conocimiento por ausencia (del inglés default knowledge). Los marcos comparten varios conceptos comunes con las redes semánticas. Cada marco representa una clase de elemento, de la misma manera en que una clase de nodo representa a un elemento de la red. Es común desarrollar redes en las cuales los nodos son marcos.

Un marco consiste de una serie de nichos que representan una propiedad o atributo del elemento representado. El nicho almacena sistemáticamente un componente de una experiencia pasada de acuerdo al tipo de los elementos.

Cada nicho se identifica por el nombre del atributo que le corresponde, incluyendo el valor o rango de valores con los que puede ser asociado. Un valor por ausencia, también puede ser indicado.

Los marcos se apoyan fuertemente en la propiedad de herencia, de manera semejante a las redes semánticas. Cualquier clase de objetos, puede ser incluida en diferentes marcos que representen objetos a diferentes niveles de especificación. Bajo estas condiciones, a cualquier nivel de especificación se pueden heredar valores para los nichos correspondientes, de niveles superiores de abstracción. Ejemplo:
Se define un trabajador de Administrativo de la UNAM con los siguiente atributos:

FRAME: Administrativo
Especialización: De la UNAM
Sexo:
Rango: (Masculino, Femenino)
Nombre:
Rango: (Ricardo, Jesús, Alberto, Oscar, Diana, Claudia)
Turno:
Rango: (Matutino, Vespertino, Indeterminado)
Categoria:
Rango: (A, B, C, D)
Salario: $ 1,500.00

Ahora se define un Trabajador en particular

FRAME Oscar
Especializacion: Administrativo
Sexo: Masculino
Turno: Matutino
Categoria: A

ya no es necesrio describir a Oscar con Salario de $ 1,500.00, por que es un atributo que puede heredar de un nivel superior. Esto es entonces, una forma poderosa y eficiente de razonamiento, ya que es posible obviar algunos atributos con la seguridad de que al ser necesarios el sistema los podrá inferir de sus ancestros.

Es posible también asociar algunos procedimientos para actualizar los valores de los nichos. Los procedimientos asociados más comunes son los procedimientos si-fuese-necesario (if-needed) y si-es-agregado (if-added). A éstos procedimientos se les denomina demonios (demons).

2.4.4 Scripts

Los scripts o guiones describen una forma de representación similar a las anteriores. Con las redes semánticas y los marcos la expectación era entorno a clases de objetos. En los guiones las espectativas son acerca de la secuencia de eventos que "deben" ocurrir en una situación dada. Esta esperanza se basa en la observación de patrones de eventos recurrentes en situaciones similares observadas en el pasado; Por ejemplo:

Si una persona entra a un restaurante entre las 14:00 hrs y las 16:00 hras, la secuencia de eventos sugiere que esa persona va a comer y que primeramente ordenar la carta para revisar el menú, luego ordenar la sopa y el guisado después. para finalizar con un postre y un café y por último pedirá la cuenta, revisarla y finalmente pagarla .

En el ejemplo anterior no se han utilizado juicios arbitrarios, sino que se ha construído un guión en base a una secuencia de eventos de un patrón recurrente que la experiencia ha probado.

Un guión es una estructura que se usa para almacenar prototipos de secuencias esperadas de eventos. Esta estructura puede formarse a partir de varios elementos, algunos de los más comunes son:

Condiciones de entrada: Son las circunstancias iniciales que deben existir para que un guión sea aplicable.

Resultados: Condiciones que aparecerán verdaderas después de que los eventos ocurran según el guión.

Roles: Patrón de comportamiento de los agentes que llevan acabo las acciones descritas en el guión.

Escenas: Secuencias específicas de eventos que conforman el guión.

No existen reglas absolutas para definir los elementos genéricos de un guión, es plausible incluir nuevos descriptores asociados a situaciones específicas.

2.4.5 Sistemas De Producción

Los sistemas de producción son el esquema de representación de conocimiento más comúnmente usado en la construcción de SE's. Sus componentes fundamentales son:

La memoria de trabajo: Es donde se almacena el estado del mundo; es decir, ahí se guardan todos los parámetros con sus valores que describen la situación actual del problema consignado al sistema.

La memoria de producción: Es el espacio reservado para cargar el conjunto de reglas que determinan la acción a seguir por el sistema.

Las reglas de producción: Son también llamadas pares condición-acción y cuya sintaxis generalmente corresponde a formas IF-THEN que constituyen la descripción del conocimiento del experto humano a su más alto grado de especialización. Estas reglas también se les suele separar por elementos como "el lado izquierdo de la regla" (antecedente) y "el lado derecho de la regla" (consecuente).

El interprete: Este módulo examina el estado mundo y decide que reglas serán disparadas de acuerdo al algoritmo de resolución de conflicto y la estrategia del motor de inferencia. Este módulo es de los más interesantes de los sistemas de producción, por que en él se encuentran dos partes fundamentales que describen la forma de razonamiento del sistema. El solucionador de conflictos y el motor de inferencia, son elementos que afectan directamente el desempeño del sistema.

2.5 ESTRUCTURA DE UN SISTEMA EXPERTO

Los sistemas Expertos utilizan una amplia variedad de estructuras, fundamentalmente por que una puede resultar mejor que otra para una aplicación determinada. La investigación en esta área sigue en marcha, y aún queda mucho por obtener.

A pesar de las significativas diferencias que puedan encontrarse entre las estructuras, existen elementos generales comunes a toda ellas, tales como:

- La Interfaze con el usuario.
- El Sistema de Almacenamiento y Generación de Conocimiento.
- Sistema de Inferencia y Almacenamiento de Conocimiento
- Base de Conocimiento.
- Memoria de trabajo
- Sistema de Explicación.

El sistema de almacenamiento e inferencia, está compuesto de una Base de Conocimiento y un Motor de Inferencia. La función de este último es la de extraer el conocimiento del almacenamiento e inferir nuevos conocimientos a partir de este.

Base de Conocimiento: La Base de Conocimiento representa el depósito de las primitivas del conocimiento disponibles al sistema. Este almacén del conocimiento es el que permite que el sistema actúe como experto.

En general, el conocimiento es almacenado en forma de hechos, reglas, pero los esquemas específicos usados para almacenar la información varia ampliamente según el formato establecido: reglas, guiones, etc... El diseño del esquema de representación del conocimiento, impacta definitivamente el diseño del motor de inferencias, al igual que el proceso de actualización de conocimientos, el de explicación del razonamiento y en general la eficiencia del sistema.

La selección del esquema de representación del conocimiento es una de las decisiones criticas en el diseño de SE.

Mecanismo de Inferencia: Las SE's deben, por su naturaleza, tratar flexiblemente con situaciones variantes. La capacidad de responder a esta variedad de situaciones depende de la habilidad de inferir nuevo conocimiento a partir del existente. Consideremos el siguiente ejemplo:

1.- Todos los animales respiran oxígeno.

2.- Todos los perros son animales.

Un nuevo hecho, "todos los perros respiran oxígeno," puede ser inferido a partir de los dos primeros hechos. Para responder a una situación determinada, el SE debe aplicar el conocimiento apropiado a tal efecto, lo cual supone que el conocimiento requerido existía como tal en la base de conocimiento o que fue inferido a partir de éste.

El proceso de búsqueda del conocimiento apropiado para a partir de este, inferir nuevos conocimientos, es el elemento clave del procesamiento de los SE's.

Por supuesto, en el ejemplo anterior era absolutamente válido el almacenar el hecho de que "todos los perros respiran oxígeno." en lugar de haberlo inferido. Sin embargo, una de las más grandes dificultades involucradas en la manipulación de primitivas, es que aunque sea un número pequeño de éstas, pueden ser combinadas en un número muy grande de opciones únicas. De ahí que el numero de posibles combinaciones para un número grande de primitivas sea astronómico. A este fenómeno se le conoce como explosión combinatoria. En otras palabras, no podemos programar todos los posibles resultados de combinar las primitivas.

El motor de inferencia es la parte del sistema que localiza las primitivas (incluidas en la base de conocimiento) e infiere nuevo conocimiento a partir de éstas. El paradigma de este mecanismo es la estrategia de búsqueda para obtener el conocimiento requerido. Muchos y diferentes paradigmas son usados en SE's, pero la mayoría están fundamentados por lo menos, en uno de dos conceptos:

encadenamiento hacia atrás, el cual consiste en un razonamiento de arriba a abajo, que inicia en la meta deseada y trabaja hacia atrás a través de las condiciones requisito, o

encadenamiento hacia adelante, el cual es un proceso de razonamiento de abajo hacia arriba que comienza con condiciones conocidas y se desplaza hacia la meta deseada.

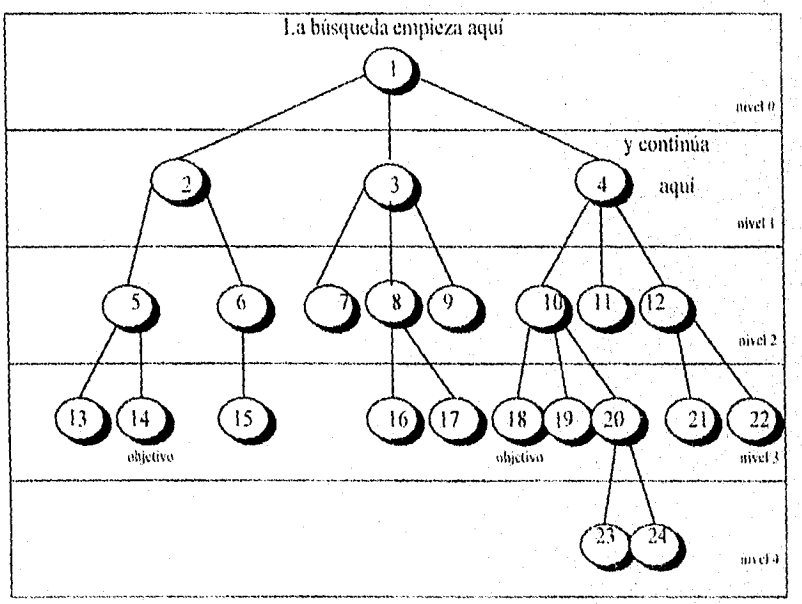
La selección del mecanismo de inferencia, considerando la explosión combinatoria, influirá fuertemente en el desempeño general del sistema experto.

2.5.1 Estrategias De Búsqueda

Existen cuatro formas fundamentales de examinar un árbol de búsquedas: búsqueda en anchura, búsqueda en profundidad, búsqueda progresiva y búsqueda regresiva. Estos metódos determinan el orden en que se examinan los nodos del árbol..

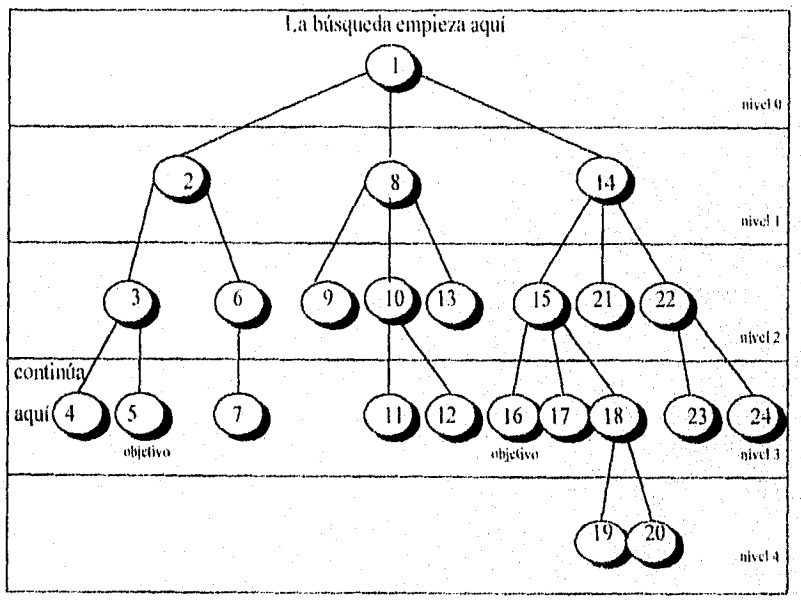
2.5.1.1 Búsqueda En Anchura

Una búsqueda en anchura comienza en el nodo raíz y continúa hacia abajo, examinando todos los nodos de un nivel antes de pasar al siguiente. Como se ilustra en la siguiente figura, la búsqueda en anchura examina de izquierda a derecha todos los nodos (identificados por números) de un nivel antes de pasar al siguiente nivel. Esta búsqueda terminaría en el nodo 14, el primer objetivo.



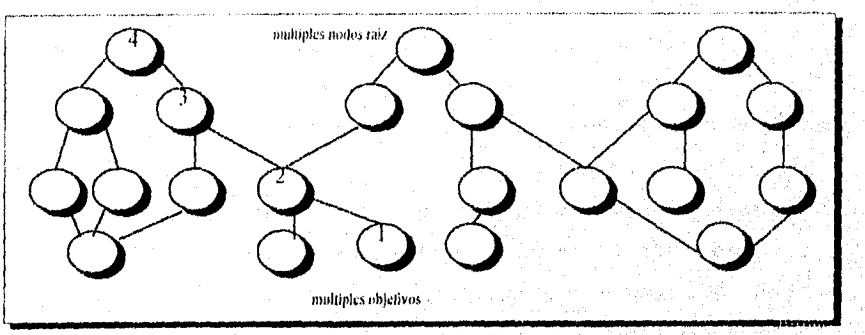
2.5.1.2 Búsqueda En Profundidad

Las búsquedas en profundidad comienzan en el nodo raíz y van recorriendo los distintos niveles del árbol hasta alcanzar el nodo más profundo de la rama más a la izquierda. Si el nodo final de dicha rama no es uno de los objetivos, se retrocede hasta un nivel en que se pueda continuar la búsqueda. El proceso de búsqueda continúa de izquierda a derecha hasta que se encuentra una solución. El sistema de búsqueda en profundidad se ilustra en la siguiente figura. La secuencia de búsqueda seguiría las etiquetas indicadas en la figura y terminaría en el nodo 5, el primer objetivo encontrado.



2.5.1.3 Búsqueda Regresiva

En las técnicas de búsqueda regresiva denominadas también de "razonamiento regresivo" o de "encadenamiento regresivo", la Búsqueda comienza en el nodo objetivo. Una forma de resolver un problema consiste en asumir que la solución tiene una forma determinada y entonces buscar pruebas que sustenten dicha suposición. La siguiente figura presenta una de las posibles secuencias de búsqueda de encadenamiento regresivo. Se ha presentado una aproximación de búsqueda en profundidad, pero podía haberse elegido una de búsqueda en anchura. Se observa que existen varios resultados posibles así como varios nodos raíz. Algunos problemas se pueden adaptar a este modelo. Si la búsqueda basada en un objetivo falla, puede iniciarse otra partiendo de otro objetivo e intentando encontrar hechos que soporten dicho objetivo. Este tipo de búsquedas se denominan controladas por objetivos y simulan un tipo de razonamiento inductivo. Para demostrar la validez del objetivo seleccionado se puede recorrer en sentido inverso el camino de búsqueda.



Las técnicas de encadenamiento progresivo y regresivo son muy utilizadas en la investigación de sistemas expertos. El razonamiento regresivo se usa más que el razonamiento progresivo, pero ambos sistemas son muy utilizados. La estrategia de búsqueda se elige de forma que se adapte a la clase de problema que se está resolviendo. Además, algunos sistemas utilizan una mezcla de búsqueda regresiva y progresiva que reduce el tiempo de búsqueda y acelera la resolución de problemas. Más adelante, se considerarán de nuevo las técnicas de encadenamiento progresivo y regresivo.

2.5.1.4 Búsqueda Heurística

Anteriormente se ha utilizado el término heurístico para describir una regla que ayuda a expresar el conocimiento. Pero el término también se puede emplear para describir las reglas que se emplean para limitar la amplitud de las búsquedas. Las técnicas heurísticas se emplean para centrar la atención de la búsqueda sólo sobre aquellas partes del árbol de búsqueda en las que es más probable que se encuentre la solución. Las técnicas heurísticas pueden eliminar gran parte del árbol de búsqueda, acelerando notablemente el proceso de solución del problema.

Existen técnicas de propósito general. Un ejemplo de técnica de búsqueda heurística de propósito general son las técnicas de búsqueda en profundidad limitada. Esta técnica se usa en la búsqueda en profundidad para evitar que la búsqueda tenga lugar en algunas bifurcaciones muy bajas y en las que no existe posibilidad de encontrar una solución. Una búsqueda limitada acota arbitrariamente el nivel máximo de profundidad de búsqueda. Una vez alcanzado éste se corta el proceso de búsqueda, forzando su continuación en un punto anterior. La dificultad es, evidentemente, saber a qué profundidad hay que cortar, de modo que no se eliminen soluciones potenciales.

Las búsquedas heurísticas de propósito específico sólo son aplicables a ciertos tipos de problemas. Un tipo de búsqueda heurística de propósito específico es el uso de reglas especiales llamadas "metarreglas", que establecen la forma de utilizar las reglas de conocimiento. A partir de una serie de datos, una metarregla limita la búsqueda al subconjunto de la base de conocimientos que más probablemente llevará a una solución. Con esta aproximación, se crea una pequeña base de conocimientos sobre cómo guiar el proceso de búsqueda de la forma más eficiente.

Otra forma de minimizar el proceso de búsqueda consiste en subdividir el problema y la base del conocimiento en una jerarquía. De esta forma, la búsqueda puede restringirse a las porciones de la jerarquía que más probablemente llevarán a una solución rápida.

2.5.2 Razonando Con Redes Semánticas

El razonamiento con redes semánticas es generalmente en una sola dirección, por que las asociaciones pueden ser hechas, simplemente rastreando los enlaces del sistema. Por ejemplo, se puede aplicar un enlace de inferencia a la red simple que se muestra y concluir que 12 es mayor que 3:

12 ⟶ es mayor que ⟶ 7 es mayor que ⟶ 3

Desafortunadamente, no existen reglas rigurosas que guíen dicho razonamiento. En sistemas como el de lógica de predicados, el razonamiento procede sobre la base de la uniformidad semántica de la manipulación de los símbolos de representación. Las inferencias resultantes de este proceso, pueden ser irrelevantes pero siempre válidas. Esto no siempre es cierto en las redes semánticas.

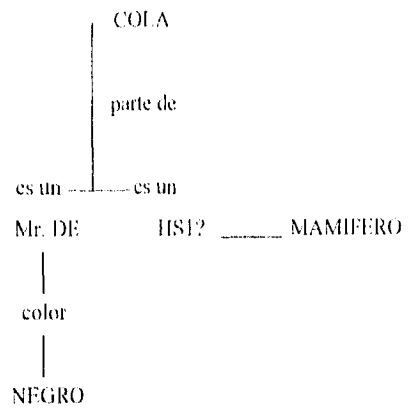
Si se navegan los enlaces simplistamente, se pueden desprender inferencias no válidas porque las relaciones representadas no son absolutamente rigurosas (en particular, porque no se reconocen condiciones de excepción). Por ejemplo, usando la red mostrada en la sección 2.4.2, aunque todos los caballos son mamíferos y todos los caballos tiene cola, existen algunos, conocidos como "bobtails", que no tiene cola.

Otra forma comunmente utilizada para inferir con redes semánticas, es la técnica de apareamiento.

Un procedimiento de apareamiento, se fundamenta en la construcción de un fragmento de la red con mezcla de nodos con valores definidos y nodos cuyos valores son deseados, pero desconocidos. Los valores deseados se representan con variables. Un fragmento como el descrito se muestra en la figura. Esta diseñado para contestar la pregunta, ""Que clase de mamífero es negro, tiene cola y se llama Mr.Ed ?"

Cuando este fragmento es aplicado, el procedimiento de apareamiento busca una red que coincida perfectamente con el fragmento que ha sido construido. Cuando esto ocurre, el procedimiento liga cada variable al nodo correspondiente de la red apareada. Así, en la red

de la figura, la variable HS1? será ligada al valor de CABALLO para satisfacer la pregunta.



De este modo se ejemplifica como navegando por la red se pueden inferir hechos y relaciones. Para mayor información sobre redes semánticas véase [Findler, 1979].

2.5.3 Razonando Con Marcos

Los humanos poseemos la habilidad de interpretar nuevas situaciones en base al conocimiento obtenido de experiencias en casos similares. Esta habilidad permite que nuestro conocimiento crezca con cada experiencia en lugar de comenzar de condiciones iniciales para cada una de ellas.

Por ejemplo, de acuerdo con la experiencia pasada, la expectativa es que los automóviles tengan ruedas y motor, que requieran combustible y que se muevan. Estos elementos son las características de definición que, asumidas totalmente, constituyen la comprensión de lo que es un coche. Estas son nuestras expectativas respecto a un coche, las cosas que, a menos que exista evidencia de lo contrario, se espera que sean verdaderas para todos los coches.

Los humanos sostenemos grandes colecciones de estructuras de conocimiento, que incluyen éstas expectativas como valores por ausencia (del inglés default values).

Representar conocimiento con marcos permite hacer rápidas inferencias, dado que la información disponible está incompleta. Por ejemplo, los humanos podemos inmediatamente asumir que un objeto tiene un volante de dirección, tan pronto como lo identifiquemos como un coche.

Para modelar este proceso de razonamiento, se escoge un marco para representar la situación predominante en el momento. Puesto que en la mayoría de los casos no se cuenta con el marco exacto para el contexto existente, frecuentemente se busca el que mejor ajuste de acuerdo a la evidencia parcial disponible. Entonces se debe instanciar el marco seleccionado de acuerdo a las condiciones imperantes.

En general el proceso de instanciación asocia a un individuo con una clase. El proceso que construye una instanciación, genera una descripción individual, que resulta de la aplicación de las características individuales a la descripción genérica de la clase.

Una de las dificultades con la representación de marcos, es el problema del establecimiento de los valores por asuencia. No existe nunca un acuerdo general entre diferentes grupos de observadores, en cuanto a las características de algún objeto. Cada visión individual está sesgada por una experiencia personal. Por ejemplo la palabra "coche" puede generar en una persona una imagen de un auto antiguo y clásico, mientras que en otra puede aparecer la imagen de un moderno auto deportivo, además una persona puede estar pensando en un modelo a escala y la otra un auto tamaño natural.

A pesar de estas dificultades, los marcos, permanecen como uno de los mecanismos más poderosos de representación de conocimiento y día a día son más extensamente utilizados. Para mayor información sobre marcos consultar [Wiston, 1987].

2.5.4 Razonando Con Scripts

Para lograr inferencias con guiones, es necesario primeramente seleccionar el guión adecuado. El proceso de seleceión es similar al de los marcos, buscando el que mejor se ajuste a las condiciones existentes.

Una vez que se tiene el guión que modela las condiciones existentes, se pueden utilizar las escenas para inferir la existencia de eventos no observados. Esta indica, que por

ejemplo, en el caso del restaurante (sección 2.5.4) el cliente que llega a comer primeramente tuvo que haber encontrado una mesa disponible, sin embargo; no existen elementos para prever situaciones futuras, por lo que no se puede inferir que pedir de comer.

Resulta útil, también, el observar la secuencia de eventos en un guión como una serie de relaciones causa-efecto que establecen una complejo cadena causa.

Usando esta forma de razonamiento se puede establecer la causa original de un evento observado.

De cualquier forma, los eventos que se han etiquetado como "causa" pueden ser simplemente antecedentes que permitir naciones posteriores. "Causa" implica un posible intento deforzar una acción [Rolston, 1988]. Sin embargo, la selección arbitraria de un antecedente como "causa" pude ser un asunto delicado. Por ejemplo, existen un consenso, general de cuales fueron las causa que originaron La Segunda Guerra Mundial, sin embargo; hoy en día no todos los observadores coinciden en cuales serían las causas que originarían una nueva guerra.

2.5.5 Interprete De Reglas De Producción

El intérprete para un conjunto de reglas de producción pude ser descrito en términos del llamado "ciclo reconoce-actúa", que consiste de los siguientes pasos:

- Aparear los patrones de ejecución de las reglas con las elementos de la memoria de trabajo.
- Resolución de conflicto, que consiste en decidir que regla disparar cuando más de una ha sido apareada con los elementos dela memoria de trabajo.
- Ejecución de la regla, que es el momento en el cual puede ser añadido o borrado algún elemento de la memoria de trabajo, cambiando el estado del mundo para colocarnos de nuevo en el paso 1.

Usualmente en la memoria de trabajo se coloca un elemento de arranque, para comenzar el ciclo descrito. El proceso se detiene cuando todas las reglas han sido apareadas o cuando existe una orden explícita de detenerse.

En el paso 2, el sistema tiene un conjunto de pares, formados por las reglas y la ligas de las variables derivadas del apareamiento de patrones. A estos pares se les conoce como instanciaciones. La resolución de conflicto ocurre cuando el sistema decide cuál de estos pares usar primero, cuando todos están activos, para lo cual sigue una estrategia de resolución de conflictos.

La estrategia de resolución de conflictos tiene una fuerte influencia en el comportamiento general del sistema. Un buen desempeño del sistema depende de su sensibilidad y de su estabilidad. La sensibilidad indica que tan rápido puede responder a los cambios en el ambiente (estada del mundo), reflejados en la memoria de trabajo, y la estabilidad se refiere a la continuidad que el sistema exhibe en su línea de razonamiento [McDermott 82 y Forgy, 1980]. Los mecanismos para solución de conflicto varían de sistema a sistema; sin embargo existen tres muy populares que frecuentemente se utilizan en combinación, y que a continuación se describen:

Refracción: Una regla no debe ser disparada más de una vez por el mismo dato. La forma de lograrlo es descartando del conjunto de instaciaciones conflicto, los pares que hayan sido apareados y disparados antes. En algunos sistemas se encuentran funciones de refresco de la memoria de trabajo, para permitir la ejecución de ciclos.

Recenticidad: Los elementos de la memoria de trabajo, frecuentemente se encuentra etiquetados con marcas de tiempo, de manera que se puede identificar en que ciclo dicho elemento fue incorporado a la memoria de trabajo. Esta estrategia ordena las instanciasiones en función de su recenticidad (la más reciente). Así, las reglas que aparean con los datos más recientes son preferidas a las reglas que aparean con datos más antiguos, de manera que siempre se trata de ir a la "punta del proceso", utilizando solo datos nuevos, mientras la línea de razonamiento no falle, en cuyo caso se requerirá buscar datos más antiguos.

Especificidad: Las instaciciones que ocurren con reglas de mayor especificidad, i.e. reglas con mayor número de condiciones y por lo mismo más difíciles de satisfacer, se prefieren sobre aquellas más generales de menor número de premisas.

Controlar el comportamiento de un sistema de reglas de producción no es un problema trivial. En general existen dos maneras de tratarlo: el control global y el control local. Los regímenes de control global tienden a ser de dominio libre, de manera que la estrategia empleada no necesita del conocimiento del dominio para un desempeño significativo. Los regímenes de control local, tienden a ser dependientes del dominio, lo que significa que reglas especiales (meta-reglas) que usan conocimiento sobre el dominio se requieren para mantener el control. Las técnicas globales se encuentra usualmente "fuertemente codificadas" en el intérprete, y dificultan los cambios que el programador pudiese hacer, mientras que las técnicas locales se encuentran con frecuencia "ligeramente codificadas", de manera que el programador puede con cierta facilidad escribir reglas acerca de las reglas (meta-reglas) para producir efectos particulares. Una meta-regla se puede distinguir de una regla ordinaria, en que su papel es el de "dirigir el razonamiento" requerido para la solución de un problema, y no el de "desarrollar razonamiento" sobre el problema [Jackson, 1986].

2.5.6 Encadenamiento Hacia Adelante Y Hacia Atrás

Al hablar de encadenamiento hacia adelante o hacia atrás, se hace referencia a una forma de resolver conceptualizar los problemas y su solución. Cuando se indica que se utilizar una estrategia de encadenamiento hacia adelante, quiere decir que la forma del problema ofrece una serie de condiciones en que se saben ciertas premisas y a partir de las cuales se podrá "caminar" hacia una conclusión. Ejemplo:

En Suscriptor, un sistema de selección de riesgo para seguros de vida una de sus reglas dice:

SI el solicitante declara haberse practicado un electrocardiograma, cuyo resultado muestra curvas normales, y el solicitante es mayor de 40 años de edad, ENTONCES no existe razón para solicitar un examen médico en busca de algún mal no manifestado.

Es claro que los primeros hechos dan la información necesaria para obtener la conclusión.

En la estrategia de encadenamiento hacia atrás, se parte de que la conclusión es cierta, tratando de probar que el valor de verdad de las premisas apoyan inobjetablemente la conclusión utilizando el mismo ejemplo:

SI no existe razón para solicitar un exámen médico en busca de algún mal no manifestado ENTONCES el solicitante declara haberse practicado un electrocardiograma cuyo resultado muestra curvas normales, y el solicitante es mayor de 40 años de edad.

Ahora se muestra que para que la conclusión sea cierta se debe demostrar que está apoyada por sus premisas.

Vale la pena hacer una distinción entre "encadenamiento" hacia atrás y hacia adelante, y "razonamiento" hacia atrás y hacia adelante. El encadenamiento se refiere a la forma en como las reglas son activadas; esto es, el apareamiento de los elementos de la memoria de trabajo con uno de los lados de la regla y su procesamiento con el otro. El modo de razonamiento describe como el programa ha sido organizado. Y como se complementa la estrategia de solución que el programador ha propuesto. De esta forma es posible instrumentar estrategias pensadas hacia atrás con máquinas de inferencia que caminan hacia adelante (El sistema R1 de DEC así funciona [Jackson, 1986]).

2.5.7 Sistemas De Reacción

En los sistemas de reacción, la parte *SI* especifican las condiciones que deben satisfacerse y la parte *ENTONCES* especifica una acción que se debe realizar. Algunas veces la reacción es con el fin de agregar una nueva afirmación existente; pero también en ocasiones su objetivo es ejecutar algún procedimiento que no implica afirmación alguna. Un sistema de reacción basado en reglas es capaz tomar decisiones en base a datos tomados de una base de datos especificada. Cuando dos o más reglas pueden ejecutarse podría haber un problema, en estos casos los sistemas de reacción adoptan un procedimiento de resolución de conflictos, que le permite determinar que regla se dispara entre todas las que

se pueden accionar, una estratégia a utilizar es el ordenamiento de reglas, que significa que la primera regla que se acciona es la que tiene que dispararse.

2.5.8 La Interfaze Con El Usuario

El usuario de un SE pude jugar diferentes roles en la operación del sistema, tales como:

Attestiguador: En este caso se verifica la veracidad del comportamiento del sistema.
Tutor: El usuario aporta nuevos conocimientos al sistema o modifica el conocimiento ya existente dentro de este.
Alumno: Aquí se busca un rápido desarrollo de la experiencia personal relacionada con el dominio de aplicación del sistema, extrayendo organizadamente conocimientos del mismo.
Cliente: El usuario aplica el sistema a una tarea específica real.

El reconocer los diferentes papeles que el usuario juega en la utilización del sistema, da un contraste más a la programación tradicional.

La interfaze de usuario debe aceptar información del operador y traducirla a una forma aceptable al resto del sistema y viceversa, es decir, aceptar información del sistema y traducirla de manera que pueda ser entendida por el usuario. Idealmente esta interface podría consistir de un sistema de procesamiento de lenguaje natural, que acepta y retorna información esencialmente de la misma forma en que lo hace un experto humano. Ciertamente, aún no existen sistemas que logren este intercambio de información de manera absolutamente natural, sin embargo, la comunicación con el SE debe ser lo más natural posible ya a que a la luz de los hechos, el sistema intenta emular el comportamiento de un experto humano.

Cuando el usuario es el desarrollador del sistema, se requiere de interfaces más complejas y poderosas. A diferencia de los sistemas tradicionales, el proceso de depuración rebasa los límites de fallas de procedimiento y asignaciones equívocas. Cuando se trabaja con bases de conocimiento y SEs el número de posibles problemas crece, por lo que es necesario tener herramientas que permitan hacer rastreo de inferencias, detección de

inconsistencias, visualización gráfica de la forma de la base de conocimientos y editores orientados al esquema de representación con el que la herramienta trabaje.

2.5.9 Mecanismos De Explicación

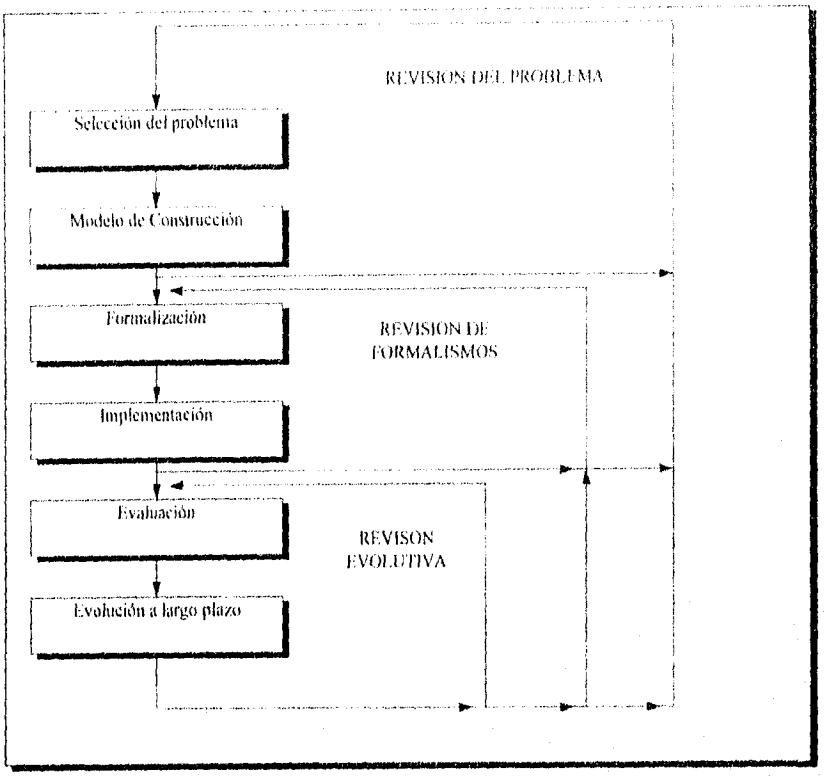
Además de obtener una conclusión cuando se encara un problema complejo, un experto es capaz de explicar el razonamiento que lo condujo a la conclusión. Un SE debe ser diseñado para proveer una facilidad similar, ya que la credibilidad del sistema depender finalmente de la capacidad que tenga de explicar su razonamiento. Esta característica es una distinción importante de los sistemas computacionales tradicionales.

Básicamente, la explicación consiste en la identificación de los pasos en el proceso de razonamiento y su justificación. El sistema debe tener capacidad para comunicar esta información, que en esencia se trata de un problema de procesamiento natural. Si bien es cierto, la construcción de un SE debe dar como resultado un sistema cuyo comportamiento sea similar al de un experto humano; Sin embargo en la práctica se puede sugerir un mecanismo de explicación ciertamente 'amigable' pero todavía lejos de operar como un sistema de procesamiento de lenguaje natural.

En cualquier caso el sistema debe tener acceso a un registro del conocimiento, que fue utilizado para obtener la solución y traducirlo de una manera inteligible para el usuario. En un sistema ideal, tal como lo hace un humano, el módulo de explicación confeccionar su diálogo de manera que corresponda al nivel de conocimiento del usuario. Sin embargo, en la mayoría de los SEs, la facilidad de explicación se reduce a enunciar la serie de reglas que le condujeron a la solución.

2.6 CICLO DE VIDA DE UN SISTEMA EXPERTO

Para el desarrollo de un SE se sigue un modelo como el que se muestra en la siguiente figura que es una versión iterativa del modelo tradicional de ciclos de vida. El modelo de ciclo de vida para un SE se basa en el reconocimiento de la naturaleza evolutiva del desarrollo de software.



Existen varias características únicas del modelo del SE:

- El cliente (y el experto) están involucrados de principio a fin en el proceso completo. Esto está en oposición con el sistema tradicional, en que el usuario especifica los requerimientos y luego se marcha a esperar el producto final.
- Demostraciones frecuentes de trabajos hasta la fecha, son estimuladas. Estas demostraciones posibilitan al cliente y al experto para visualizar la funcionalidad del sistema completo y solicitar cambios.
- El cambio se visualiza como saludable de hecho es el concepto central. El cambio es motivado especialmente durante la fase de prototipo. La intención es identificar los cambios durante la fase en que es más fácil de implementar.

En las siguientes secciones se describe detalladamente las fases del ciclo de vida de un Sistema Experto.

2.6.1 Selección De Problemas

El proceso de selección de problemas, que es una de las fases más críticas de un proyecto de construcción de un SE, se puede descomponer en los siguientes pasos:

- ***Investigación del Problema:*** El primer paso en el proceso de selección del problema es tener una sesión de tormenta de ideas para compilar una lista de problemas que pudiesen ser considerados.
- ***Selección del candidato:*** El propósito de la fase de selección de problemas es el de reducir la lista a las aplicaciones que recibirán consideración seria. Para seleccionar a aquellos candidatos, cada uno de los elementos en la lista original se evalúa con relación a un conjunto de criterios de filtración. Solamente las aplicaciones que cumplen virtualmente todos los criterio de filtración avanzan a la lista de candidatos. Los siguientes son criterios de filtración basicos:
    - ¿Requiere la tarea el empleo de conocimiento experto?
    - ¿Es escasa la pericia?
    - ¿Están disponibles los expertos quienes saben como realizar la tarea?
    - ¿Existen razones para creer que la solución algoritmica tradicional sería difícil de implementar?
    - ¿Requiere la tarea una cantidad razonable de conocimientos de juicio o enfrenta algún grado de incertidumbre?
    - ¿Requiere la tarea habilidades verbales primeramente?
    - ¿Es muy valiosa una resolución del problema para la organización, es decir, vale la pena resolverlo?
    - ¿Una solución que sea valiosa para el día de hoy, permanecera útil durante los próximos años?
    - ¿Es aceptable para el sistema que ocasionalmente falle en encontrar una solución; Esta bien que produzca una respuesta subóptima por lo menos en algunos casos?
    - ¿Hay disponibilidad de tiempo para construir el sistema (6 meses al menos)?

El propósito de esta actividad es filtrar rápidamente los problemas que obviamente fallan en cumplir los criterios. Cada candidato es luego analizado con más detalle.

- ***Análisis del candidato:*** En este punto tenemos cuatro pasos que son:
    - *Aplicabilidad del dominio*: Este paso consiste de un análisis más detallado de la apropiabilidad del dominio en cuanto a la aplicación de SE.
    - *Disponibilidad experta*: La importacia de este punto radica en el principio de dicta que es imperativo que un experto apropiado este disponible si un proyecto de SE va a tener éxito
    - *Alcance del problema*: Seleccionar un proyecto de un SE que parezca muy pequeño. El sistema propuesto debe tratar un problema bien delimitado en un domino reducido
    - Analisis beneficio/Costo: Aunque es muy difícil determinar el beneficio/costo de un proyecto de SE, ya que los resultados son a largo plazo, es importante considerar en ciertos casos este punto aunque sea con relatividad.
- ***Selección final del Candidato:*** En base a los puntos anteriores se hace una selección óptima.

2.6.2 Modelo De Construcción

Una vez que se haya seleccionado el dominio del problema, la siguiente tarea es construir un prototipo que represente una pequeña parte del sistema final. Típicamente emprende 5 a 10 casos de prueba y requerirá desde unas pocas semanas hasta unos pocos meses para completarlo, dependiendo del alcance y dificultad del problema. Los pasos en este proceso son:

- Adquisición del conocimiento inicial
- Aproximación al problema básico
- Modelo de consulta general
- Selección del paradigma de inferencia
- Selección de la representación de conocimientos
- Selección de herramientas
- Implementación del prototipo
- Prueba del prototipo

- Demostración del prototipo
- Revisión del proyecto

El propósito primario de un prototipo de demostración es aprender más acerca del mundo, siendo la forma de implementar un prototipo, expresar su mejor entendimiento, aun si es erróneo, luego criticarlo y revisarlo. Además se deben conservar aquellas partes del prototipo cuyo nivel de calidad en la implementación sea semejante al esperado en el sistema final.

Aún si se descarta el prototipo completo, puede como mínimo servir de guía inmediata para una nueva implementación.

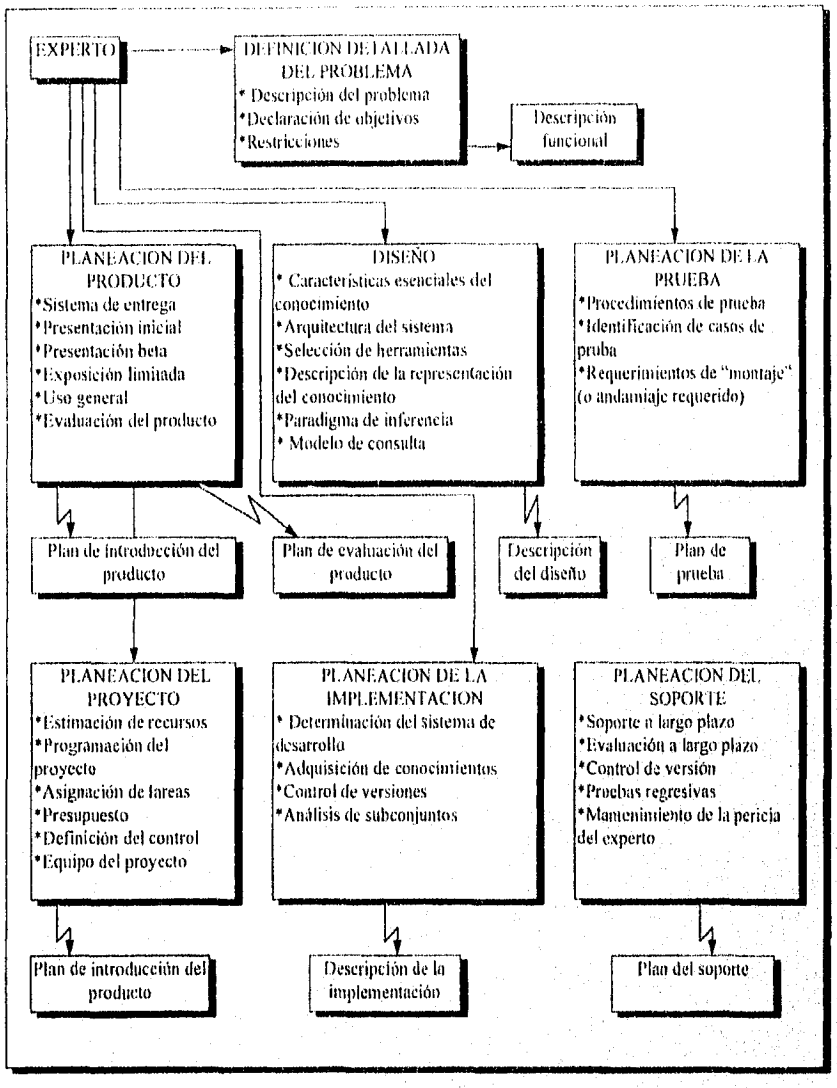
2.6.3 Formalización

Una vez verificado el prototipo, se lleva a cabo la fase de formalización. Los propósitos principales esta fase son:

- Capturar y registrar la comprensión clave que se desarrolló durante la fase de prototipo
- Forzar a que la planeación se haga, antes de comenzar la implementación completa
- Registrar las decisiones respecto a las estrategias de implementación .
- Ofrecer visibilidad de los puntos de verificación para posibilitar a la administración del proyecto y al usuario, a que se involucren en el proyecto.
- Ofrecer visibilidad a todo tipo de pensamiento corriente para posibilitar que más gente contribuya al proyecto.
- Posibilitar el desarrollo concurrente de pruebas, el despliegue y el soporte de medios a largo plazo.

La formalización es tan importante en el desarrollo de un SE como lo es en el desarrollo de cualquier otro gran sistema complejo de software. De hecho puede ser aún más importante porque existe la necesidad de captar una comprensión un problema que inicialmente no se había entendido bien, y porque siempre existe el peligro de deteriolarlo. El lema en la construcción de un SE no es evitar el formalismo sino por el contrario lo que se trata es de evitar el prematuro formalismo.

A continuación en la figura se muestra el proceso de formalización de un SE.



- ***Análisis detallado del problema:*** El primer paso de formalización, *definición detallada del problema*, es desarrollar una descripción funcional que defina, tan claro como sea posible, el problema que se va a resolver, los objetivos a lograr y lo más importante cualquiera de las restricciones que sirvan para limitar el alcance del proyecto y las expectativas del usuario.
- ***Diseño:*** En el diseño más tradicional, el software se descompone hasta un nivel en que las funciones específicas se identifican y se asignan jerárquicamente en módulos, se especifican interfaces, se desarrollan estructuras de control al igual que diccionarios de datos. En el diseño de SE's el dominio no está lo suficientemente bien comprendido para producir este nivel de detalle, por el contrario el diseño de SE's se centra en la identificación de la representación correcta del conocimiento, el paradigma de inferencia y la arquitectura del sistema. La descomposición requerida del sistema se lleva a cabo (iterativamente) durante la fase de implementación.
- ***Planeación del Proyecto:*** Presenta un estimativo de los recursos que se requerirán para completar el proyecto junto con el presupuesto que describe la forma como se asignaran los recursos. También incluye una declaración de la propuesta de de programación del proyecto que muestra las tareas y sus interdependencias. Cada una de estas tareas se asigna a un miembro del equipo del proyecto propuesto. El plan de proyecto debe establecer específicamente qué tanto tiempo del experto se espera durante cada fase del desarrollo.
- ***Planeamiento de la Prueba:*** Hace énfasis en la definición de los elementos requeridos para verificar la validez del sistema después de que éste se haya desarrollado. Esto incluye la identificación de los casos de prueba y procedimientos para ejecutarlos. La selección de los casos de prueba debe centrarse en la identificación de un conjunto de casos nominales como también de casos que fuercen al sistema mediante presión a trabajar en las condiciones límites.
- ***Planeación del Producto:*** Trata con las actividades necesarias para producir un producto final a partir de una implementación inicial. El *plan de presentación del producto* describe la forma como el producto se ha de presentar a los usuarios. La presentación inicial normalmente es sólo para algunos usuarios más allegados y conocedores. Con frecuencia la sola diferencia entre la actividad de prueba y la presentación inicial del

producto es que el sistema se usa para atacar problemas reales. Los resultados de la presentación inicial se emplean para revisar el sistema y después de que se haya desarrollado un razonable nivel de confianza, se entrega el sistema para unos cuantos sitios beta. Después de que se haya establecido un aceptable nivel de confianza, se entrega el sistema para uso general. El plan de evaluación del producto describe como se evaluará el sistema durante su uso rutinario. Esta evaluación pretende establecer la utilidad del sistema. Se usa para orientar la modificación del proyecto a largo plazo y posiblemente para influir en decisiones con relación a proyectos de desarrollo futuro.

- ***Planeación del soporte:*** Aunque el sistema continuará cambiando durante este período, es necesario dar soporte en la marcha del sistema que está en uso, para resolver problemas reales. Aquí se produce un plan de soporte que identifica los elementos que se requerirán para dicho soporte. Este plan describe como se conducirá la administración de la configuración y como se ofrecerá el soporte a largo plazo.
- ***Planeación de la implementación:*** Produce un documento de descripción de la implementación que señala como se conducirá el desarrollo del sistema. También describe el sistema que se empleará para el desarrollo, el proceso para control de las iteraciones y la segmentación del proceso de desarrollo en subconjuntos.

2.6.4 Implementación

Los pasos en la fase de implementación son:

- Revisión del prototipo
- Desarrollo de la infraestructura del sistema
- Adquisición de conocimientos esenciales
- Desarrollo de software auxiliar
- Integración interna
- Verificación interna

El proceso de revisión del prototipo consta de los siguientes pasos:

- Revisión de las decisiones de representación e inferencia
- Revisión del nivel de representación de los componentes
- Establecimiento de la partición de la base de conocimientos

- Verificación

Una técnica efectiva para enfrentar la complejidad es descomponer el sistema en pequeñas partes manejables. Los trozos de conocimientos deben ser agrupados de tal manera que maximice la cohesión y minimice el acoplamiento.

A continuación de la revisión del prototipo, comienza la parte más larga del desarrollo. Esta incluye la implementación de la infraestructura básica para el sistema, la base de conocimientos esenciales y cualquier software auxiliarque se requiera. Estas actividades generalmente se pueden ejecutar en paralelo. La base de conocimientos esenciales consta del conocimiento requerido para completar los casos de prueba identificados en el plan de prueba. El nuevo conocimiento adquirido se debe de integrar con el antiguo conocimiento. Los dos mayores desafíos en este proceso lo constituyen el determinar si el conocimiento a ser agregado ya está presente en la base de conocimientos en forma ligeramente diferente o en forma más general y asegurar que el nuevo conocimiento sea consistente con el conocimiento existente.

El software auxiliar se desarrollan en paralelo con la infraestructura del sistema y por el empleo de técnicas tradicionales de desarrollo de software.

2.6.5 Evaluación

Para muchos de los dominios atacados por un SE, es imposible identificar una respuesta que sea "absolutamente correcta" para cualquier problema dado. En la evaluación de SE la respuesta correcta se toma como aquella dada por la persona experta para la misma pregunta. Las respuestas de un SE se deben evaluar con relación al experto del dominio y luego con relación a las respuestas dadas por un grupo de expertos.

2.6.6 Evolución A Largo Plazo

Como cualquier sistema de software grande, un SE continúa evolucionando de principio a fin de su vida. Varias clases de evolución están involucradas.:

- Incremento de la funcionalidad en general
- Correcciones, particularmente a la base de conocimientos

- Adiciones a la base de conocimientos para hacer una base más completa
- Expansión del dominio
- Revisiones requeridas por las modificaciones externas

El aspecto más significativo de la evolución a largo plazo resulta del efecto que tiene sobre un experto el desarrollo de un SE. El proceso de evolución de un SE a largo plazo viene luego a ser un crecimiento combinado del experto y del sistema.

2.7 ARQUITECTURAS DE SISTEMAS EXPERTOS

Los SE's emplean una amplia variedad de arquitecturas específicas en sus sistemas, principalmente porque una arquitectura es más aplicable que otra cuando se considera una aplicación dada. A continuación hablaremos de algunas de las más usadas.

2.7.1 Modelos De Pizarrón.

Quizá una de las arquitecturas más interesante y con más probabilidades de establecer la vanguardia en construcción de SEs, sea el modelo de pizarrón. Por su forma es el modelo que mejor se adapta a sistemas computarizados de multiproceso y de proceso distribuido. El modelo de pizarrón, fue desarrollado para el proyecto HEARSAY-II que consiste en un sistema de reconocimiento de voz.

La idea que se encuentra detrás del modelo de pizarrón, puede ser explicada como sigue:

Se tiene a un grupo de expertos, cada uno de los cuales está altamente especializado en un campo específico. Se trata de coordinar el conocimiento de estos expertos para resolver un problema difícil. Los expertos no hablan entre sí, pero para ayudar a resolver el problema, están de acuerdo en interactuar através de un coordinador o calendarizador y en leer y escribir en un pizarrón.

De ésta forma se tiene a los expertos en un cuarto con un pizarrón muy grande en donde se escriben las indicaciones iniciales del problema. Los expertos leen éstas y empiezan a resolver el problema. Cada vez que un experto tiene una idea relevante o una hipótesis importante, escribe en el pizarrón para que todos la vean. Esto ayuda a los otros expertos, pues les da importantes claves basadas en conocimiento externo a su dominio.

Eventualmente uno de los expertos resolverá el problema y escribirá la solución final en el pizarrón.

Cuando este modelo se usa en un contexto de SE's, se le denomina a cada experto participante fuente de conocimiento (del inglés"Knowledge Source"). Cada una de éstas puede estar estructurada por diferentes esquemas de representación y métodos de inferencia.

Desde un punto de vista general, el modelo de pizarrón tiene que lidiar con tres distintos problemas que aparecen, cuando el tamaño de la base de conocimiento crece:

- El sistema se hace difícil de entender, dado que existen muchas reglas, hechos, etc.
- Se requiere integrar diferentes tipos de conocimiento, diferentes esquemas de representación y diferentes métodos de razonamiento.
- El tiempo de respuesta se empieza a deteriorar conforma el número de operaciones (computaciones) se incrementa.

Una arquitectura de pizarrón está constituida de tres partes fundamentales:

- Una base de datos global (el pizarrón)
- Fuentes de conocimiento independientes que tienen acceso al pizarrón.
- Un calendarizador para controlar la actividad de las fuentes de conocimiento.

El pizarrón se distingue más como una arquitectura para solución distribuida de problemas, que como una forma más de representación de conocimiento [Parsey, Chignell, 1988], lo que indica que las tres ventajas distintivas del pizarrón son:

- Permite organizar el conocimiento en forma modular.
- Permite integrar fácilmente diferentes métodos de representación de conocimiento.
- Pude ser ejecutado en ambientes de computación distribuidos para mayor eficiencia.

En máquinas de un solo procesador la tercera ventaja no existe, por el contrario, el problema de calendarización se hace más complejo, al coordinar en una sola unidad un variado número de esquemas de representación y mecanismos de inferencia. No obstante es

indudable que la mayoría de los problemas del mundo real presentan esquemas híbridos que con una sola y exclusiva técnica restan a la solución naturalidad y eficiencia [Bañares,1989].

2.7.2 Modelos De Clasificación.

Se ha aplicado ampliamente para la configuración de reglas, en especial en tareas de diagnóstico y de interpretación. Este modelo se emplea para organizar el razonamiento (a partir de las observaciones hasta las conclusiones) que se basa en clasificación: la selección de una conclusión a partir de una lista preespecifica de conclusiones. Típicamente se implementa como un sistema de producción modificado con una base de conocimientos, mecanismos de control y memoria de trabajo.

El segmento de memoria de trabajo de este modelo es más sencillo que el correspodiente al típico sistema de producción. El modelo de clasificación emplea el área de memoria global para almacenar proposiciones sencillas (como declaraciones que son falsas o verdaderas) y de esta manera implementa una forma de la lógica proposicional más sencilla.

El proceso de razonamiento en un modelo de clasificación, conceptualmente, usa una forma simple de inferencia fundamentada en la representación de las implicaciones mediante reglas y la aplicación de *modus ponens*[*] basada en la evidencia recogida.

Este proceso comienza en las observaciones iniciales y aplica reglas para desarrollar nuevos hallazgoso hipótesis intermedias basadas en las observaciones iniciales. Luego procede a través de varios niveles de análisis intermedios hasta que se logre una conclusión final.

En un modelo de clasificación se usan dos clases generales de reglas:

- Evidencia-a-conclusión. Estas reglas se usan para enumerar las conclusiones que se señalan por la evidencia y para conducir el proceso de razonamiento descrito anteriormente. Como ejemplo de esta clase de regla está:

[*] *modus ponens*: se expresa simbólicamente como: [P1^(P1→P2)]→P2 o en español "si P1 es verdadera y P1 siendo cierta implica que P2 sea verdadera, entonces P2 será verdadera".

SI el elemento incandescente en una estufa luce rojo

ENTONCES concluya que el elemento está muy caliente.

- Conclusión-a-evidencia. Esta clase de regla describe la evidencia que debe estar presente si existe una condición dada. Estas reglas se emplean ara describir evidencias y buscar confirmar una hipótesis sospechada. Como ejemplo tenemos:

  SI las luces se apagaron porque en la compañía de servicios se cayó la energía

  ENTONCES Las luces de los vecinos deben estar apagadas

Dentro de estas clases generales de reglas se emplean varias clases específicas de reglas:

- Hallazgo-a-hallazgo. Esta clase de regla describe el hecho que un hallazgo se pueda inferir a partir de otro. Por ejemplo: una regla que diga que una vez que el lector haya concluído la batería del automóvil está muerta, entoces el lector también puede concluir que es verdadero el enunciado "las luces delanteras no encederan". Esto constituye un ejemplo de la aplicación de un principio general. Se debe evitar hacer preguntas para recuperación de información que se podría deducir de la información ya disponible.
- Hallazgo-a-hipótesis. Esta clase de reglase emplea para generar hipótesis intermedias a partir de las observaciones o de los hallazgos. Por ejemplo, si el circuito eléctrico se ha saltado, entonces concluya que hay un problema eléctrico.
- Hipótesis-a-hipótesis: Esta clase de regla se usa para pasar de hipótesis intermedias a otras hipótesis o conclusiones finales. Por ejemplo:

  SI si toda la energía eléctrica se corta repentinamente y toda la energía del vecindario está caída

  ENTONCES concluya que en la compañía de servicios se ha caído la energía.

CAPITULO 3
SISTEMAS EXPERTOS EN MEXICO

3.1 INTRODUCCION

En el presente capítulo se da una idea de la situación los Sistemas expertos en nuestro país. La información que aquí se ofrece fue recogida de profesionistas y académicos que han trabajado durante un tiempo razonable en proyectos sobre IA y SE's en México como en el extranjero. Para reconocer a las personas adecuadas para obtener de ellas la información, se tomaron en cuenta varias elementos de selección.

El objetivo de las entrevistas fue el obtener una descripción real de la situación que priva en los diferentes sectores de nuestro país, referente a la contracción, comercialización, y conocimiento general de los Sistemas Basados en el Conocimiento. La forma en la cual se pudo contactar a los investigadores se hizo de forma directa; através de consulta en revistas especializadas se conoció su localización actual de algunos de ellos, los cuales a su vez nos recomendaban más gente relacionada con el desarrollo de esta tecnología.

Las entrevistas se realizaron con gente que desarrolla esta tecnología en diferentes ciudades del país (Monterrey, Veracruz, Puebla y D.F.) Así como en diversas universidades, UNAM, ULSA, ITAM, IBEROAMERICANA y ITSEM. La gente entrevistada pertenece a la Sociedad Méxicana de Inteligencia Artificial.

El capítulo se inicia con una breve historia de la IA y los Sistemas Expertos en el mundo, posteriormente nos ubicamos en el surgimiento del desarrollo de esta tecnología en nuestro país, luego se hace una descripción de la situación actual, donde se tratan temas como la problemática que ha enfrentado y que enfrentará la tecnología de SE's, las alternativas que existen, que se esta haciendo y cuales son los sectores mas interesados. Posteriormente se muestra un perfil académico con las expectativas de los especialistas en el área. Se hace mención de algunas universidades que cuentan dentro de alguna de sus carreras con materias referentes a I.A. o a SE's.

Además se reproducen las opiniones de los expertos en cuanto a lo que les espera, en los próximos años a los SE's en México.

3.2 EVOLUCION DE LA I.A. Y LOS SISTEMAS EXPERTOS

En la evolución de la Inteligencia Artificial (I.A.) se, presentaron dos fuerzas importantes entre 1930 y 1940, las cuales se presentaron mediante la lógica matemática, siendo el primer surgimiento a finales del siglo.

La lógica matemática ha continuado siendo una área activa, en la investigación de la I.A, pero desde mucho antes la existencia de la formulación matemática del razonamiento lógico da forma a la concepción que se tienen generalmente de la relación entre ordenador e inteligencia.

Hablando de los ordenadores, los primeros que existieron fueron las calculadoras numéricas, que surgen sin ninguna inteligencia real, pero incluso antes de estas máquinas que fueron diseñadas por Church y Turing, ellos observaron que los números, eran un aspecto no esencial en la comprensión.

Turing el padre de la I.A. defendió la posibilidad de mecanismos computacionales que se comportaban de manera que se consideraban inteligentes. La lógica de resolución de problemas y la computación eran corrientes intelectuales.

Se piensa que la naturaleza de la mente puede ser adquirida estudiando la lógica de los programas que se ejecutan.

En 1950 fué el año, en que se empezó las investigaciones, con los cuales se han creado docenas de técnicas de programación que soportan de un modo u otro algún comportamiento inteligente, los sistemas experimentados de I.A. han sido generados en forma inteligente y entusiasta en la industria; de una forma que resuelve programas de problemas complicados en disciplinas como Química, Biología, Geología, Ingeniería y medicina que habitualmente precisan expertos humanos, es por ello que promueven dispositivos que manipulan robots para dar solución a algunas tareas usuales, repetitivas y relacionadas.

Con esto queremos decir que los programas de la I.A. juegan un papel importante en el entorno informático dentro de la vida humana.

Durante la Segunda Guerra Mundial, los americanos y los británicos fueron los primeros en utilizar los ordenadores para tareas complejas, como son los cálculos numéricos e interpretación de claves, actividades que correspondían a realizar a la inteligencia humana.

El brillante matemático Alan Turing ayudo a diseñar uno de los primeros ordenadores que se llegaron a construir, de ahí que surgió, la idea de crear una máquina que usara un sistema de códigos binarios, que realizara cualquier operación matemática (algorítmica). En 1950 Turing escribió un artículo en el que sugería la posibilidad de que las máquinas pudieran pensar, y fue como surgió la idea de la I.A., es por ello que se considera el padre de la I.A. Para poder demostrar que su artículo funcionara realizó una prueba que el denomino "PRUEBA TURING" donde intenta determinar el sexo de una persona haciéndole preguntas a través de un ordenador y debe tratar de determinar si el que responde, es una persona o un ordenador.

En la misma época que Turing, otros científicos empezaron a estudiar la posibilidad de realizar máquinas inteligentes tales el caso de:

Warren Mc.Cullouh, se inclinó por el desarrollo de un mayor entendimiento del funcionamiento del cerebro humano, realizó una comparación entre ordenador y cerebro, lo cual contribuyo a la I.A.

Claude Shannon, gran matemático propuso un sistema numérico binario, usado en los ordenadores digitales con el fin de describir los circuitos eléctricos de computación. Shannon sospechaba que si el algebra boolena, servía de representación del pensamiento humano, se podía usar para describir los circuitos eléctricos, y quizá estos podían ser utilizados para describir, el pensamiento.[1]

La revolución de la Inteligencia Artificial comenzó en el verano de 1956, en un pequeño pueblo de New Hampshire, sede del College en Hanover. En este lugar se reunieron alrededor de una docena de científicos representantes de diferentes disciplinas, tales como:

---

[1] Mishkoff, Henry C., A fondo: Inteligencia Artificial, Ediciones Anaya Multimedia, Madrid 1988, p. 39

matemáticas, neurología, psicología e ingeniería eléctrica, entre otras. El común denominador entre los asistentes, era que todos estaban utilizando la computadora como la herramienta más poderosa en sus investigaciones. Entre otros asistentes a la conferencia se encontraban [Mishkof, 1986]:

John McCarthy: (Organizador) Profesor de matemáticas en Dartmouth College, inventor de LISP y profesor de Ciencias Computacionales en la Universidad de Standford.

Marvin Minsky: (Organizador) Fué, investigador asociado en matemáticas y neurología en la Universidad de Harvard. Fundador del laboratorio de IA en el MIT, presidente de la AAAI (American Asosiation of Artificial Inteligence) y ganador del "Turing Award."

Nathaniel Rochester: (Organizador) En aquellos años Gerente de Investigación en Información de IBM, especialmente interesado en máquinas que exhibieran comportamiento inteligente en la solución de problemas.

Clude Shannon: Representante de Bell Labs. y poseedor de una firme reputación en el area de las Ciencias de la Información.

Allen Newell y Herbert Simon: Presentaron en la conferencia el primer trabajo de IA (el teorista lógico), aún cuando no se conocía como tal. Simon es poseedor de una sólida reputación en ciencias políticas, administración de empresa, psicología, ciencias computacionales y economía en donde fue galardonado con el premio Nobel en 1978. Newell es egresado del programa de matemáticas de la Universidad de Princeton. Su interés principal ha sido el modelamiento del comportamiento humano en computadora y el estudio del comportamiento organizacional.

Oliver Selfridge: Egresado de MIT participó como asistente de Norbert Wiener en la preparación del trabajo de Cibernética en 1948. Fue el primero en realizar un programa de reconocimiento de patrones.

A partir de la conferencia de Dartmouth se dio el consenso para que los científicos ubicaran sus interes de investigación bajo un mismo campo del conocimiento, que desde entonces se conocería como Inteligencia Artificial.
El concepto de Sistemas Expertos fue creado por primera vez en el Congreso Mundial de la I.A. por Feigenbaum. En el desarrollo de los sistemas expertos se consideran cuatro etapas que son las siguientes:

1) Etapa de invención. Se ubica entre 1965 y 1970, existía un poder limitado de estrategias de resolución general de problemas y lleva a muchos investigadores a trabajar para resolver problemas estrechamente definidos. Los sistemas Dendral y Macsyma pertenecen a esta etapa.

2) Etapa de prototipos. Los investigadores están concientes de que la estructura de estos sistemas se basaba en el conocimiento y comienzan a desarrollar teorías de representación y proceso del conocimiento. El período de esta etapa se encuentra entre 1970 y 1977, los sistemas que pertenecen son Caswet, Mycin, Heasay II y Prospector.

3) Etapa de experimentación. Comprendida entre 1977 y 1981, en dicha etapa los conceptos de los prototipos se aplican a distintas áreas, retoma ideas de los sistemas de las etapas anteriores. Se manifiestan los primeros intentos de formalización del proceso de construcción de sistemas expertos, surgiendo los sistemas Metadrenal derivado del Drendal, Teresias, Emycin, Rita y Rosie derivados de Mycin y Kas derivado del Prospector.

4) Etapa de industrialización. comienza con el surgimiento de empresas dedicadas al diseño de sistemas expertos. Comprende los sistemas R1 y Drillin Advisor.

En las etapas de evolución de los sistemas expertos se mencionaron los correspondientes a cada una. Haremos una breve descripción de los existentes, el desarrollo de cada uno es

diferente, ya que algunos son productos comerciales en funcionamiento, mientras que otros solamente se desarrollaron a nivel laboratorio:

a) Drendal. Permite decidir la estructura molecular de los compuestos químicos a partir de una información primaria procedente de ensayos.

b) Macsyma. Realiza cálculos matemáticos, tales como, ecuaciones diferenciales, integración de funciones, cálculo de matrices, etc.

c) Caswet. Sistema experto para consulta de diagnóstico y tratamiento de glaucoma.

d) Mycin. Sistema enfocado a diagnóstico y tratamiento de enfermedades infecciosas de la sangre.

e) Prospector. Contiene una base de conocimientos acerca de geología que permitiera razonar sobre las posibilidades de existencia de yacimientos minerales partiendo de información descriptiva de la zona.

f) Expert. Se ha utilizado en la construcción de modelos de consulta en oftalmología, endocrinología y reumatología.

g) Metadrenal. Retoma a Drendal y le añade conocimiento para proponer y seleccionar reglas de fragmentación de estructuras orgánicas.

h) Teresias. Ayuda a construir grandes bases de conocimiento, facilita la transferencia de experiencia de los humanos a la base de conocimientos por medio del diálogo.

i) Emycin. Diagnostica enfermedades pulmonares.

j) Rosie. Proporciona un sistema de programas de propósito general para la construcción de sistemas expertos.

k) Drilling Advisor. Sistema experto asesor en la interpretación y operación de sonidos.

l) Airplan. Planificación de lanzamientos y despegues, recuperación de aviones sobre una base de portaaviones.

m) Delta. Experto en recuperación de locomotoras eléctricas y diesel.

3.3 HISTORIA EN MEXICO DE LA IA Y LOS SISTEMAS EXPERTOS

Para la realización de esta sección se consultaron las siguientes fuentes:

- **Cristian Lemaitre (LANIA, JALAPA Ver.)**
- **Artículos periodísticos (INEGI)**
- **Memorias de la SMIA**

Históricamente queda demostrado que la IA recibe su gran herencia de la Cibernética. Es indudable que los trabajos realizados por Norbert Wiener y Arturo Rasemblueth en México en 1951 influyeron en los académicos mexicanos y en sus alumnos, algunos de los cuales participarían en la fundación de la comunidad de IA en nuestro país.

Lo que sucedía en la incipiente comunidad IA, en los EUA, en 1958 de alguna manera se repite en México en la década de los sesentas.

La primera computadora en México se instala en el centro de cálculo de la UNAM en 1960, bajo la dirección del Ing. Beltrán. A partir de este momento el Ing. Beltrán se dedica a promover la computación electrónica, para lo cual invita a varios investigadores de nivel internacional, a dar pláticas, conferencias y coloquios sobre tecnologías de punta. Hacia el año 62 y 63 organiza un coloquio sobre Lisp al cual asiste el doctor McCarthy y el profesor

Macintosh, quien posteriormente se establecería en México. Sus trabajos involucran a físicos y químicos de la UNAM y el IPN. Su línea de trabajo se basaba en programación en Lisp. Uno de sus alumnos fue el Doctor Adolfo Guzmán Arenas, quien en ese tiempo hacía su tesis de licenciatura sobre un lenguaje de programación que se llamaba "Comlab".

Después de presentar su tesis para la ESIME, Guzmán se va al MITen Boston USA. La tesis de doctorado de Guzmán en MIT sobre visión fué un trabajo que se consideró clásico por mucho tiempo y además a nivel internacional fue el primer trabajo de computación que se dio a conocer de un mexicano.

El Ing. Alejandro Medina estuvo haciendo estudios de doctorado en USA, hasta el comienzo de la 2a. Guerra Mundia. Según nos cuenta el Dr. Lemaitre, el caracter y la curiosidad del Ing. Medina era comparable a la de Norbert Wiener y al propio VonNeumann.

En México dio cursos en la Facultad de Ciencias de la UNAM, sobre cibernética e Información y control, donde se tocaban temas de teoría de control, física, investigación de operaciones, teoría de autómatas e IA. El Ing. Medina dirigía un grupo de investigación del centro de estudios nucleares, que en realidad era un laboratorio de cibernética. En torno al Ing. Medina se formó la primera generación de estudiantes involucrados en la IA.Un dato curioso es que la actualidad con la que el Ing. Medina se desenvolvía hizo que libros de reciente publicación como el "Machine Intelligence" de la Universidad de Edínburgo, llegaran a sus alumnos con toda oportunidad. La diversidad de los temas que manejaba en su grupo de investigación, hizo que se subdividieran sus alumnos en grupos de diferentes temas. Este grupo de alumnos sale del centro de estudios nucleares hacia la Facultad de Ciencias, donde a la muerte del Ing. Medina en 1970, queda acargo la maestra Gertrudiz Curts de Lara. Ella estuvo dando cursos de cibernética en la Facultad de Ing. y de ciencias, donde se trataban temas en torno a las neurociencias. Entre sus alumnos se encuentra el Doctor Francisco Cervantez quien ahora se dedica a redes neuronales en laboratorio de Fisiología celular de la UNAM. El doctor Lemairtre y Armando Hinich son alumnos del Ing.Medina.

En 1971 el CIMAS bajo la dirección de Fernando Imrriaga, contrata al Ph.D. en matemáticas, Bob Yates de la Universidad de Stanford, uno de los centros pioneros en la IA

(Standford ResearchInstitute). Ahí trabajó en el proyecto de Cordel Green sobre un demostrador de teoremas. Bob Yates dió cursos sobre demostración automática de teoremas (Resolución de Robinson) y es el primero en dar un curso formal en IA, en 1972, en la Facultad de Ciencias, con base en el libro de Nilsson [Nilsson, 1971]. Desde este año existe un curso sistemático en IA en la facultad de Ciencias y cuyos profesores han sido Bob Yates, Armando Hinich y Cristian Lemaitre.

Desde entonces el interés por la IA ha crecido tanto en el ámbito académico, como en el comercial, lo cual queda demostrado por el creciente número de trabajos, tesis y artículos que se han publicado en los últimos años, algunos de los cuales han sido recopilados en las Memorias de la reuniones Nacionales de la SMIA.

Afortunadamente existen un sin número de acontecimientos en la historia de la IA en México, y en los siguientes párrafos han sido registrados algunos de ellos.

En 1972 en el CIMAS se desarrolló un intérprete de Lisp por Mario Magini y Raymundo Segovia.

Desde su nacimiento como disciplina de investigación, la IA mostró la posibilidad de obtener grandes logros. Para 1973 los proyectos de IA en todo el mundo se vieron escasos de fondos, dado que no se habían obtenido resultados prometedores. La credibilidad en los proyectos de IA era muy baja. Paradójicamentees en esta época es cuando se construyen los SE's MYCIN y PROSPECTOR que son el parteaguas entre las aplicaciones académicas y las aplicaciones comerciales. En México consecuentemente, el apoyo a los proyectos de investigación también se debilitó. Fueron los esfuerzos individuales de algunos entusiastas los que mantuvieron el interés de la comunidad científica en la IA.

En 1975 después de trabajar en el Centro científico de IBM, ingresa al IMAS de la UNAM, Adolfo Guzmán. Para 1980, el propio DR. Guzmán ya había desarrollado su máquina jerárquica (máquina Lisp). En este mismo instituto estuvo colaborando Cristina Loyo.

En el escenario mexicano, resulta interminable la lista de personajes importantes como el Dr. Antonio Sánchez Aguilar,especialista en redes neuronales, el Dr. José Negrete M. quien fuera presidente de la SMIA hasta 1989 y muchos científcos más, que sostuvieron sus proyectos bajo condiciones realmente adversas.

En 1986 la visita de los investigadores españoles José Cuena, Feliza Berdejo y Joma Agustí, sirvió de catalizador para la formación de la Sociedad Mexicana de IA. Es en este momento cuando se inicia la consolidación de lo que fueron esfuerzos aislados hacia un objetivo de índole nacional. Para 1986 la IA a nivel mundial ya era un producto comercial interesante, los SE's habían dejado los laboratorios de investigación para ofrecer jugosos dividendos [Fiegenbaum,1988]. Esto junto con la revolución tecnológica (hardware) y las novedosas arquitecturas computacionales de los 80'as, permitió a la IA, convertirse en un producto más interesante para el sector empresarial.

Algunos meses después, la Sociedad Mexicana de Sistemas organiza un coloquio al cual invita al Dr.Cuena para dar una plática. El interés que despertó esta conferencia fué el suficiente, como para que Bancomer SNC a través de Javier Márquez Díaz Cancdo, director de sistemas de la empresa, contratara al Dr. Cuena para desarrollar un SE, para asistir en operaciones de crédito.

En 1987 la empresa del gupo Autrey, de la industria farmaceutica, Bytec decide hacer un SE para control de inventarios de almacenes de medicinas. Esta puede ser considerada la primera aplicación comercial de un SE en México y que además ha mostrado interesantes dividendos.

Paralelamente los cursos de IA se empezaron a popularizar en las universidades. El número de investigadores, instituciones y áreas de interés, se incremento, tal como lo muestran los Volúmenes V yVI de las memorias de las reuniones nacionales de la SMIA.

Aunque indirectamente, la política de desarrollo tecnológico que instrumentó el Gobierno Federal, a través de organismos como CONACYT, que favorecía los postgrados y proyectos de investigación en ciencias computacionales, benefició el desarrollo de la IA en nuestro país.

En 1988 se realizó el primer Simposium Internacional de IA, bajo los auspicios del ITESM campus Monterrey, con la participación de invetigadores de talla internacional como el Dr. Woodrow Bledsoe, el Dr.Adolfo Guzmán, el Dr. Francisco Cervantes, y el Dr. Masaru Tomita entre otros [ITESM, 1988]. Este mismo simposium se llevo acabo por segunda ocasión al siguiente año.

En ese mismo año se realizó la primera reunión IBERAMIA, con la participación de la Sociedad Española para la Inteligencia Artificial y la SMIA principalmente. En este mismo año, en el mes de julio, se realizó el segundo encuentro IBERAMIA. La importancia de estos eventos radica en el valioso intercambio de experiencias. El Dr. Lemaitre afirma que el avance mostrado por la comunidad española es notable, debido a que, en España el financiamento sistemático ha permitido a los grupos de IA despegar hacia proyectos de gran magnitud, como los proyectos ESPRIT. Estos proyectos se formalizan con la participación de empresas y grupos académicos, de diversos países y la Comunidad Económica Europea.

El año de 1989 también tuvo otros eventos de relevancia. El ITESM campus Edo. de Mexico, en el mes de agosto, impartió el primer Diplomado en Sistemas Expertos, en el Centro de Competitividad Internacional, colocando a dicho curso al nivel de otros diplomados que tradicionalmente se imparten ahí [ITESM, 1989].

Instituciones como el IPN, también han mostrado activa participación en la promoción de la IA en México. En el mismo año de 89, pero en el mes de noviembre, el CINVESTAV realizó su Segundo Curso de Internacional de Sistemas Expertos.

No se puede dejar de mencionar a instituciones como la UAM, en la que se han realizado ciclos de conferencias, coloquios y talleres con temas sobre SE's e IA. Asi como las importantes contribuciones que la UDLA ha tenido con exitosas tesis de licenciatura sobre procesamiento de lenguaje natural y robótica. Asi mismo es importante mencionar que en el primer semestre de 1990 se celebró el Primer Seminario Internacional de Robótica en el ITESM campus Edo. de México.

Afortunadamente la lista se ha vuelto interminable y esto pone de manifiesto el interés nacional en torno a la IA y losSE's, además de pronosticar un futuro lleno de retos y grandes oportunidades.

Una tendencia muy difundida en nuestro país es el tratamiento enciclopédico de un tema, que generalmente se centra en un "glosar alrededor del tema" en lugar del tratamiento del tema.

Estamos convencidos que si la IA se debe desarrollar en México, nuestra actitud pedagógica debe ser diferente a la descrita: debemos habilitarnos en la experimentación de

la Inteligencia Artificial, de tal manera que los conceptos básicos de esta disciplina no nos queden difusos, borrosos y a fin decuentas inútiles. Nuestro pais requiere de profesionales que programen conceptos de IA con la misma facilidad con que respiran, pues sólo así podemos aspirar a un desarrollo propio.[Negrete, 1988]

3.4 SITUACION ACTUAL

Esta sección se documento mediante:

Plan Nacional de Desarrollo (CONACYT)

Entrevista Silvia Guardati (ITAM)

Entrevista Osvaldo Cairo (ITAM)

Entrevista a Horacio Carvajal (Banamex)

Folletos. (Operational Expert System Applications in México, Francisco Cantú, ITESM).

Folletos. (Expert Desing System, Carlos Zozaya Gorostiza, CODUMEX)

Análisis de las memorias de S.M.I.A.

Expert Systems Research (Science april 15-83, Richar O. Duda)

Entrevista Agustín Perez (SofTek)

Evolución del mercado en U.S.A. (Expert Sistema Strategies)

Artículos del INEGI

y bibliografia.

Como hemos observado a lo largo de la esta investigación, ya se tienen varios años trabajando en IA en México, lo cual permite reconocer a profesionales y académicos del área como valiosas fuentes de información acerca de la situación actual y futura. Para la obtención de información, se estableciéron parámetros que permiten realizar un análisis cualitativo, de manera que se obtuviera una descripción de las condiciones actuales, en lugar de una medición. Si se considera que la IA en Mexico aún se resiste a salir de las universidades y centros de investigación, pudiese resultar poco productivo, una cuantificación de aplicaciones y beneficios de una industria incipiente en México.

Los cuestionarios de las encuestas se dirigieron a los gerentes de sistemas y líderes de proyecto de las empresas, con quienes se logró contactar. Las preguntas se formularon para determinar:

- Tipo de trabajo realizado o giro de la empresa.
- Tiempo que la empresa tiene trabajando en SE's.
- Tipos de herramientas (Hardware y Software) que usan.
- Perfil del personal y número de personas involucradas en los proyectos.
- Tamaño de los proyectos.
- Objetivos de utilización de SE's (expectativas).
- Estrategia.

Las entrevistas se realizaron con profesionales y académicos reconocidos por la Sociedad Mexicana de Inteligencia Artificial. La estructura de éstas no fué la misma en todas. Esto se hizo, para lograr flexibilidad y extraer lo más fielmente posible, las experiencias de los entrevistados, tanto en el ámbito académico, como en el comercial. Sin embargo, se utilizó un núcleo de preguntas comunes que versaron, entre otros temas, sobre:

- ¿Quiénes hacen SE's en México?;
- ¿Cuáles son sus Objetivos?;
- ¿Qué perspectivas se contemplan?;
- ¿Cuáles son las áreas de oportunidad?;
- ¿Cuál es la problemática al repecto dentro de las empresas?
- ¿Como son los presupuestos?;
- ¿Cuál es el perfil del profesionista en IA y SE's?
- ¿Cuáles son los retos?

Problemática

Dado el proceso de desarrollo que experimenta nuestro país, y las condiciones generales de nuestra industria, donde todavía se suelen encontrar algunos procesos artesanales, no es raro encontrar sistemas de bajo grado de automatización a pesar del "boom " computacional en el que nos encontramos. Lo cual indica que en general, ademas de la escasez de recursos, las implicaciones culturales sobre la organización en la aceptación de nuevas tecnologías no son un problema fácil de resolver. La falta de cultura y conocimiento acerca de los SE's es notable en muchos de los sectores productivos. Hoy en día la mayoría de las empresas no sienten la necesidad de utilizar SE's, puesto que en su realidad cotidiana no se han detectado problemas de competitividad, productividad, o rentabilidad, susceptibles de ser resueltos por estos sistemas. Además los SE's en México se han desarrollado preferentemente en ambientes académicos, por lo que no existen experiencias en los sectores empresariales que hayan probado que esta tecnología puede ser un arma competitiva muy valiosa. Por esta misma razón, los desarrollos empresariales se ven muy limitados de recursos. Es difícil que se inviertan fuertes sumas de dinero en proyectos vanguardistas y relativamente inseguros.

En general, se puede decir que las empresas de nuestro país hoy en día, siguen alguna de estas estrategias en cuanto a SE's se refiere:

- Las compañías tienen algún problema y la resolución de este requiere necesariamente de la aplicación de la tecnología de SE's, sin embargo en muchas ocasiones no se les llama expresamente así, para no comprometerse.
- Las compañías que están interesadas en el área y que generalmente inician los trabajos debido a que han encontrado dentro de su personal del Depto. de Sistemas a alguien con inquietudes personales en torno al tema, y que están dispuesto a dedicarse a desarrollar aplicaciones, sin mucho conocimiento de causa y con recursos notablemente limitados. Ejemplo : Grupo Condumex [Zozaya, 89].
- Las empresas que ya están convencidas de que los SE's son una tecnología útil para su negocio y que tiene los recursos suficientes buscan hacer inversiones más fuertes

contratan personal altamente capacitado en el área para iniciar sus desarrollos. Este esquema es el menos común de los dos y sin embargo ha mostrado mejores resultados en lapsos mucho más cortos. Ejemplo [ITESM, 88].

Otra circunstancia en nuestro país, que detiene la propagación de aplicaciones de SE's, es como nos indica el Ing. Horacio Carvajal "los programas presupuestales no contemplan desarrollos a corto plazo, ya que entre otras razones, se trata con una tecnología todavía muy costosa y con alternativas de inversión no muy diversas". Quienes han incursionado en el campo han detectado que aún no existen suficientes profesionistas del área que desarrollen proyectos empresariales. Además las universidades, que son las entidades que más involucradas deben estar con las tecnologías de punta, no tienen la experiencia en proyectos fuera del ámbito académico y algunas no han podido formar grupos de trabajo formales en el área y los avances mostrados son logros de algunos solitarios y entusiastas investigadores. Además de que pocos grupos científico-académicos han mostrado interés en compartir objetivos con los grupos empresariales. Durante la plática que tuvimos con el Ing. Agustín Pérez detectamos que el sector empresarial exige que para que un proyecto de inversión sea viable, garantice:

- Rentabilidad a corto plazo.
- Eficiencia en el proceso y
- Disponibilidad de asesores reconocidos en el campo.

Algunas instituciones de educación superior ya han incursionado en el terreno de las aplicaciones comerciales y con desarrollos formales han pretendido sentar las bases para la apertura del mercado a los SE's, un claro ejemplo lo tenemos con el ITESM que cuenta con gente como Francisco Cantú quién a iniciado este proceso, sin embargo la mayoría de los proyectos han tenido que detener su evolución debido a que no han logrado la confiabilidad y respeto de los usuarios. Es común que las empresas que cuentan con estos desarrollos no hayan evaluado objetivamente sus sistemas. Algunos directores administrativos esperan obtener beneficios incuantificables de ésta tecnología y muchos otros la han subestimado, aunque si bien es cierto parece que ninguno de estos proyectos han cumplido con las tres

prerrogativas empresariales y han quedado al margen del proceso productivo para el que fueron diseñados.

Se puede decir que los principales problemas que los primeros SE's en aplicaciones comerciales en México tendrán que resolver son:

- Probar que su tecnología resuelve problemas de la vida real.
- Romper la barrera de la cultura organizacional.
- Optimizar sus costos de desarrollo.
- Garantizar su confiabilidad
- Facilitar su operación para usuarios finales.
- Motivar a los expertos a dar mejores aportaciones.

Además el M. en C. Osvaldo Cairo hace referencia a que la parte más difícil en la elaboración de los SE's es sin duda alguna, la obtención del conocimiento ya que como el indica "muchos expertos humanos están demasiado ocupados en diversas actividades como para dedicar un tiempo largo en la tarea de dar el conocimiento para la elaboración de uno de estos sistemas, las empresas están dispuestas a invertir en computadoras y sistemas de desarrollo, pero en realidad muy pocas están dispuestas a ceder el tiempo de sus expertos para el proyecto, ya que éste es quizá el recurso más preciado de la compañía. Al mismo tiempo, los expertos humanos a quienes se pretende modelar, tampoco estan muy convencidos de invertir su tiempo en un experimento, e incluso hay quienes llegan a pensar que lo que se pretende es sustituirlos".

Algo más... (puntos de vista)

Los SE's en México se encuentran en un momento en el que pueden marcar notables ventajas competitivas para quien los sepa aprovechar, dado lo cual todos los proyectos al respecto se manejan con un alto grado de confidencialidad. Además por tratarse de una tecnología tan nueva, los riesgos de desarrollo son elevados, debido a la falta de experiencia

en este campo en las empresas, y es por eso que quien esté trabajando en dichos proyectos no se expondrá al fracaso público. Según dice el Ing. Horacio Carvajal "Banamex actualmente no incursiona en la aplicación de SE's dentro de sus sistemas debido a que no se quiere experimentar y que la valoración de los resultados lleva mucho tiempo, la gente con la que cuenta trabaja en otro tipo de proyectos dentro del área computacional, áreas que de momento tienen mayor prioridad..." con lo anterior podemos detectar ese temor al cambio, a llevar a cabo algo diferente, a invertir recursos económicos y humanos en algo "que puede o no funcionar". En algunos programas se propone hacer coinversiones entre los proveedores y el cliente para minimizar el riesgo de ambos. Los proveedores requieren de apoyarse siempre en sus sucursales del extranjero salvo en ocasiones en las que subcontratan a investigadores de las universidades del país y los clientes no están dispuestos a cargar con toda la responsabilidad ante los directores y consejos de accionistas. Otras empresas (algunas de las cuales llevan hasta tres años tratando de construir SE's) han decido contratar asesores externos para colaborar con sus proyectos como apoyo, pero sin intervención directa. Esto ha puesto en evidencia, dada la prontitud de resultados de quienes han obtenido el asesoramiento adecuado, que se habían estado gastando recursos inadecuadamente y que los fracasos de este lapso no se debían a la debilidad de la tecnología de SE's, sino a una deficiente estructuración de los proyectos.

Como se había mencionado, una de las prerrogativas de los proyectos de inversión es la rentabilidad. Se puede decir que uno de los sectores empresariales donde la rentabilidad de un proyecto tiene mayor impacto a más corto plazo es el bancario y de seguros, por eso es que las empresas más interesadas en esta tecnología, en el caso de nuestro país, pertenecen a este sector . Además existen otras dos buenas razones para que dichas empresas se conviertan en campo de fértiles desarrollos. Primeramente la disponibilidad de recursos es amplia, y las políticas presupuestales gozan de menores restricciones; y en segundo lugar estas empresas, tienen un sin número de tareas diarias afectando directamente sus líneas productivas, con todas las características idóneas para que un SE pueda trabajar y mostrar resultados al corto plazo.

Algunas empresas multinacionales de la industria manufacturera en México, ya tienen planes de trabajo que contemplan desarrollos de SE's. Su estrategia es similar a la que

en este campo en las empresas, y es por eso que quien esté trabajando en dichos proyectos no se expondrá al fracaso público. Según dice el Ing. Horacio Carvajal "Bananex actualmente no incursiona en la aplicación de SE's dentro de sus sistemas debido a que no se quiere experimentar y que la valoración de los resultados lleva mucho tiempo, la gente con la que cuenta trabaja en otro tipo de proyectos dentro del área computacional, áreas que de momento tienen mayor prioridad..." con lo anterior podemos detectar ese temor al cambio, a llevar a cabo algo diferente, a invertir recursos económicos y humanos en algo "que puede o no funcionar". En algunos programas se propone hacer coinversiones entre los proveedores y el cliente para minimizar el riesgo de ambos. Los proveedores requieren de apoyarse siempre en sus sucursales del extranjero salvo en ocasiones en las que subcontratan a investigadores de las universidades del país y los clientes no están dispuestos a cargar con toda la responsabilidad ante los directores y consejos de accionistas. Otras empresas (algunas de las cuales llevan hasta tres años tratando de construir SE's) han decido contratar asesores externos para colaborar con sus proyectos como apoyo, pero sin intervención directa. Esto ha puesto en evidencia, dada la prontitud de resultados de quienes han obtenido el asesoramiento adecuado, que se habían estado gastando recursos inadecuadamente y que los fracasos de este lapso no se debían a la debilidad de la tecnología de SE's, sino a una deficiente estructuración de los proyectos.

Como se había mencionado, una de las prerrogativas de los proyectos de inversión es la rentabilidad. Se puede decir que uno de los sectores empresariales donde la rentabilidad de un proyecto tiene mayor impacto a más corto plazo es el bancario y de seguros, por eso es que las empresas más interesadas en esta tecnología, en el caso de nuestro país, pertenecen a este sector . Además existen otras dos buenas razones para que dichas empresas se conviertan en campo de fértiles desarrollos. Primeramente la disponibilidad de recursos es amplia, y las políticas presupuestales gozan de menores restricciones; y en segundo lugar estas empresas, tienen un sin número de tareas diarias afectando directamente sus líneas productivas, con todas las características idóneas para que un SE pueda trabajar y mostrar resultados al corto plazo.

Algunas empresas multinacionales de la industria manufacturera en México, ya tienen planes de trabajo que contemplan desarrollos de SE's. Su estrategia es similar a la que

describe E. Feigenbaum en su libro "The Rise of the Expert Company" [Feigenbaum,88] acerca de los SE's desarrollados por Dupont. Estas organizaciones han creado un departamento de SE's dentro de su departamento de Sistemas e Informática como un centro de apoyo a usuarios. La idea central es que sean los propios usuarios quienes desarrollen sus propios SE's. Desde luego estos primeros desarrollos se contemplan como pequeñas islas de automatización de las tareas administrativas individuales de algunos departamentos dentro de la compañía. Por supuesto dichos sistemas están todavía muy lejos de afectar directamente las líneas de producción. Sin embargo, esta estrategia permite que usuarios completamente ajenos a los sistemas computacionales, utilizando herramientas muy sencillas para plataformas computacionales de escritorio, se introduzcan a una tecnología que en pocos años marcará la pauta dentro de la organización. De esta manera los riesgos se minimizan, ya que lo peor que puede ocurrir es que los procesos que se pretendieron automatizar se sigan haciendo a mano como hasta ahora. Sin embargo es tal el número de pequeños SE's en desarrollo que la posibilidad de obtener un buen número de proyectos exitosos es muy alta, lo que da a la compañía una prueba fehaciente de la viabilidad de ésta tecnología.

Sectores con Mayor Interés

En México las empresas del sector bancario y de seguros han sido las más interesadas en esta tecnología, por que en su ámbito es más fácil probar la rentabilidad de estos sistemas. Además pertenecen a un sector con mayor liquidez y menores recortes presupuestales.

A grosso modo, se puede decir que las áreas de oportunidad para los SE's dentro de una empresa, son básicamentede dos tipos:

Areas operativas

Areas productivas

Las empresas del sector bancario y de seguros están sujetas a procesos productivos, con las características ideales para la aplicación de SE's. En ellas encontramos tareas poco estructuradas, pero con políticas de acción claras y consisas.

Por ejemplo, no es difícil pensar en un sistema que auxiliara a un centro de atención de usuarios para autorización de crédito (American Express) [Figenbaum, 1988] o un sistema de emisión de pólizas de seguros de vida como el de "Nipon Life" [Fiegenbaum,1988], por mencionar algunas aplicaciones. Esto no quiere decir que en nuestro país no existan otros sectores igualmente interesados.

El M.en C. Osvaldo Cairo opina "la medicina es una de las áreas de mayor apertura a este tipo de tecnología, los expertos en esta área son más abiertos a dar su conocimiento, y existe recursos disponibles para la implantación de estos, además la aportación de estos SE's en medicina a la sociedad es importantísima, imagínense que las personas que viven en lugares muy remotos y que no pueden viajar a los lugares donde estan los médicos expertos humanos, tengan acceso al conocimiento de ellos mediante esta tecnología".

La industria de manufactura, es otro de los sectores que también han mostrado interés. De acuerdo a lo reportado por el ITESM, en sus proyectos con la industria textil [ITESM, 88], los sistemas que más les han interesado son aquellos que están orientados a cuidar la consistencia de los lotes del producto. En esta industria, existen procesos con estrictos controles de calidad y en muchos de ellos, por imperfecciones en el mismo, pierden diariamente varios lotes del producto. La gran mayoría de éstas fallas no tiene un origen único, sino que en realidad son multifactoriales, y la mayoría de las veces dependen fuertemente de la experiencia del operador. Esta resulta entonces, una problemática muy adecuada a los tipos de problemas que resuelven los SE's.

Una buena aplicación de los SE's en la industria petrolífera, puede ser el tratamiento (mantenimiento) de pozos petroleros de explotación; proceso que se realiza todos los días. A medida que este tratamiento es mejor, el pozo produce más. El tratamiento de estos pozos implica manipular numerosas variables con criterios que varían notablemente de pozo a pozo y de terreno a terreno, actualmente según nos indica el Ing. Agustín Pérez Softek esta trabajando en varios proyectos con el IMP, precisamente para desarrollas "expertos" en sus áreas problemáticas. Los expertos humanos en esta tarea no son abundantes y además son personas que han acumulado su experiencia a través de muchos años.

Silvia Guardati M. en C. opina que "las áreas de aplicación son muchas, sin embargo se debe crear una cultura de SE's para generar credibilidad en los clientes, y que de esta forma no sientan temor al invertir en esta tecnología"

En general, se ha coincidido en que los sectores mas interesados son:

Bancos
Casas de Bolsa
Aseguradoras
Industrias Quimicas
Industrias Petroleras
Mercaderías (Supermercados) y
Empresas de Comunicaciones Electrónicas de amplia difusión.

3.5 FORMACION ACADEMICA

- Planes UNAM
- Osvaldo Cairo. (ITAM).
- Silvia Guardati. (ITAM).
- Agustín Pérez (Softek)
- Horacio Carvajal (Banamex)

Para determinar un perfil del egresado de licenciatura, especialista en tópicos de IA y SE's, hay que empezar por distinguir los tres campos de acción que existen en Mexico, que según lo advierte el M.C. Osvaldo Cairo, son:

Desarrollo Academico
Investigación Aplicada
Desarrollo de sistemas comerciales

Sentado lo anterior, es posible definir las cualidades y las necesidades que cada uno de los campos demanda.

Desarrollo Académico

La importancia de trabajar en un área como ésta es fundamental en la consecución de la autonomía tecnológica. Según lo señala la M. en C. Silvia Guardati "Sin un buen desarrollo Académico de esta tecnología no es posible manejar, desarrollar y en un momento dado comprar ya que no existirían los fundamentos concretos que sustentaran esta tecnología." El patrimonio cultural de una universidad y por ende, el de un país, depende en gran medida de los descubrimientos científicos obtenidos. Grandes problemáticas nacionales pueden ser resueltas, por lo menos parcialmente, gracias a sus científicos. De ahí que sea importante apoyar la formación de grupos de ciencia básica.

En las universidades del país existen un sin número de planes de estudio, de diferentes carreras y grados académicos, que contienen materias relacionadas a la IA y SE's. Sin embargo, los temarios varían de acuerdo a las preferencias del profesor. La M. en C. Silvia Guardati apunta en la entrevista: "Se puede incluir el estudio de los métodos de búsqueda heurística en cualquier curso de programación algorítmica de nuestras carreras de ciencias computacionales, aún cuando los alumnos no estén involucrados en IA". Por ejemplo el M. en C. Osvaldo Cairo, en los curso de Sistemas Expertos que imparte en el ITAM, dice "Primero los alumnos empiezan creando programas expertos como por ejemplo uno que aprenda a jugar dominó y posteriormente se les encamina a casos más reales haciendo ingeniería de sistemas, lo importante es que aprendan una metodología". El Dr. Lemaitre opinó que "una formación básica y sólida en IA, es más valiosa que cursos aislados de SE's. Para mí un curso básico debe versar sobre temas como: lógica, métodos de búsqueda, problemas de representación, tratamiento de lenguaje natural y aprendizaje. A los alumnos de maestría, en los cursos de SE's, se les debe de pedir que programen -todo!, la máquina de inferencia, las interfaces, etc."

Esta variedad de opiniones de los especialistas, deja ver un punto en común: el tratar de abarcar todas las áreas en un solo curso es inaceptable. Esto justifica uno de los objetivos del Laboratorio de Desarrollo de Sistemas Expertos, que se tratarán más adelante. Tomando en cuenta que los proyectos de investigación deben ser formativos, se le debe proporcionar

al alumno, una herramienta que le permita dirigir sus intereses académicos con la debida solidez de conocimientos.

Quienes se interesen en este campo, no pueden olvidar que los resultados son a largo plazo y que como indica el Ing. Agustín Pérez "las personas que están en desarrollo prefieren establecer equipo de trabajo que duren muchos años, por que saben que es la única forma de ofrecer resultados trascendentes". Al respecto, el M. en C. Osvaldo Cairo apunta: "es difícil trabajar con alumnos de Ingeniería, ya que sus proyectos no pueden extenderse en tiempo y algunos problemas interesantes de resolver, se deben descomponer en subproblemas demasiado pequeños, y caemos en crear 'expertitos académicos', lo que intento es a que los tesistas de Ingeniería se les despierte el interés en formar equipos de investigación básica...". Sin embargo, en la UDLA de Puebla existen trabajos de tesis sobre procesamiento de lenguaje natural, que han alcanzado grandes logros.

Investigación Aplicada

Esta es probablemente una de las áreas mas desatendidas y en donde se encuentran muchos de los problemas de las relaciones Universidad-Industria. Este no es un problema exclusivo de nuestro país; en los EUA por ejemplo, investigadores como el Dr.Simon reconocen la necesidad de formar recursos humanos en este ramo.

El profesionista que atienda esta área, debe caracterizarse fundamentalmente, por tener la habilidad de estructurar soluciones através de una metodología de solución general de problemas. Al respecto, el M. en C. Osvaldo Cairo opina que "quienes trabajen en este campo deben tener una gran capacidad de abstracción, para plantear soluciones alternativas bajo diferentes enfoques tecnológicos, el Ingeniero del Conocimiento debe de estar a la vanguardia de varios temas para ser capaz de comprender las ideas que el experto humano le expresa y poderlo plasmar en reglas". El Ing Horacio Carvajal advierte que "el papel que juega el Ing. del Conocimiento es importante, pero no por ello se debe caer en crear una Licenciatura de esta materia ya que aunque en USA ya se hizó no es lo óptimo sería como hacer de la ingeniería de análisis una Licenciatura" El especialista de IA y SE's en esta rama, debe tener una visión global de las posibilidades que le brinda, el aplicar técnicas de

IA a problemas que tradicionalmente requerían de cómputo intensivo o de modelos muy simplificados para su solución. Lo más importante de éste profesionista no es conocer detalladamente los más especializados algoritmos o métodos de búsqueda, su compromiso es el de reconocer la gama de herramientas que la IA le proporciona y saberlas aplicar adecuadamente.

Desde luego la mejor forma de dominar una herramienta es utilizándola, de ahí que los convenios de la industria con la Universidad sean fuentes importantes de experiencia. Varios de los especialistas entrevistados, coincidieron en que los tesistas, aún de Ingeniería, son buenos candidatos para dichos proyectos. Pero al mismo tiempo advirtieron de la gran responsabilidad que la Universidad y sus estudiantes deben asumir, para cuidar que la tecnología de SE's sea introducida al sector industrial de una manera convincente y lo suficientemente confiable como para generalizar su uso. Contar con un laboratorio de desarrollo de SE's, resulta indispensable en la consecusión de este objetivo.

Desarrollo de Sistemas Comerciales

El área de los sistemas comerciales se encuentra dividida en dos grandes grupos. El primero es el de las empresas que se dedican explícitamente a desarrollar SE's, y el segundo, el de empresas o corporaciones, que dentro de sus departamento de sistemas, tiene un grupo de personas dedicadas a SE's. Los requerimientos de personal, para cada una de las áreas son diferentes. Según indica el Ing. Agustín Pérez, un desarrollo formal de un SE debe estar dirigido por personal con formación de postgrado, en vista de la complejidad técnica y económica que implica. El coordinar sesiones de adquisición de conocimiento, plantear esquemas de representación, etc., no son actividades triviales cuando se trata de problemas del mundo real. Para las empresas que desarrollan SE's, su compromiso es de gran magnitud y no debe permitir errores. En cambio para las empresas que como parte de sus proyectos de desarrollo, trabajan en SE's, las más de las veces se apoyan en consultores que fungen como líderes de proyecto y responsables del mismo, de ahí que el personal asignado sólo requiera conocimientos básicos sobre tópicos de IAy SE's. Para las empresa que buscan su independencia tecnológica, esta puede ser una estrategia que ofrezca buenos

resultados temporalmente, pero a la larga la capacidad y habilidades de su personal habrá de ser mejor.

3.6 PLANES DE ESTUDIO

-PLAN ITESM
-PLAN UTAM
-PLAM UNAM
-PLAN ULSA
-PLAN IBEROAMERICANA

El siguiente material presentado se obtuvo de las diferentes universidades a las que se hace mención. La idea es revisar si contemplan dentro de sus Licenciaturas relacionadas al área de Computación materias que tengan que ver con SE's o I.A. La siguiente gráfica muestra cuantas licenciaturas de cada una de éstas universidades, ofrece alguna materia o materias con tópicos relacionados a Inteligencia Artificial y/o SistemasExpertos.

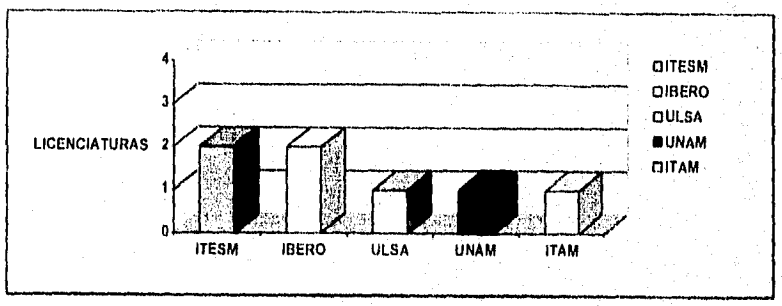


Ilustración 1

En la gráfica se puede observar que apesar de que existen ya diversas instituciones dedicadas a la formación de recursos humanos en el área, el curriculum que contemplan para sus licenciaturas tiene un muy bajo porcentaje de materias relacionas con SE's y/o IA. Probablemente sería recomendable proponer toda una área de opción terminal o subsistema, para las licenciaturas relacionadas con ciencias computacionales e ingeniería, de manera que al finalizar la Ingeniería, el alumno tenga un mejor conocimiento de la materia, y su decisión sobre su siguiente etapa profesional esté mejor fundamentada. Hasta donde la información disponible lo permite, y a decir de los profesores y personas relacionadas, la mayor parte de los egresados en esta especialidad, se dedican a la investigación en ciencias básicas y muy pocos se incorporan a la investigacion aplicada o al desarrollo de sistemas para empresas. De acuerdo a la gama de aplicaciones de SE's, se podría proponer que cualquier licenciatura en cualquier área de ingeniería, debería contemplar en sus planes de estudio, asignaturas con temarios sobre conceptos básicos en esta tecnología, y aquellas que además se encontraran fuertemente relacionadas con las ciencias computacionales, tuvieran la opción de profundizar en el tema, a continuación se muestra en la gráfica el No. de materias relacionadas con I.A. o SE's que tienen las licenciaturas en su plan de estudios.

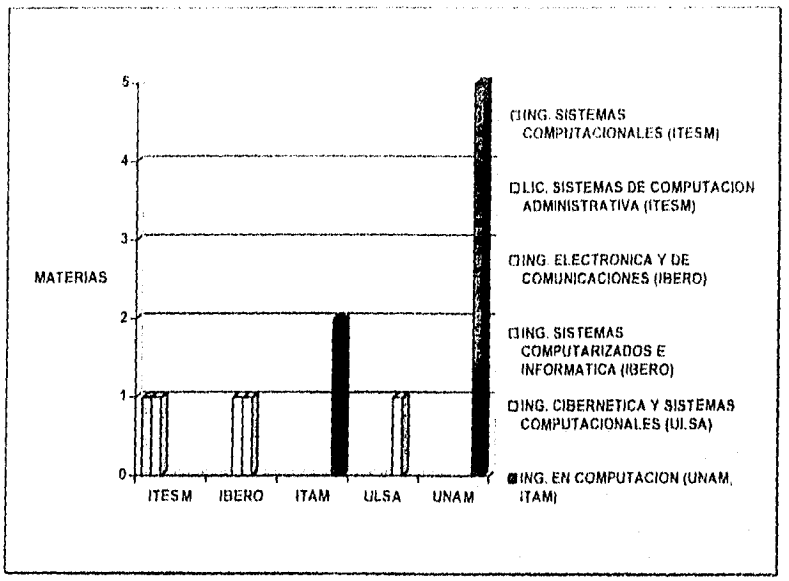


Ilustración 2

3.7 PROYECCIONES FUTURAS

Este punto se documento con:

Entrevista Osvaldo Cairo

Entrevista Silvia Guardati

Información del CONACYT

Información del INEGI

El mercado para SE's en México es potencialmente inmenso. Las áreas de oportunidad son muy extensas; sin embargo, el desconocimiento de una tecnología nueva, complica su

aceptabilidad en el medio. Además, existen otros factores como la dificultad de que las empresas se involucren en tecnologías vanguardistas cuando muchas de ellas no han podido resolver eficientemente muchos de los problemas computacionales tradicionales, tales como nóminas, contabilidad, control de inventarios, y demás procesos administrativos. En nuestro país todavía existen organizaciones empresariales que se resisten al cambio tecnológico y a la adquisición de procesos de automatización más eficientes. A veces son los usuarios finales quienes rechazan los sistemas, a veces los directivos y a veces el mismo personal de las áreas de cómputo. Para el futuro, es indispensable, que los SE's sean introducidos en las áreas de computación tradicional de las empresas, de manera que los proyectos al respecto, muestren un respaldo ante la dirección y se cumpla con la tercera de las prerrogativas empresariales antes mencionadas.

Como en cualquier actividad económica, el posicionamiento de un producto en el mercado es el resultado de un proceso multifactorial. En México el mercado de SE's no es distinto y su comportamiento se ve afectado por factores como:

- Difusión de la tecnología.
- Disponibilidad de profesionistas capacitados en el área.
- Disponibilidad de recursos.

Por lo que a la difusión de la tecnología se trata, ya se ha mencionado que hasta ahora muchos profesionistas, directores y usuarios no la conocen y mucho menos reconocen sus beneficios.

En cuanto a la disponibilidad de profesionistas reconocidos en el área también se ha mostrado que es escasa. Es indudable la necesidad de formar recursos a los más altos niveles tanto para impulsar al sector empresarial como para nutrir a los grupos científicos y académicos del país. No hay que buscar mucho para darse cuenta de lo útil que puede ser tener personal trabajando en aplicaciones comerciales, como en investigaciones científicas para acortar el camino hacia la independencia tecnológica.

En nuestro país existen ya distintas instituciones académicas que cuentan en sus planes de estudio de licenciatura y maestría con programas orientados a la formación de profesionistas especialistas en el área.

Al hablar de la disponibilidad de recursos hay que tomar encuenta algunos factores que rigen el mercado de los sistemas computacionales en general. Como ya se dijo, la mayoría de las empresas interesadas en el campo no estan dispuestas o en disponibilidad de asignar grandes recursos para sus desarrollos. Aquellas que cuentan con infraestructuras y recursos computacionales de tamaño considerable, los tienen dedicados en un alto porcentaje a sus procesos productivos y en la mayoría de los casos al desarrollo de sistemas tradicionales, por lo cual la posibilidad de involucrarlos en proyectos experimentales es reducida. Por otra parte, la vertiginosa evolución de las plataformas computacionales de escritorio, muestra interesantes alternativas para la asignación de recursos, minimizando el riesgo para la organización y el impacto sobre las líneas de producción. Además, el mercado de computadoras de escritorio se encuentra en plena ebullición competitivamente hablando, sobretodo si se toma en cuenta que ahora no existe gran diferencia en precios entre máquinas basadas en los microprocesadores más grandes de la familia Intel, las máquinas basadas en procesadores de la familia Motorola y las estaciones de trabajo de tecnología RISC, dado lo cual es posible instalar un pequeño laboratorio con herramientas profesionales para aplicaciones serias con inversiones relativamente pequeñas que oscilan entre los $ 15,000 y $ 25,000 dólares, que para una compañía grande no resulta realmente caro. Esta es una razón por la cual varias compañías han adoptado la estrategia de desarrollar y producir sobre máquinas de escritorio, pero tambien existen otras dos razones muy importantes:

La primera es, que nadie está dispuesto a trabajar en proyectos muy ambiciosos con una tecnología "experimental"

y la segunda es la capacidad instalada que muchas compañías tienen en plataformas computacionales de escritorio. De esta manera si sus proyectos tienen éxito habrán resuelto dos problemas: uno, el de la confiabilidad de los SE's para resolver casos de la vida real, y dos, el aprovechamiento integral-cooperativo de los recursos computacionales y la

distribución del proceso productivo, que han marcado puntos críticos en la operación de las empresas.

3.8 CONCLUSIONES

La tecnología de SE's es una realidad que se encuentra lejos de ser estudio trivial. El desarrollo de esta tecnología tiene diversidad de problemas y en ocasiones son de alto grado de complejidad técnica y conceptual. Sin embargo en general la problemática se puede resumir del siguiente modo:

El desarrollo de SE's no han tenido éxito en su aplicación en sistemas convencionales como son nóminas, contabilidad, finanzas, etc., los cuales son fundamentales en la organización empresarial de allí su falta de comercialización, si a esto le agregamos lo nuevo de la tecnología que usa y el desconocimiento de ésta, tenemos un fuerte obstáculo a librar, sin embargo es de hacer notar que otras tecnologías que hoy son poderosas como las de comunicaciones, redes, antes no eran comerciales y se tenían las mismas dificultades en un principio.

Las empresas que están incursionando en desarrollar SE's, están dispuestas a contratar egresados de licenciatura con conocimientos de programación en técnicas de IA; sin embargo, no excluyen la necesidad de dar capacitación complementaria. Otras empresas de gran tamaño han preferido empezar con pequeños SE's, para lo cual el personal a cargo, necesita una formación más solida en métodos generales de resolución de problemas, que en conceptos de IA. Sin embargo, todos los entrevistados coinciden en que hace falta formar recursos humanos en IA.

Es muy pronto para estimar el crecimiento de los Deptos. de IA y SE's en las diferentes empresas, sin embargo se observa un creciente interés de dichas entidades en esta tecnología.

Los recursos humanos en las empresas, no son extensos; ya que cuesta mucho trabajo encontrar programadores de IA e Ingenieros del Conocimiento para aplicaciones empresariales.

Muchas industrias de nuestro país, no han encontrado profesionistas y/o empresas que las apoyen en su desarrollos, por lo que han traído asesores de USA. Esta es una opción

cara, considerando que dichas asesorías cuestan entre $120.00 y$1000.00 USD. la hora, además de los viáticos.

Los sectores más interesados son:

Bancos

Casas de Bolsa

Aseguradoras

Industrias Químicas

Industrias Petroleras

Medicina

Mercaderías (Supermercados) y

Empresas de Comunicaciones Electrónicas de amplia difusión.

Muchas empresas en México que se dicen creadoras de SE's no son lo suficientemente competentes en el ramo por lo que en lugar de generar confiabilidad en los clientes crea inseguridad en ellos.

Es muy importante que las universidades tengan un laboratorio de SE's, pues además de generar personal capacitado en esta área, pueden colaborar a la apertura del mercado, desarrollando prototipos para acercar al sector empresarial a esta tecnología. Al mismo tiempo, su responsabilidad es muy grande, pues la seriedad con la que se afronten los proyectos, ser la que contribuya a la apertura o clausura del mercado, ya que un desarrollo irresponsable puede retrasar considerablemente la evolución de esta tecnología en México.

Dadas las condiciones del mercado, la tendencia es de desarrollar SE's en plataformas computacionales de escritorio.

CAPITULO 4
HERRAMIENTAS DE CONSTRUCCION DE SISTEMAS EXPERTOS

4.1 INTRODUCCION

Antes de implementar cualquier sistema de computadora se deben considerar los factores de ajuste exacto para el problema en cuestión. El criterio para seleccionar el software apropiado es interdependiente y a veces conflictivo; esto es, en ocasiones las características de la herramienta simplifican la labor del programador pero a costos muy elevados. Cuando se escoge una herramienta se deben considerar costos, características principales, ambientes de trabajo para desarrollo/producción, y el tipo de problema a resolver. Todos los criterios de selección deben ser aplicados juntos antes de tomar una decisión.

En las siguientes secciones se exponen razones y características que permiten formar criterios más detallado sobre la utilidad de usar lenguajes y/o hasta shells para la construcción de sistemas expertos.

En general, se puede decir que la problemática de construcción de sistemas expertos comparte características generales con la construcción de sistemas tradicionales. Sin embargo, históricamente, la comunidad de Inteligencia Artificial ha tenido requerimientos novedosos y diferentes al resto de la comunidad en áreas computacionales, y ha tenido que resolverlos creando y adaptando sus propias técnicas y herramientas. Algunos de estos requerimentos son los que se han derivado de la representación de conocimiento con métodos de solución de problemas que usualmente involucran búsqueda, generación de soluciones oportunísticas y análisis de medios y fines.

Estas mismas necesidades han determinado que en el mercado surjan una gran cantidad de herramientas orientadas a cubrir éstos requerimentos. Una manera de clasificar estas herramientas es como sigue:

- **Lenguajes:** Que pueden ser de propósito general tales como C,Lisp, Pascal, etc. o especializados como el intérprete de reglas OPS5 o Framekit que opera con marcos.

  Esqueletos: Que se derivan históricamente de sistemas expertos como Mycin o Prospector y que han sido utilizados para resolver problemas similares para los cuales fueron creados.
- **Ambientes y/o Shells:** Como KEE o Knowledge Craft, que presenta características similares a las de un lenguaje pero que incluyen estructuras previamente construidas como el motor de inferencia con métodos de búsqueda intercambiables, o bien con alguna forma de representación de conocimiento. Es comunmente reconocido que el término shell se refiere a un ambiente de programación experto que provee de paradigmas generales de razonamiento de Inteligencia Artificial. Estos ambientes son llamados así usualmente porque sus paradigmas de computación se encuentran apoyados sobre otros substratos como Lisp o C. Nexpert Object, PC Consultant Plus por ejemplo, no sólo están escritos en alguno de éstos lenguajes, sino que incluyen además un número grande de módulos de todo tipo, como representación de conocimiento, métodos de inferencia, explicación, interfaces, etc.

4.2 *CONSTRUIR O COMPRAR?*

Siempre existe la necesidad de decidir si comprar o construir módulos y paquetes completos para la realización de un proyecto. Las herramientas existentes son bastante adecuadas para la mayoría de las necesidades de programación. El costo de las herramientas para desarrollo de sistemas basados en conocimiento, no difiere mucho del costo de los sistemas convencionales. Las herramientas para PC's se encuentran en un rango de entre $100 y $15,000 USD., las que operan sobre minicomputadoras y estaciones de trabajo llegan a costar hasta $75,000 USD, y las de sistemas mayores que van desde los $25,000 hasta los $250,000 USD. Estos costos exorbitantes, no lo parecen tanto si se considera el costo de desarrollo y los beneficios esperados de la aplicación de los sistemas.

Generalmente el software de minicomputadoras es 10 veces más caro que el de PC's y a su vez el software de sistemas mayores es de un orden de 10 veces más caro que el de las minicomputadoras. Desde luego un precio más alto no garantiza una mayor funcionalidad. Muchas herramientas de PC's son tan funcionales como las de sistemas mayores. Es importante empalmar las funciones existentes en una herramienta con las necesidades del problema, mientras se consideren los ambientes de desarrollo y producción. Es perfectamente válido desarrollar en PC y colocar el sistema en un sistema mayor para ser explotado, habiendo tomado en cuenta los costos de migración de un ambiente a otro. La funcionalidad es crítica. Se debe escoger la herramienta que provea las funciones necesarias para el proyecto. Si se tiene una herramienta incompleta para el problema en cuestión, es probable que se requiera de más inversión, ya sea para comprar otra herramienta que se ajuste mejor o para programar módulos que cubran las deficiencias encontradas.

Bajo ciertas circunstancias resulta conveniente construir un shell apropiado. Si se utilizan reglas, marcos u objetos por ejemplo, puede ser mucho más rapido usar una herramienta especializada, que se ajuste a las necesidades del problema. Los shells que se especializan en la captura del conocimiento en un dominio específico frecuentemente reducen el tiempo de codificación del conocimiento. Sin embargo se puede construir, por ejemplo; una herramienta especializada en administración de procesos de manufactura. Así se construye una herramienta que "entienda" procesos de manufactura (control de inventarios,líneas de ensamble, tiempos de entrega, etc.); entonces no será necesario tratar éstos procesos como casos especiales ya que serían parte del ambiente de programación. De esta manera, la atención se mantendrá sobre los aspectos importantes del dominio en lugar de los conceptos que deberían ser periféricos al problema.

Otra circunstancia bajo la cual se debe construir una herramienta propia, es cuando las existentes no tienen las facilidades internas que se consideran importantes tanto para desarrollo como para mantenimiento del programa. Es común encontrar limitaciones con respecto a la gama de fuaciones, que proporcionan los shells, lo cual es natural en una industria joven como la de los sistemas expertos. Sin embargo, muy pronto estarán en el mercado nuevas y mas poderosas herramientas, que sin duda harán más recomendable el desarrollar sistemas expertos en ellas.

Una de las principales desventajas de construir la herramienta propia, es la económica. Las herramientas se construyen generalmente para propósitos específicos y pocas veces son adaptables a la generalidad de los problemas sin un trabajo extra en la programación. Es definitivo que construir una herramienta para un problema en particular, es poco efectivo tomando en cuenta el tiempo requerido, los recursos y otros costos, a menos de que se trate de un sistema que pueda dar muy grandes ahorros. El costo de desarrollo de una herramienta es tan solo una parte. Puede ser más significativo el costo de mantenimiento y corrección de fallas o la inevitable actualización que garantice la vitalidad del sistema. En conclusión la construcción de herramientas propias es extremadamente costosa en tiempo y dinero y es indispensable hacer una evaluación de ambos antes de tomar una decisión.

4.3 *LENGUAJES PARA INTELIGENCIA ARTIFICIAL*

Según R.A.Corllet [Corlett 86] el lenguaje de programación ideal, es aquel en el cual, cualquier problema de cualquier clase puede ser codificado. Sin embargo, es sabido que cada lenguaje parece más adecuado para ciertas tareas que otros. Así, el problema de seleccionar un lenguaje para una aplicación determinada, depende de las características de éste.

Programar en IA es básicamente modelar y simular habilidades cognositivas humanas. Probablemente, el aspecto más importante de la IA es la creencia de que un comportamiento inteligente puede darse a través de la manipulación de símbolos. Esta creencia es llamada hipótesis del sistema de símbolos físicos.

Los elementos básicos que necesitan ser manipulados por los programas de IA son símbolos. Estos símbolos pueden ser usados para representar cosas que pueden ser incorporadas a estructuras complejas que representen cosas aún más complejas. Típicamente, los programas de IA desean construir estructuras de datos arbitrarias y complejas, puesto que el conocimiento, la esencia de la inteligencia, es arbitrariamente complejo. La solución a muchos problemas de IA consisten por lo tanto, en generar un gran número de estructuras de datos intermedios, las cuales actúan como representaciones de lo que hay que seguir . Por ejemplo, el procesamiento de lenguaje natural involucra el uso de

árboles de analizadores lexicográficos, y los solucionadores generales de problemas usan estados intermedios del mundo para representar soluciones parciales de un problema. Ni el número, ni la estructura de algunos de estos objetos pueden ser predeterminados al momento de escribir el programa. Esto es intrínseco a la naturaleza de los problemas que han de ser resueltos. De tal forma, lo único que se puede decir de estas estructuras de datos, es que pueden contener componentes igualmente impredecibles.Uno de los resultados claves que han aflorado de la investigación en IA, es que para construir sistemas complejos inteligentes, es necesaria una considerable cantidad de conocimiento. Este conocimiento existe en diferentes formas. Necesita ser creado, consultado y modificado continuamente.

Desafortunadamente, así como se necesita una cantidad considerable de conocimiento para solucionar problemas serios de aplicabilidad práctica, existe el peligro de que la búsqueda de el conocimiento relevante a una subtarea en particular, pueda tomar una gran cantidad de tiempo, computacionalmente hablando. En un sistema de IA, la búsqueda de conocimiento relevante es generalmente manejada utilizando técnicas de apareamiento de patrones.

Muchos de los programas de IA usan implícitamente técnicas de búsqueda, tanto del conocimiento almacenado en el sistema, como de rutas alternativas de solución. Complejos regímenes de control son empleados en algunos de éstos programas, lo que hace que el comportamiento del sistema sea impredecible hasta que es ejecutado. Esta es frecuentemente la razón por la cual las técnicas de IA son empleadas en aquellos problemas que no tienen una solución claramente algorítmica. En otras palabras, si se conoce un algoritmo para resolver un problema, lo recomendable es usar un lenguaje algorítmico como FORTRAN, Pascal o C; pero si se sabe que no existe dicho algoritmo, entonces las técnicas de IA pueden resultar una buena alternativa.

4.3.1 Implicaciones del Lenguaje

Según E. Rich [Rich 1983] existe una lista de características requeridas en cualquier lenguaje para construir sistemas grandes y complejos; tales como los de la IA:

- ***Variedad de tipos de datos*** para describir las muchas clases de información que un sistema grande necesita.
- ***Habilidad para descomponer el sistema*** en unidades más pequeñas y comprensibles que permitan hacer cambios fácilmente a sólo alguna parte del sistema.
- ***Estructuras flexibles de control*** que permitan recursión y la descomposición en paralelo del sistema.
- ***Habilidad para comunicarse interactivamente con el sistema***, tanto durante su desarrollo como en su explotación.
- ***Habilidad para producir código eficiente*** de manera que el desempeño del sistema sea aceptable.

A pesar de que no existe nada sorprendente en la lista anterior, se sugieren las siguientes características adicionales para sistemas de IA:

- ***Orientado a símbolos*** de manera que manejar éstos sea sencillo.
- ***Procesamiento de listas*** Las listas encadenadas se han encontrado particularmente adecuadas para representar aquellas estructuras de datos arbitrarias y dinámicas.
- ***Localización automática de almacenamiento y recolección***, de manera que, el programador no deba preocuparse por donde alojar las estructuras interrelacionadas y tampoco como recolectar basura de la memoria ya utilizada.
- ***Oportunidad de hacer instanciaciones tardías*** para poder usar procedimientos cuyos parámetros o argumentos sean de tipos indeterminados y que éstos se definan durante la ejecución.
- ***Facilidades de apareamiento de patrones*** para identificar datos y determinar el control.
- ***Habilidad para implementar estrategias y estructuras de control alternativas*** de manera que puedan pasarse procedimientos como datos y crear nuevos procedimientos durante la ejecución.
- ***Fusión de procedimientos y datos*** para poder combinarlos en objetos singulares de manera tal, que como una sola entidad puedan ser pasados como un valor. Este empacado de datos y procedimientos dentro de un objeto singular es conocido como programación orientada a objetos.

- ***Estructuras de datos para representar procedimientos*** esto hace posible escribir programas que manipulan la estructura de programas muy fácilmente. Existen varios ejemplos de programas que construyen nuevos programas durante la ejecución. Esta técnica es mas usada cuando el rango de posibles comportamientos es demasiado grande para codificar estructuras de datos eficientemente, pero que pueden ser fácilmente expresadas como combinación de fragmentos de procedimientos.

A continuación iniciaremos una exposición sobre los lenguajes utilizados para el desarrollo de sistemas basados en el conocimiento.

4.3.1.1 Lenguajes Imperativos.

Los lenguajes imperativos, son aquellos en el que el control del programa pasa siempre a la siguiente línea del programa salvo que se le ordene lo contrario.

El programador debe marcar en el programa, mediante el número de la línea o mediante una etiqueta el flujo de la ejecución de las instrucciones. El modo de funcionamiento del ordenador aparece de forma clara al programador.

Si partimos de un lenguaje de este tipo o incluso de un nivel todavía más bajo como puede ser el lenguaje máquina o un lenguaje ensamblador para la contrucción de un sistema experto hay que realizar la totalidad de las tareas expuestas en secciones anteriores.

Las ventajas que presentan los lenguajes imperativos en la construcción de los SE's, son las siguentes:

- Flexibilidad total
- Conocimiento muy generalizado de los lenguajes tanto a nivel teórico como práctico, por lo que se ahorra tiempo en el aprendizaje.
- Eficiencia máxima.

Los inconvenientes son:

- Desarrollos muy largos, costosos y complejos.
- Los resultados finales son poco reutilisables en otras aplicaciones.
- Los programas son difíciles de leer y de modificar.

Por todas estas razones raramente se parte desde este nivel, a no ser que las especificaciones del sistema así lo requieran, se desee una gran optimización en el producto final, se busque unos fines didácticos o simplemente una satisfacción personal.

Los lenguajes imperativos que se vienen utilizando con mayor frecuencia son:

- Basic, por su sencillez y por ser, tal vez el lenguaje de programación más conocido.
- Pascal, por su estructuración en sus programas .
- Lenguaje C, cuyo uso se ha generalizado en los últimos años debido a su portabilidad y a su alto rendimiento. Como ejemplo del uso de C, podemos comentar el uso de CLISP (C Language Integrated Production System) por parte de la sección de Inteligencia Artificial del Centro Espacial Johnson de la NASA.

4.3.1.2 Lenguajes Funcionales

Los lenguajes funcionales, o también llamados aplicativos son aquellos en el que el flujo del programa viene marcado por las necesidades que aparecen al evaluar una función.

El programador solamente tiene que indicar el orden de evaluación de las funciones, pero no es necesario que se preocupe de referir su posición física dentro del programa.

En los lenguajes funcionales, el control de tipo secuencial, ha sido sustituido por cuatro pasos de evaluación, similares a los que se tiene cuando se quiere resolver una función matemática.

Esta filosofía va impregnando poco a poco a los lenguajes de tipo imperativo, en los que van apareciendo instrucciones que permiten estructuras de control funcionales, como pueden ser: Las subrutinas, los procedimientos, etc. que posibilitan procesos funcionales como es la recursión. Resumiendo el proceso :

- Análisis de la función a evaluar.
- Busqueda de la primera subfunción.
- Evaluación de la subfunción.
- Retorno a la función anterior o valor final de la evaluación sí se encuentra en la primera función.

Como se ve el proceso es por definición recursivo.

Los programas se evaluan es decir, a la función (que está compuesta en general de subfunciones) se le aplican los datos de entrada obteniendose unos resultados, y es por está razón que resulta más propio hablar de evaluación de un programa que de ejecutar o correr un programa .

Si partimos en el desarrollo de un sistema experto de un lenguaje funcional nos ahorramos solamente el sitema de tratamiento simbólico, gracias a la potencia de la estructura de datos que poseen (lista o sucesión de funciones). por otra parte la realización del motor de inferencia resulta más sencilla a partir de un control funcional que apartir de un control de tipo imperativo.

Las ventajas que tiene la utilización de un lenguaje funcional para la construción de un sistema experto son:

- Poseen sistemas de tratamiento simbólico.
- Sencillez en la construcción de motores de inferencia.
- Sencillo de aprender y de utilizar.

Los inconvenientes que presentan son:

- Es caro por los requerimientos mínimos que necesita para funcionar correctamente (necesita gran capacidad de memoria, siendo aconsejable el empleo de memoria virtual y velocidades de ejecución muy altas).
- Poco eficaz sobre arquitecturas tradicionales.
- Entornos cerrados (poca integración con otros lenguajes y entornos).
- Por definición, pueden crecer con facilidad por lo que no existen dos versiones del lenguaje iguales, ni tan siquiera si están inspiradas en un mismo estandar.

Los lenguajes más funcionales son: LISP y LOGO.

4.3.1.2.1 LISP

El LISP (List Processing, Lenguaje de Programación de Listas) fue creado por John McCarthy del MIT en el año 1960, es por tanto el más veterano de los lenguajes de programación que se utilizan en aplicaciones de I.A., y uno de los primeros lenguajes de programación de alto nivel. El LISP permite un potente manejo de listas, una gran

flexibilidad y extensibilidad (esta última característica a propiciado la aparición de infinidad de versiones dificultando su normalización), una sintaxis muy sencilla (paréntesis) lo que lo hace muy fácil de aprender y da gran seguridad en las tareas de programación, permite la recursión, existe una uniformidad entre los datos y las funciones, es muy dialogante o interactivo y es último es muy modular.

Veamos ahora algunas características del lenguaje.

En LISP existen dos elementos fundamentales:

- El átomo o dato elemental es la unidad más pequeña del lenguaje y puede ser un número (entero, racional, real o complejo) o un símbolo. El átomo es indivisible, si bien puede tener propiedades, ejemplo: Pino.
- La lista o dato compuesto, es un conjunto de átomos o de listas encerrados entre paréntesis y separados por comas. Al primer elemento de las listas se le llama "nombre de la función" mientras que al resto se les llama "parámetros" o "atributos" . Un ejemplo de lista: (árbol, pino, encino) .

A los átomos y a las listas se les llama de forma conjunta **"expresiones simbólicas"** .

LISP puede evaluar átomos que si son numéricos lo hacen así mismos y si son simbólicos lo hacen a su valor (lógica de orden 0). En cuanto a la evaluación de las listas se toma el primer elemento de lista como nombre de la función y el resto como parámetros.

Toda evaluación de LISP devuelve siempre un valor como resultado obtenido aunque este carezca de sentido. Una nueva función se define mediante "la función primitiva" DEFUN.

Existen muchas más funciones primitivas o ya definidas en el sistema de las cuales hay que destacar:

- CAR da como resultado de su evaluación el primer elemento de la lista por ejemplo: (CAR QUOTE(pino,encino))→pino .
- CDR da como resultado de su evaluación el resto o cola de la lista por ejemplo: (CRD QUOTE(pino,encino))→encino.
- CONS da como resultado de su evaluación la unión de un átomo a otro átomo, un átomo a una lista o una lista a otra lista por ejemplo: (CONS QUOTE(olivo) QUOTE(pino,encino))→(olivo, pino, encino).

Un predicado es una función que da como resultado al ser evaluada los átomos TRUE (verdadero) o NIL (falso).

Algunos de los predicados más comunes son:

- ATOM que evalua si evalua un átomo o no por ejemplo: (ATOM, pino)→verdad
- EQUAL que evalua la igualdad de dos expresiones por ejemplo:

  (EQUAL, pino, encino)→falso.

Dos características muy prácticas del lenguaje para realización de un sistema experto es la función primitiva COND o condición IF o si y THEN o entonces (en la formulación de reglas) y la facilidad con la que las estructuras en árbol se pueden describir las siguientes listas.

Un ejemplo de regla en LISP:

```
(REGLA 1
    (IF     (NOMBRE_A ES_PADRE_DE   NOMBRE_B)
            (NOMBRE_B ES_PADRE_DE   NOMBRE_C))
    (THEN   (NOMBRE_A ES ABUELO_DE  NOMBRE_C)))
```

Debido a las caraterísticas del LISP muchos lenguajes se escriben como extensiones de él, (por ejemplo MICROPLANNER, CONNIVER, SCHEME,...) y también muchos entornos de desarrollo (por ejemplo KEE,...).

Existen tres grandes grupos de versiones del lenguaje que son: el MacLISP del MIT, el InterLISP de Xerox y el Common LISP estandar propuesto por el gobierno americano.

4.3.1.3 Lenguajes Orientados Al Objeto

Los lenguajes orientados a los objetos se caracterizan por que no existe distinción entre los procedimientos y los datos. Los programas están formados por los objetos que son a la vez los procedimientos (también llamados: métodos, procedimientos locales o

comportamientos, que pueden ser por ejemplos los algoritmos) y los datos (también llamados : datos locales o facetas).

La estructuración del programa en objetos es la base fundamental de su gran modularidad.

Las acciones que realiza el programa son realmente mensajes que se envían a los objetos. Cada objeto interpreta el mensaje que le llega.

Un objeto es la particularización de una clase, heredando en este proceso las propiedades de la clase, molde o prototipo, que lo ha originado. Debido a este proceso generativo de los objetos, éstos estan estructurados jerárquicamente en clases y subclases.

Por ejemplo el objeto "pino" hereda las propiedades de su antecesor que puede ser árbol.

Los objetos tienen una serie de propiedades que les caracterizan y que se llaman atributos.

Por ejemplo es objeto "Documento Nacional de Identidad" Tiene una serie de atributos como son: "nombre","primer apellido".

Cada atributo posee una serie de facetas como pueden ser los valores permitidos, los valores por defecto, etc.

La posibilidad de la toma de valores por defecto, es muy importante ya que es una de las características que posee el razonamiento humano.

Los objetos pueden estar en tres estados: Activo es decir que estan en la línea de hipótesis, semiactivo si ha sido propuesto como hipótesis pero no de forma suficiente clara e inactivo cuando no se considera o se ha realizado como hipótesis.

La programación en un lenguaje orientado a objetos se realiza siguiendo estos pasos:

- Identificar los objetos que aparecen a lo largo del problema y en la solución.
- Clasificar los objetos por sus semejanzas y diferencias.
- Redactar los mensajes que interaccionan a los objetos.
- Implantar los métodos o procedimientos en los correspondientes objetos.

El control en un lenguaje orientado a objetos es en pocas palabras el siguiente: entran los datos, se propone una serie de hipótesis, se ordenan las hipótesis, se verifican, si se han creado nuevas hipótesis se vuelve otra vez al segundo punto y si no queda una más por verificar se propone una solución en base a las hipótesis válidas.

Este sistema de control es fácilmente convertible en un motor de inferencia; por otra parte se puede representar en una forma híbrida marco-regla de producción, y todo ello lo acerca a las necesidades de programación de un sistema experto.

Además de forma dinámica, es decir durante la evaluación del programa se pueden crear y modificar los objetos, siendo, está propiedad como ya hemos comentado importante para el aprendizaje de los programas. La tarea de construir un sistema experto se reduce notablemente si hacemos uso de un lenguaje orientado a objetos.

Algunas de las inconvenientes a la hora de desarrollar expertos son:

- Poco eficaz (por ejemplo las variables no están declaradas por lo que se deben comprobar el tipo de variable que intervienen en cada momento).
- Filosofía de programación diferente a la tradicional.

Los lenguajes orientados a los objetos, si bien normalmente incluyen otros tipos de representación del conocimiento de forma que si bien pierden "pureza" se gana flexibilidad. Vamos a ver algunos de ellos a continuación:

- ***DUCK*** Desarrollado por McDermott, está orientado a la representación mediante reglas de producción.
- ***FLAVORS*** Lenguaje orientado a objetos desarrollado por Moon del MIT. Disponible sobre máquinas simbólicas.
- ***FRL*** está orientado a la representación en forma de marcos desarrollado por Roberts y Goldstein del MIT.
- ***KEOPS*** es un lenguaje orientado a objeto.
- ***KOOL*** que además de estar orientado a los objetos lo está a los marcos y reglas de producción siendo prácticamente un sistema de desarrollo al incluir también: Motor de Inferencia de orden 1 con encadenamiento hacia adelante, hacia atrás y mixto, metarreglas, ponderación de reglas, interface con el lenguaje LISP. Está escrito en LISP.
- ***XLISP*** es un intérprete que realmente es una extensión del lenguaje LISP orientada a los objetos. Esta escrito en lenguaje C. Este lenguaje tiene una característica muy importante que es, que es de libre difusión. Está disponible para : PDP-11,VAX-11.
- ***SMALLTALK*** además de las características de los lenguajes orientados a los objetos es muy modular y flexible, contiene una gran economía de conceptos (objeto, clase,

herencia y mensaje) y viene acompañado de un entorno de programación o conjunto de útiles.

- C++ Es clasificado como un lenguaje híbrido orientado a objetos, en contraste con uno puro u ortodoxo, debido a que se sustenta en un lenguaje orientado a procedimientos más tradicionales. Viene acompañado de un entorno de programación o conjunto de útiles.

Otros lenguajes orientados a objetos son:

OBJECT PASCAL, NEON, ACT, FRAIL, etc.

4.3.1.4 Lenguajes Declarativos

Los lenguajes declarativos son aquellos en los que solamente hay que indicarle al programa el objetivo que queremos demostrar, especificando en el programa el universo sobre el que debe demostrarlo y las reglas que puede utilizar, es decir se especifica el "que" pero no el "como".

Para que el lenguaje sea capaz de enfrentar el problema es imprescindible que incorpore un motor de inferencia que controle el proceso de demostración. Los lenguajes declarativos más conocidos son el PROLOG y el OPS. Los lenguajes declarativos cuentan con:

- Un sistema de representación del conocimiento en forma de reglas de producción.
- Un Motor de Inferencia de orden 1 comunmente, que es el propio intérprete o compilador, basado en la unificación, la búsqueda en profundidad y la marcha hacia atrás en el caso del PROLOG; y en la comparación de patrones, la estrategia de resolución de conflictos en el OPS5. Permiten el encadenamiento hacia adelante o guiado por datos y el encadenamiento hacia atrás o guiado por objetivos de las reglas.
- Un sistema sencillo de diálogo.

Así la realización de un sistema experto en PROLOG o en OPS5 se reduce a la presentación y a los modulos de justificación de los resultados y explicación del proceso.

Otras ventajas de los lenguajes declarativos para la construcción de SE's son:

- Lenguajes muy compactos, al no necesitar ninguna estructura de control.

Como inconvenientes a la hora de contruir sistemas expertos tenemos:

- En versiones de lenguajes no actualizadas, las operaciones de E/S en estos lenguajes son algo complejas de programar, pero son posibles.
- En algunos casos el Motor de Inferencia es fijo o no parametrizable.
- Sistema de Representación del Conocimiento único.
- Carencia de entornos, son lenguajes cerrados.
- Poco eficaz sobre arquitecturas tradicionales.
- Poco fiable al no aparecer explícito el comportamiento del programa, pues el Motor de Inferencia, es decir, la estructura de control es implícita.

En resumen, en este nivel seguimos teniendo todavía un lenguaje de programación con toda flexibilidad que esto representa, muy potente al poder trabajar con predicados y además cuenta con un Motor de Inferencia.

4.3.1.4.1 PROLOG

El PROLOG (PROgramming in LOGic) fue creado basado en el principio de resolución automática .

El elemento fundamental de PROLOG son los predicados, que es aquello que se afirma o se dice de un sujeto, y están compuestos por un nombre y una serie de atributos por ejemplo:

abuelo(Nombre_A, Nombre_B).

que indica que el abuelo de Nombre_A es Nombre_B.

Los predicados pueden ser creados por el programador o estándares, es decir ya definidos por el lenguaje.

Los predicados creados pueden ser "autodefinidos" que contienen información acerca del sujeto sin que para ello dependa de ninguna condición. Si esta información es inmutable se les denomina "hechos" (por ejemplo: fact (1,1). este enso es un "hecho" puesto que no depende de ninguna variable y siempre se mantiene constante) y si puede variar "dato" (por ejemplo: max (A,B) if A>B. en este caso se cumple la regla siempre que A sea mayor que B y estos parámetros son variables, es decir dependen quizá de otra regla).

Los predicados definidos por reglas que también se les llama reglas son aquellos en que dependen de otros predicados para que sean ciertos o no.

La estructura de una regla en PROLOG es:

```
OBJETIVO        if      ANTECEDENTE 1 [OR:AND]
                        ANTECEDENTE 2 [OR:AND]
                                :
                        ANTECEDENTE n [OR:AND]
```

Un ejemplo es la siguiente regla:

```
abuelo(NOMBRE_A, NOMBRE_C) if
    padre(NOMBRE_A, NOMBRE_B)
    and padre(NOMBRE_B, NOMBRE_C)
    or
    madre(NOMBRE_A, NOMBRE_B)
    and madre(NOMBRE_B. NOMBRE_C).
```

Cuando se cumplen todas las condiciones en base a otras reglas y/o afirmaciones que forman el universo del programa se dice que el predicado se ha deducido o con otras palabras que es cierto.

PROLOG admite dos tipos de control en programa sobre su proceso de demostración que son:

- Búsqueda de todas las soluciones posibles, que se consigue con la marcha atrás.
- Restricción en el número o tipo de soluciones y en las reglas utilizadas que se consiguen con el corte de la marcha atrás.

El proceso de deducción se realiza mediante la unificación cuyo funcionamiento a grandes rasgos es el siguiente:

- Una variable libre puede ser unificada por un término, pasando a ser una variable particularizada para los siguientes términos de la regla.
- Una constante puede ser unificada con ella misma o con una variable libre.
- Una variable particularizada puede unificarse con una variable libre o con ella misma.

- Un hecho puede unificarse con otro hecho compuesto, si tienen el mismo functor y el mismo número de argumentos, unificándose cada argumento según los tres primeros puntos vistos.

PROLOG unificará de forma secuencial los hechos y las reglas del primero al último, y en una regla procederá de izquierda a derecha, hasta que demuestre el objetivo, si no puede demostrarlo se parará el proceso de unificación dando como resultado el valor verdadero.

Cuando PROLOG encuentra un hecho que resulta ser falso, busca otro que sea cierto, si esto no es posible, vuelve marcha atrás, hasta el nudo o bifurcación donde había tomado esa solución y vuelve a intentarlo, y así cuantas veces haga falta, hasta agotar todas las posibilidades.

Existen dos predicados de control que son FAIL y CUT (!). El uso abusivo de estos predicados de control convierte al PROLOG en un lenguaje funcional y en el peor de los casos imperativo, por lo que son un buen indicador de la aplicación correcta del lenguaje PROLOG.

PROLOG también inicia una marcha atrás si encuentra el predicado instalado "fail", aunque ya haya demostrado todos los subojetivos.

La utlidad de este predicado está en llamar explícitamente a la marcha atrás porque se desean todas las soluciones posibles que pueden generarse.

Otra utilidad está precisamente en detenerse el proceso para lo cual se combina con el ! (corte) (como se verá más adelante) de tal forma que el predicado:

cualquier_predicado, !, fail.

causará que al llegar el curso normal de la demostración del objetivo cualquier_predicado hasta el fail se abandone definitivamente dando fallo.

Cuando el PROLOG ha iniciado una marcha atrás por alguna de estas razones:

- porque haya fracasado
- forzado por un fail
- forzado de forma implícita por un Goal

nunca podrá superar un corte o cut ("!").

Cuando PROLOG avanza no tiene ninguna influencia el predicado ! en el proceso.

El predicado ! puede ponerse en cualquier punto de la parte de acciones y condiciones de una regla (parte derecha), como cualquier otro predicado.

Una de las ventajas que tiene PROLOG es que al estar basado en la lógica matemática conceptualmente no presenta ningún problema.

Entre los inconvenientes se encuentra en resolución de algoritmos, en los procesos de entrada y salida de datos.

Como resumen el PROLOG es un lenguaje con un sistema de representación del conocimiento en forma de reglas de producción, restringidas a clausulas de Horn, con un motor de inferencia de orden 1, empleando la unificación, con encadenamiento hacia atrás con marcha atrás y búsqueda exhaustiva en profundidad, por lo que es completo. Existen tres grandes estandares de PROLOG que son: PROLOG-10, M-PROLOG, PROLOG II.

En el apéndice "B" se muestra un ejemplo de un programa en PROLOG aplicado en la resolución de un problema que requiere SE's.

4.3.2 Conclusiones

Las características más importantes de los lenguajes de IA es la habilidad de construir grandes y complejas estructuras de datos que contengan símbolos, sin el compromiso de tamaño o tipo de elementos en éstas, y sin la necesidad explicita de administrar el almacenamiento de memoria. Lo que es aún más interesante es la forma en como muchos problemas tienen características similares a los paradigmas de la IA. Estos, son usualmente problemas pobremente definidos que requieren de una gran interacción con un usuario humano. Tales problemas tienen necesidades de ambiente parecidas a los sistemas de IA. Los tipos de ambientes requeridos para estos problemas consisten de herramientas poderosas que asisten al programador ampliando sus potencialidades. Esto puede ser logrado reduciendo la necesidad del mismo de recordar detalles, para que se concentre directamente en el problema.

4.4 *SELECCIÓN DE SOFTWARE PARA SISTEMAS EXPERTOS*

Las herramientas comerciales para construir sistemas expertos tienen sus orígenes en sistemas de IA desarrollados por investigadores de universidades y centros de investigación. Ha sido reportado que estas herramientas hacen posible el desarrollo de sistemas expertos en un tiempo considerablemente menor que el que sería requerido con el uso de lenguajes tradicionales en el área, tales como LISP y PROLOG.

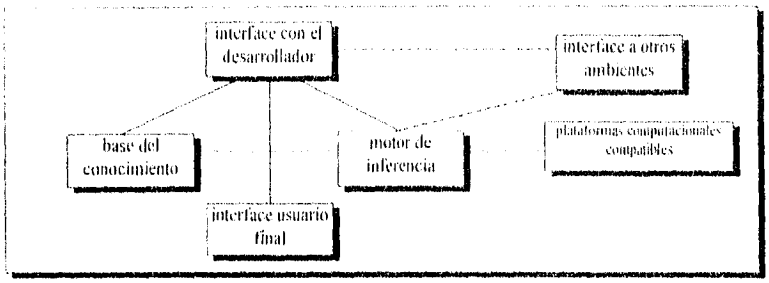
El núcleo de un sistema experto consiste de una base de conocimiento y el motor de inferencia que opera sobre ésta para obtener la respuesta o solución. Desde luego, la utilización de tal sistema requiere de una interface con el usuario. Para facilitar el desarrollo de sistemas expertos, una HECSE (Herramienta de Construcción de Sistemas Expertos) debe incluir también una interface con el desarrollador, así:

- La base de conocimiento puede ser construida para una aplicacion de dominio particular
- Se puede desarrollar una interface de usuario apropiada y
- Se podría incorporar instrucciones especiales para el motor de inferencia (sistema de razonamiento), que sean requeridas para un dominio en particular.

El carácter y calidad de estas interfaces son las dos principales diferencias entre las HECSE comerciales y las utilizadas para investigación. Otras estructuras importantes de estas herramientas son:

- Interfaces con otros programas y bases de datos.
- Las computadoras sobre las cuales corren, no sólo en desarrollo sino también en producción.

A continuación se muestra la estructura de una herramienta de construcción de SE's en donde las líneas sólidas representan relaciones básicas y las líneas punteadas representan aspectos relacionados.



4.4.1 Atributos de las HECSE's

En la tabla siguiente se puden observar los diferentes atributos de una HECSE y su importancia en las implicidad de construcción de sistemas expertos [William, 1978]) para diferentes aplicaciones. El asterisco (*) indica que un atributo es muy saludable en la construcción de la aplicación indicada. Una círculo vacío (O) indica que se trata de un contribuyente poco significativo y una celda vacía quiere decir que dicho elemento no ofrece una contribución importante. Desde luego la evaluación anterior es subjetiva puesto que depende de la óptica y habilidad del desarrollador. Algunas funciones pueden ser descompuestas de forma tal que se consigan estructuras similares a las que demanda el problema.

	1	2	3	4	5	6	7	8	9
INFERENCIA									
Encadenamiento hacia atras	*	O	O		O	O	O	O	O
Enc. al enc. atras y razonamiento hacia adelante	O	*	*	*	*	*	O	*	*
Encadenamiento hacia atras y hacia adelante	*	*	*	O	O	O	*	*	O
Razonamiento hipotetico	*	*	*	O	O	O	O	*	
Orientado a objetos	O	O	O	*	O	O	O	O	*
Pizarrón	*	*	*	O	O	O	O	*	
Inducción	*	O	O			O	O		O

DESCRIPCION DE LOS OBJETOS									
Marcos	*	0	*	0	0	0	0	*	0
Marcos con Herencia	*	0	*	0	0	0	0	*	0
Objetos	0	0	0	*	0	0	0	0	*
Pares parámetro valor	0	0	0		0	0			0
Lógica	0	0	0	0	0	0	0	0	0
Reglas	0	0	0	0	0	0	0	0	0
Certidumbres	*	*	0	0	0	0	0	0	0
ACCIONES									
Reglas	*	*	*	*	*	*	*	*	*
Ejemplos	*	0	0	0		0	0		0
Lógica	*	*	*	*	0	0	0	0	0
Mensajes	0	0	*	*	0	0	0	*	*
Procedimientos	0	0	*	*	*	0	0	*	*

1. CLASIFICACION Y DIAGNOSTICO
2.-INTERPRETACION Y ANALISIS DE DATOS
3.-DISEÑO Y SINTESIS
4.-SIMULACION Y PREDICCION
5.-MONITOREO
6.-CONSULTORIA
7.-ASISTENCIA INTELIGENTE
8.-PLANEACION Y CALENDARIZACION
9. CONTROL

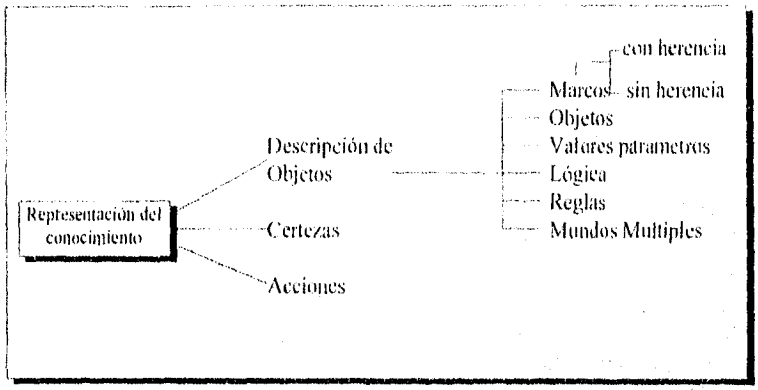
4.4.2 Formas de Representación de Conocimiento

Existen tres aspectos de la representación de conocimiento fundamentales en estas herramientas, la figura muestra los métodos más comunes que acontinuación se describen:

Descripción de objetos: Conocimiento declarativo como hechos y aseveraciones.

Certidumbres y

Acciones.



Una forma comunmente usada para representar objetos son los marcos, con o sin herencia (la herencia permite que las bases de conocimientos sean organizadas como arreglos jerárquicos de marcos que heredan información de los marcos superiores. Así, un mecanismo de herencia provee de una forma de inferencia). Los marcos son datos tabulares que permiten organizar representaciones de objetos y situaciones prototípicas. Un marco tiene nichos que son llenados con datos sobre objetos y relaciones apropiadas a la situación. Un paradigma de programación que utiliza marcos es el conocido como programación orientada a objetos. Existen paradigmas que utilizan objetos que incorporan facilidades para el paso de mensajes entre si mismos; anclados a estos procedimiento existen otros procedimientos que pueden ser activados a la recepción de estos mensajes. El conocimiento

declarativo también puede ser representado por parametros de valor, por el uso de notación lógica y con algunas extensiones, por reglas.

Las acciones cambian una situación y/o modifican la base de datos. Las acciones son más comunmente representadas por reglas. Estas pueden ser agrupadas en módulos para facilitar su mantenimiento y agilizar el acceso. Las acciones también pueden ser representadas en términos de ejemplos, los cuales indican las conclusiones o decisiones logradas. Los ejemplos son una forma particularmente deseable para facilitar la adquisición de conocimiento, y los sistemas inductivos explotan esta situación. Los ejemplos son más faciles de extraer de un experto que las reglas, y pueden frecuentemente ser una forma natural al dominio de conocimiento. Las acciones pueden ser expresadas también en notación lógica, que es una forma de escribir reglas. Finalmente, las acciones pueden ser expresadas como procedimientos extraídos ya sea por mensajes (en Programación Orientada a Objetos), cambios observados en la base de datos por demonios (los demonios son procedimientos que monitorean una situación y responden ejecutando una acción cuando sus condiciones de activación aparecen).

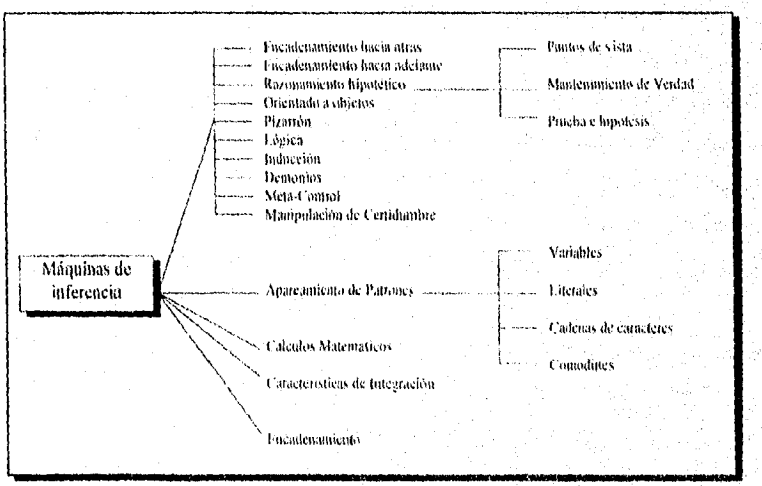
Además de la representación de objetos y acciones, se deben considerar el grado en el cual el conocimiento o los datos son correctamente certificados. Así, muchas HECSE's tiene facilidades para representar certezas.

La forma más común de hacerlo es incorporando factores de confiabilidad que son una derivación del método usado en Mycin. Lógica difusa y probabilidad son también usados a veces.

Una foma alternativa de manejar incertidumbres son los modelos profundos (los cuales son modelos causales) del sistema que pueden ser realmente construidos con la herramienta en cuestión, como una ayuda en la base-modelo de razonamiento. Finalmente, el tamaño del sistema (por ejemplo medido con el número de reglas necesarias) puede ser de crítica importancia, dado que quizá existan efectos importantes en los requerimientos de memoria, administración del sistema y ejecución del mismo.

4.4.3 Motor de Inferencia

Existen múltiples formas alternativas de como un HECSE ejecuta las inferencias como se observa en la figura que sigue. El método mas usual es el de clasificación, el cual es apropiado a situaciones en las cuales existe un número fijo de posibles soluciones. Un grupo de conclusiones hipotetizadas de éstas, son evaluadas para saber en que forma están apoyadas por la evidencia. Esta evaluación es efectuada por encadenamiento hacia atrás a través de reglas IF-THEN(antecedente-consecuente), empezando con las reglas que tienen las conclusiones hipotetizadas como su consecuente. Las reglas buscadas, son entonces aquellas que tienen como consecuente las acciones que apoyan el antecedente en la regla de la conclusión hipotetizada. Este proceso continúa recursivamente hasta que la conclusión es completamente apoyada o negada, o hasta que un camino sin salida sea alcanzado. Si alguno de los últimos dos eventos ocurre, pueden ser probadas hipótesis adicionales hasta alcanzar alguna conclusión o que el proceso termine. A esta estrategia de resolución se le conoce también como dirigida por la meta.



Encadenamiento hacia adelante es otra estrategia que comienza con datos de entrada o con la situación global presente en la base de datos, para ser apareados con los antecedentes de cada una de las reglas involucradas, y así determinar la aplicabilidad de una o varias de ellas a las circunstancias presentes (La situación global presente en la base de datos, es usualmente representada por un conjunto de atributos y sus valores asociados). Una de las reglas apareadas se selecciona y su consecuente es utilizado para agregar información a la base de datos, o para activar un procedimiento que cambie la situación global. La estrategia prosigue recursivamente, terminando ya sea cuando una conclusión es alcanzada o cuando todas las posibles reglas aparentes son agotadas. Las combinaciones de encadenamientos hacia atrás y hacia adelante suelen resultar muy efectivos en la mayoría de los problemas.

El razonamiento hacia adelante puede ser considerado como una forma más general del encadenamiento hacia adelante. En esta estrategia, el control puede ser instrumentado con reglas dirigidas por datos, o procedimientos dirigidos por datos (demonios).

El razonamiento hipotético se refiere a estrategias de solución en las cuales se pueden hacer suposiciones, para habilitar la ejecución de procedimientos de búsqueda. Mas tarde, a lo largo de la ruta de búsqueda, puede encontrarse que ciertas suposiciones son inválidas por lo que tienen que ser retractadas. Este razonamiento nomonotónico (razonamiento en el cual conclusiones o datos pueden ser retractados, a la luz de nuevas informaciones) se puede manipular en diferentes formas. Un método que reduce la dificultad de cómputo es el de acarrear múltiples soluciones en paralelo y descartar las inapropiadas de acuerdo como la evidencia que las contradiga vaya apareciendo. Esta aproximacion es referida como puntos de vista, contextos y mundos en diferentes herramientas. Otra forma es el mantener seguimiento de las suposiciones que apoyan la ruta de búsqueda actual y retroceder al punto apropiado de la rama cuando esta ruta es invalidada.

La programación orientada a objetos es una estrategia en la cual la información acerca de los objetos y los procedimientos apropiados a tales objetos, son agrupados juntos en una estructura de datos similar a un marco. Esta información es actualizada por mensajes que son enviados al objeto, por un control, central o por otro objeto. Este método es

particularmente apropiado para situaciones que involucran simultáneamente agrupamiento de objetos distintos entre sí, y para procesamiento de señales en tiempo real.

El método de inferencia de pizarrón está asociado con un grupo de sistemas expertos cooperativos, que se comunican compartiendo información de una estructura de datos común,conocida como pizarrón. Se puede utilizar un mecanismo de agenda para facilitar el control del desarrollo de la solución sobre el pizarrón.

El método de lógica es otra forma que las HECSE's pueden presentar como mecanismo de inferencia. Es un método orientado a la prueba de teoremas mediante la unificación de variables, que básicamente consiste en hacer que dos elementos aparescan idénticamente. Las implementaciones de lógica más comunes se encuentran en versiones de lenguajes de programación lógica como PROLOG, que utilizan métodos exhaustivos de búsqueda en profundidad.

Un importante mecanismo de inferencia que encontramos en algunas herramientas, es aquel que permite generar reglas o árboles de decisión inductivamente a partir de ejemplos. Los expertos humanos frecuentemente articulan su experiencia de mejor forma con ejemplos que con reglas. Así, las técnicas de aprendizaje inductivo, son frecuentemente métodos ideales de adquisición de conocimiento para construcciones muy rápidas de prototipos, especialmente cuando los ejemplos pueden simplemente ser expresados en la forma de una conclusión asociada con una simple colección de atributos. Los constructores humanos del sistema experto resultante pueden entonces, refinarlo interativamente criticando y modificando los resultados inductivamente producidos.

La inferencia inductiva usualmente procede iniciando con uno de los parámetros de entrada y buscando en un árbol un número mínimo de decisiones necesarias para alcanzar una conclusión. Este árbol de profundidad mínima es generado apartir de recorrer todos los parámetros como posibles nodos iniciales. Usando una aproximación teórica de información es posible seleccionar el orden de éstos para ser usados por los nodos restantes y para determinar cuales de éstas parámetros son superfluos. Una aproximacion teórica de informacion es aquella que escoge la solución que requiere de la mínima cantidad de información para representarla. La profundidad del árbol es con frecuencia, relativamernte pequeña y así numerosos ejemplos pueden resolverse en árboles anchos y poco profundos.

Algunas herramientas ofrecen diversas formas de mecanismos de inferencia ó procedimientos de búsqueda. En sistemas construidos con tales herramientas, se permite al desarrollador controlar la estrategia de inferencia. Dicho control es conocido como meta-control. Una forma de instrumentar el meta-contol, es usando bloques de control, los cuales son procedimientos genéricos que le indican al sistema los siguientes pasos a tomar en una situación determinada, de manera que la búsqueda se vea reducida, permitiendo que un gran número de reglas pueda ser manejado, sin que el espacio de búsqueda se convierta en combinatoriamente explosivo.

Puesto que la certidumbre de los datos, reglas y procedimientos es usualmente menor al 100%, muchos sistemas incorporan facilidades de manejo de incertidumbres. Así se tienen varios métodos para combinar reglas inciertas e información para determinar la certidumbre del valor del resultado.

El apareamiento de patrones es frecuentemente necesario para mecanizar técnicas de inferencia, particularmente para aparear los antecedentes de las reglas con el estado del sistema. La sofisticación de la técnica de apareamiento de patrones afecta la capacidad del sistema. Los tipos de apareamiento varían desde aparear cadenas de caracteres idénticas a variables, literales y "comodines" hasta apareamientos aproximados que pudiesen servir como razonamiento analógico.

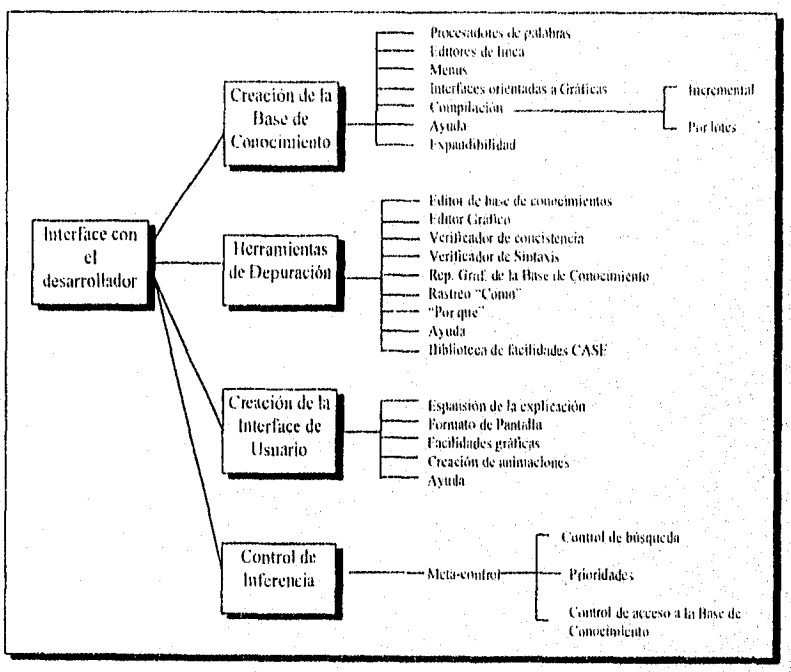
Otras capacidades de las HECSE's difieren de herramienta a herramienta. Algunos motores de inferencia ofrecen rápidas y sofisticadas capacidades de cálculo matemático. Una de las más valiosas habilidades provistas por una máquina de inferencia es la de manipular bases de conocimiento modularizadas o soluciones modularizadas permitiendo acceso y encadenamiento de estos módulos según sea necesario.

Otra importante consideración en una herramienta es el grado de integración de sus distintas características. Una integración completa es deseable de manera tal que todas las capacidades puedan ser explotadas, si es necesario, en la solución de un sólo problema. Por ejemplo, en el caso de HECSE's que incorporan representación de objetos y reglas con encadenamiento hacia adelante y hacia atras, es deseable que los desarrolladores de sistemas expertos tengan la posibilidad de mezclar estos tipos de encadenamiento de las

reglas libremente y puedan razonar con la información contenida en los objetos cuando sea necesario.

4.4.4 La Interface con el Desarrollador

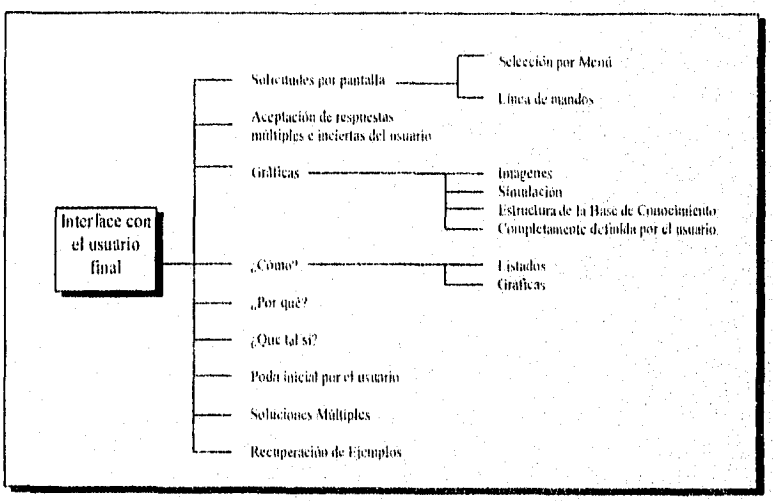
Las distintas herramientas ofrecen diferentes niveles y facilidades al desarrollador de sistemas expertos para utilizar el molde del sistema. Las más sencillas son Shells dentro de los cuales el conocimiento es insertado en una estructura específica.Otras más sofisticadas son generalmente más difíciles de aprender, pero permiten al desarrollador un número de opciones más amplio de representación de conocimiento, estrategias de inferencia y modos de interface con el usuario final. Así mismo existen varios niveles de depuración como se muestra a continuación en la figura.



Otra característica importante es la verdadera modularidad que la herramienta otorga. Muchos "shells" por ejemplo, presentan al menos una forma de agrupar reglas que permite distribuir mejor el conocimiento en la fase de desarrollo y facilitan el mantenimiento del sistema. Sin embargo, al igual que en los sistemas tradicionales, es frecuente que varios programadores e ingenieros del conocimiento necesiten trabajar en módulos separados, sobre todo cuando se trabaja con varios expertos de diferentes dominios para sistemas interdisciplinarios, por lo que es deseable que exista algún mecanismo para finalmente unir estos módulos con cierta naturalidad.

4.4.5 La Interface con el Usuario Final

Una vez que el sistema experto ha sido construido, su aplicabilidad depende en gran parte de la interface del usuario final (a continuación se muestra una figura de las interfaces con el usuario final).



Dado que muchos sistemas expertos son realmente "asistentes inteligentes", la interface de usuario debe construirse de forma que permita un diálogo interactivo. Este diálogo y/o

entrada inicial debe frecuentemente de aparecer al usuario como un menu o arreglo estructurado de datos, de manera que le permita contestar las preguntas originadas por el sistema. En algunos casos, para aumentar la flexibilidad de éste se hace que el sistema acepte múltiples e inciertas respuestas del usuario, y que aún asi obtenga una conclusión congruente. En sistemas más sofisticados, frecuentemente se usan gráficas para mostrar al usuario la línea de razonamiento cuando éste pregunta "¿como?"; en sistemas más simples, una lista con las reglas que apoyan a la conclusión puede ser suficiente. Las HECSE's frecuentemente responden a preguntas del usuario del tipo "¿Porqué necesitas esta información?", desplegando la regla para la cual dichos datos son requeridos. La habilidad del sistema de responder a las preguntas "¿como?" y "¿porqué?" del usuario es importante para incrementar la credibilidad de éste en el sistema y su habilidad para tomar decisiones.

Otras facilidades frecuentemente encontradas en las HECSE's son aquellas que:

Permiten al usuario escoger valores alternativos de los parámetros y observar los resultados sobre todo el sistema (estas facilidades manejan preguntas del tipo "¿Que tal si?").

Permiten efectuar una poda inicial de la línea de cuestionamientode manera que el sistema no investigue en áreas que el usuario considera irrelevantes o inecesarias.

Dan capacidad de guiar, dar y recobrar ejemplos para futuras consideraciones o uso.

Algunas herramientas muy sofisticadas frecuentemente incluyen gráficas y facilidades de simulación de modo interactivo, que incrementan el entendimiento del usuario final y la forma de control de la representación del sistema. Por sobretodo, la interface del usuario final tiene que ser "amigable" si el sistema ha de ser aceptado.

4.4.6 Consideraciones con respecto a los Lenguajes de Programación

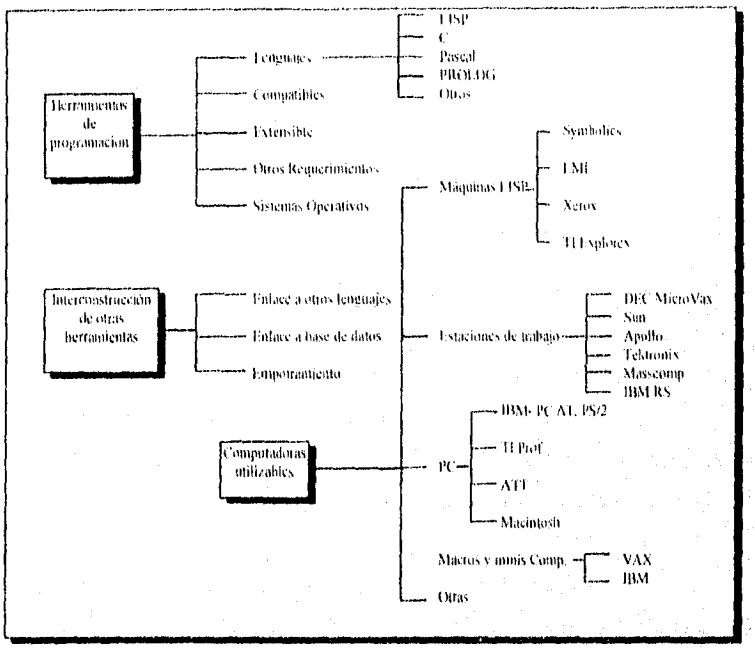
Además de la estructura y de los paradigmas provistos por una herramienta, una caracteristica de gran importancia, es el lenguaje de programación sobre el cual está escrita. El lenguaje determina si el sistema experto es compilable, y en que forma, ya sea incrementalmente o en forma batch. El ser compilable reduce los requerimientos de memoria e incrementa la velocidad de ejecución asi como la de desarrollo del sistema experto.

En general las herramientas más sofisticadas han sido escritas en Lisp. Sin embargo, estas herramientas está empezando a ser escritas en lenguajes como 'C', para incrementar la velocidad, reducir los requerimientos de memoria y para garantizar su disponibilidad en diferentes tipos de computadoras. Sin embargo, algunos nuevos intentos de mejorar los intérpretes de LISP (por ejemplo GCLISP), podrían reducir la velocidad y las necesidades de memoria, logrando ventajas similares a las de 'C'.

El usuario puede extender las herramientas escritas en Lisp escribiendo funciones adicionales en este lenguaje. Esto también es cierto para otros lenguajes como PROLOG y Pascal. Algunas extensiones similares se pueden lograr en herramientas que tienen conexión con otros lenguajes, programas y demás medios de información. En algunos casos, el sistema experto generado por la herramienta está completamente empotrado dentro de otro, lo cual incrementa su autonomía.

Las herramientas que operan en diversas Plataformas computacionales, dependen primeramente del lenguaje en el cual han sido escritas y del sistema operativo, y en segunda instancia, de la memoria, las capacidades de procesamiento y las facilidades gráficas de la computadora en cuestión. La tendencia de construir shells para sistemas expertos disponibles en PC's se debe en parte al incremento de las capacidades en general de estas computadoras, pero también a la posibilidad de escribir herramientas en lenguajes más

rápidos como C, y a la habilidad de modularizar las bases de conocimiento. Como se mencionó anteriormente, tal modularización require de la descomposición del problema en subproblemas, y de la facilidad de proveer mecanismos de encadenamiento apropiados entre los módulos durante la ejecución, la siguiente figura muestra aspectos de las herramientas relativos la Hardware y al Software.



4.4.7 Funciones

Las Capacidades Funcionales en Inteligencia Artificial son la siguientes:

- *Clasificación*
- *Interpretación de mediciones*: Selección de hipótesis basada en evidencia

- *Diagnóstico*: Selección de mediciones e Interpretación (frecuentemente involucra modelos organizacionales de sistemas y su comportamiento).
- *Síntesis y Diseño*: Obtención de restricciones y guías
- *Predicción*: Elaboración de pronósticos
- *Consultoría*: Cómo dar consejos
- *Asistencia Inteligente*: Obtención de herramientas de decisión
- *Calendarización*: Secuencialización de tareas en tiempo, deacuerdo a restricciones de recursos
- *Planeación*: Selecciones complejas que afectan a otras selecciones
- *Monitoreo*: Ejecución de acciones en tiempo real
- *Control*: Control de procesos
- *Asimilación de Información*: Evaluación de circunstancias
- *Descubrimiento*: Generación de nuevas relaciones o conceptos
- *Depuración*: Proveer de acciones correctivas
- *Razonamiento Basado en Ejemplos*: La fuente de la mayoría de las reglas

De las primeras consideraciones para evaluar herramientas para desarrollo de SEs son las aplicaciones de las funciones rápidamente construíbles con una HECSE's. Un vistazo a las aplicaciones más usuales se presenta a continuación.

Clasificación. Una de las funciones más comunes realizadas por un sistema experto es la de clasificación. La clasificación se refiere a seleccionar una pregunta de un conjunto fijo de alternativas en base a la información de entrada.

En seguida se muestran algunas de las subcategorías de la clasificación.

Interpretación de mediciones. Esta se refiere a la selección de hipótesis efectuadas sobre la base de los datos de medición e información corolaria.

Diagnóstico. Aquí el sistema no sólo interpreta datos para determinar la dificultad, sino que también busca datos adicionales cuando éstos son necesarios para dirigir su línea de razonamiento.

Depuración y/o Reparación. Estas funciones se refieren a las acciones necesarias o mediciones recomendadas para corregir una situación adversa que ha sido diagnosticada.

Consejero de Usuarios. Un sistema experto como frontal de un programa, puede ser de gran utilidad para un usuario inexperto. Tal sistema depende tanto de las metas del usuario como de la situación actual, para sugerir que hacer al respecto. Así, el consejo evoluciona con el estado del mundo y sus cambios. Usar consejeros puede ser útil también como guía de usuarios en procedimientos de dominios o áreas de aplicación ajenas a ellos.

Las funciones aplicativas mencionadas, pueden usar dos tipos de razonamiento:

razonamiento superficial y razonamiento profundo. En el razonamiento superficial, no se hace ningún modelo del sistema: la forma de atacar el problema es escribiendo una colección de reglas: cada afirmación de cada regla acerca de cierta situación garantiza cierta respuesta o conclusión (Esta relación de situación-respuesta se describe con reglas heurísticas adquiridas de la experiencia). En el razonamiento profundo, el sistema barre sobre un modelo causal o estructural del dominio de interés, para ayudar a alcanzar una conclusión. Así, el sistema que emplea modelos profundos es potencialmente más capaz y puede degradarse más gradualmente que aquellos que se apoyan en razonamiento superficial.

Síntesis y Diseña. La síntesis y diseño se refieren a configurar un sistema sobre la base de un conjunto de alternativas posibles. El sistema experto agrega restricciones que el sistema debe usar como guía para cumplir los objetivos del usuario.

Asistencia Inteligente. Aquí se hace énfasis en tener un sistema que dependiendo de las necesidades del usuario, pueda dar un consejo, proporcione información, o desempeñe varias subtareas.

Predicción. Se refiere a predecir lo que ocurrir en el futuro en base a la información actual. Esta predicción puede depender de la experiencia solamente, o puede involucrar el uso de modelos y fórmulas. Mientras más dinámico es un sistema, más simuladores usa en sus predicciones.

Calendarización. Se refiere al ordenamiento en tiempo de un conjunto dado de tareas, que deben ser ejecutadas con los recursos disponibles sin interferir unas con otras.

Planeación. Es la selección de una serie de acciones tomadas de un complejo conjunto de alternativas para alcanzar los objetivos del usuario. Esto es más complejo que calendarizar un conjunto dado de tareas ya que en muchos casos, las restricciones de tiempo y recursos,

no permiten que todas los objetivos sean cubiertos. En estos casos lo que se busca es que el resultado sea de lo mejor posible.

Monitoreo. Esta función indica una observación de una situación progresiva para su predicción o para alertar al usuario si su curso es normal o anormal. Las aplicaciones típicas se dan en procesos industriales, atención de pacientes en recuperación,en acciones de defensa e incluso en vuelos espaciales.

Control. El control es una combinación de monitoreo de un sistema y de ejecución de acciones apropiadas tendientes a corregir el curso del proceso de acuerdo a una meta. En muchos casos, tales como la operación de vehículos o máquinas, la respuesta tolerable del sistema debe producirse en milisegundos o lapsos más pequeños. A los sistemas que trabajan bajo este régimen se les conoce como sistemas de tiempo real.

Procesamiento de Información. Un sistema que ejecuta tal función puede tomar información y devolverla en una nueva forma de organización o sintetizarla. Una aplicación puede ser la determinación inductiva de un árbol de decisión a partir de ejemplos, y otra por ejemplo, puede tratarse de la evaluación de las circunstancias del mercado en base a los datos disponibles einformación adicional.

Descubrimiento. El descubrimiento es similar al procesamiento de información excepto que hace énfasis en encontrar nuevas relaciones, órdenes, o conceptos. Esta sigue siendo un área de investigación; algunos ejemplos incluyen búsqueda de nuevos conceptos matemáticos y de leyes elementales de la física.

Existen otras funciones tales como aprendizaje, que pueden ser directamente clasificadas o descompuestas de acuerdo a algunas de las ya enumeradas. Así por ejemplo, el diseño y alguna otra función podrían ser separadas como subtareas de clasificación.

4.4.8 Consideraciones respecto al desarrollo

En cuanto al ambiente de desarrollo es muy importante conocer:

La transportabilidad de la herramienta. Es común encontrar algunos shells que no dejan compilar fácilmente la aplicación desarrollada. En algunos casos, ciertas funciones de gran poder deben ser reescritas antes de intentar compilar (proceso que además no siempre este

garantizado). También se da el caso de herramientas que aparentemente corren en cualquier plataforma, pero que sin embargo las versiones de estaciones de trabajo o de minicomputadoras no son directamente transportables a ambientes de plataformas de escritorio, lo cual no resulta sorprendente, dado que el poder computacional tampoco es comparable.

El tiempo requerido para el aprendizaje de ciertas herramientas, sobre todo de sistemas mayores, que ofrecen ambientes realmente poderosos, requieren para su total aprovechamiento de varios cursos de capacitación o de fuertes interacciones con desarrolladores de sistemas tradicionales. En general, son pocas las compañías o instituciones que están dispuestas a pagar el costo de la curva de aprendizaje. Muchas veces se encuentran aplicaciones de juguete en ambientes muy poderosos, que fueron necesarias desarrollar durante algunas semanas para adiestrar al personal del área.

La flexibilidad de las interfaces hacia otras herramientas y ambientes, en ocasiones pueden complicar el desarrollo integral del sistema. En aplicaciones ambiciosas, a veces no se cuenta con herramientas que presenten clara conectividad hacia sistemas tradicionales ya en operación y el desarrollar estas interfaces implica un alto costo en recursos computacionales y humanos.

las facilidades de depuración que ofrece unas herramientas, pueden disminuir o agrandar el tiempo de desarrollo. Las facilidades de seguimiento de procesos a veces no son todo lo completas que se quisiera. Por ejemplo, algunas muestran la línea de razonamiento indicando que reglas han sido disparadas y el valor de los parámetros, pero no permiten hacer cambios de valores durante la ejecución. Otras no permiten ejecutar módulos aislados temporalmente del sistema, sino que hay que correr todo el proceso para rastrear las últimas etapas.

La integrabilidad sobre las plataformas computacionales disponibles es crítica, ya que la mayoría de las herramientas hacen uso exhaustivo de los recursos computacionales y muchas veces este consumo de recursos crece proporcionalmente al tamaño de la base de conocimiento en desarrollo, lo cual generalmente, trae como consecuencia que las máquinas procesen más lentamente cada vez. Definitivamente el poder de la herramienta se ve determinado por el hardware que la soporte, así que aunque se desarrolle sobre máquinas

muy grandes es necesario considerar si el desempeño del sistema será comparablemente similar, cuando se ejecute sobre la plataforma computacional del usuario final.

La magnitud del problema a resolver es un factor importante. Algunos gerentes de sistemas, de acuerdo a la estrategia de su empresa se dedican a desarrollar SE's de pequeñas dimensiones, que muchas veces se escogen aplicando "la prueba del teléfono". Esta prueba consiste en determinar si el problema en cuestión puede ser resuelto satisfactoriamente por el experto vía telefónica. Desde el punto de vista académico o de investigación, esto puede estar generando grandes cantidades de aplicaciones de juguete, para las cuales no se requieren shells muy poderosos y que se pueden programar en casi cualquier herramienta de computadora de escritorio que se respete. Sin embargo la experiencia de empresas como Dupont [Figebaum 1988] ha mostrado que las aplicaciones pequeñas o de "juguete", muchas veces desarrolladas por los propios usuarios, han aportado considerables ahorros.

4.4.9 Consideraciones respecto a la producción

Considerando el ambiente en el cual el sistema va a trabajar o a producir, es indispensable tomar en cuenta:

La plataforma computacional sobre la cual va a correr. En algunos casos el problema a resolver puede ser único en todo una comunidad, que cuenta con una máquina central muy poderosa, por lo que el sistema esta apoyado en importantes recursos computacionales a los cuales todo usuario de dicha comunidad tiene acceso. Sin embargo, se da el caso también en el que el sistema pretende hacer llegar la experiencia de algunos o alguno, a lugares donde no existen cuantiosos recursos, o en donde diponer de ellos sería muy costoso y no solo eso, sino que ademas existen empresas cuya estrategia ha distribuído un enorme poder computacional por todas sus oficinas en computadoras decescritorio, individualmente no muy poderosas.

El presupuesto de inversión es muchas veces uno de los recursos más escasos, lo cual implica que aún habiendo encontrado una herramienta con características idóneas para enfrentar el problema, sea necesario descartarla por su alto costo, o por requerir

infraestructuras muy caras. Algunos autores definen a un ingeniero como la persona que es capaz de hacer con $1 U.S.D., lo que alguien común y corriente haría con $10 USD.

El tiempo que se dispone para desarrollo pude determinar la herramienta, dado que algunas de ellas ofrecen posibilidades inagotables, es común que el programador tenga que resolver módulos básicos, que ni siquiera se encuentran preconstruidos para lo cual se requiere de muchas líneas de programación adicional, algo que no es deseable cuando los resultados que se esperan son a corto plazo.

La magnitud del problema a resolver es un buen indicador, tratándose de una tecnología tan novedosa. Es común que todos los problemas sean propuestos para ser resueltos por un SE, sin embargo, como ha sido mencionado en secciones anteriores, existe todo un proceso de selección del problema, que fundamentalmente debe dimensionarlo y clasificarlo como un problema con un tipo de razonamiento homogéneo. Dicho en otras palabras, se deben buscar problemas que interactuén con objetos de un solo tipo o de características similares de manera que un solo esquema de representación sea útil a todo el problema. Hay que recordar que las herramientas híbridas no son muy comunes, son costosas y generalmente consumen muchos recursos computacionales.

Dependiendo del tipo de usuario final al que el sistema le de servicio, es posible requerir o prescindir de algunas facilidades de la herramienta. En algunos lugares se desarrollan SE's que sirven de apoyo a expertos, por lo que seguramente la interface de explicación tendrá que ser muy robusta y confiable para ganarse la aceptación y confianza del usuario. En otros casos se puede requerir de un sistema de tiempo real para monitoreo de algún proceso, en el que sea más importante el contol del ambiente que las posibles explicaciones. A veces con una bitácora de eventos es más que suficiente. Existen herramientas que carecen por completo de interfaces de explicación, pero que ofrecen toda la flexibilidad para desarrollar una propia y a cambio ofrecen buenas conexiones con procesos numéricos.

Si trabajar en línea o en procesos por lotes, puede ser un indicador del tipo de interface de usuario que se debe ofrecer. Algunas herramientas proveen de vistosas interfaces, lo que desde luego tiene un costo asociado, en dinero y elementos computacionales; en los procesos por lotes, muchas veces se espera que el sistema permita conclusiones que no necesitan de sofisticadas explicaciones o interrogatorios muy especializados. En cambio es

probable que en sistemas personalizados de apoyo algún equipo de ventas, la interface necesite ser lo más amigable posible y que permita sesiones en forma conversacional.

4.4.10 Conclusiones

En esta sección se descubrieron los diferentes paradigmas involucrados en problemas afines a la IA y, más concretamente, en la construcción de sistemas expertos. Asimismo, se discutieron las facilidades que las distintas herramientas ofrecen, de manera que ahora es posible hacer un resumen de los criterios a seguir en la selección.

Sin pretender seguir un órden, los principales elementos aconsiderar al diseñar un SE son los siguientes:

Tipo de problema a resolver
Diagnosis
Monitoreo
Planeación
Diseño
Representación de Conocimiento
Métodos de solución de problemas
Control
Encadenamiento hacia adelante y hacia atrás
Factores de certidumbre
Inducción
Programación Orientada a Objetos (POO)
Interface de Desarrollo
Editores orientados a la sintáxis
Menús
Gráficos
Explicación
Simulación Gráfica
Lenguaje de implementación

compilable (si/no)
comunicación con otros lenguajes
comunicación a otros paquetes (e.g.bases de datos)
sistema operativo
Plataformas computacionales
Requerimientos de memoria
Costo
desarrollo del sistema
ejecución del sistema
Capacidad (e.g. número de reglas sin degradarse)
Desempeño
Capacidad
Facilidad de aprendizaje
Portabilidad
Documentación
Entrenamiento
Soporte Técnico

4.5 Algunas HECSE's en el Mercado

A continuación se presentan algunos atributos de las diferentes herramientas que se pueden conseguir comercialmente.

4.5.1 ART

Comentarios

ART es una herramienta versátil que incorpora un sofisticado banco de trabajo. Corre en computadoras avanzadas y estaciones de trabajo como Symbolics, LMI, TI, Apollo, y VAX. En esta forma de razonamiento, varias soluciones se acarrean en paralelo hasta que las restricciones son violadas o hasta que se encuentran mejores soluciones. En tales puntos, las soluciones inapropiadas son descartadas. ART proporciona interfaces gráficas para

visualizar tanto sus puntos de vista como sus redes de esquemas (marcos). Es fundamentalmente un sistema de encadenamiento hacia adelante, con un sofisticado apareamiento de patrones (instanciación) definido por el usuario; ésta es una versión fortificada de un esquema de indexación derivado de OPS5 [Forgy, 1980]. La Programación Orientada a Objetos está disponible en torno a la asignación de procedimientos en los objetos. ART tiene herramientas gráficas flexibles en las cuales se pueden crear interfaces y simulaciones gráficas. Esta herramienta fue diseñada para trabajar con un desempeño muy cercano a tiempo real, para lograrlo, compila su base de marcos así como su conocimiento relacional en afirmaciones de tipo lógico.

Aplicaciones:

Planificación, calendarización, simulación, generación de configuraciones, y diseño.

Representación de Conocimiento:

Reglas de producción estructuradas, marcos con herencia, objetos, procedimientos.

Métodos de inferencia:

Encadenamiento hacia atrás y hacia adelante, meta-control, lógica, cálculos matemáticos, contextos.

Interface con el desarrollador:

Procesador de palabras, editor de la base de conocimiento, menús, reportes gráficos de la base de conocimiento, rastreo de herencias, utilerías gráficas para construcción de interfaces de usuario final, formateo de pantallas, capacidad de simulación gráfica, interface de explicación (¿como?, ¿porqué?), expansión de explicaciones, ayuda en línea, ayuda en sintéxis, extensibilidad de la herramienta.

Lenguaje Nativo:

LISP, aunque ya existe una versión en lenguaje C

Interfaces a Bases de Datos:

IMS, DB2, VSAM.

Plataformas computacionales:

IBM PC/AT, PS2 y compatibles, IBMmainframes, SUN 3 & 4, DEC, Apollo, TI Explorer I & II, Symbolicsserie 3600.

Sistema operativo:

MS-DOS, MVS, VMS, Unix.

Requerimientos de memoria:

Para PC's :

memoria convencional 640 K

memoria extendida 1 a 2 MB

disco duro 8 MB

Precios:

de $8,000 a 100,000.00 USD

4.5.2 ESE

Comentarios:

El Expert System Environment de IBM, tiene un extenso conjunto de herramientas diseñadas para contruir amplias bases de conocimiento, en grandes computadoras (mainframes). Por su estructura, permite a los desarrolladores y usuarios finales aprender a usarlo en corto tiempo. Una vez que la base de conocimientos ha sido creada, se puede usar en múltiples formas:

interactivamente en VM/CMS Y MVS/TSO

empotrado en MVS, CICS/VS, IMS/VS

descargado en un diskette para consulta de usuarios finales en PC

Aplicaciones:

Problemas de Diagnóstico, Planeación y Toma deDecisiones.

Representación de Conocimiento:

Reglas de producción

Métodos de inferencia:

encadenamiento hacia atr s y hacia adelante

Interface con el desarrollador:

Permite modularizar la base de conocimiento, rompiéndola en bases pequeñas que pueden ser integradas en una sola o ser llamadas dinámicamente durante la ejecución. Tiene pantallas por "default", que puden ser reescritas por el desarrollador; además, tiene editores especializados que facilitan la actualización y modificación de la base de conocimiento.

Puede manejar gráficas, permite hacer rastreos del proceso de inferencia, tiene capacidad de deshacer y rehacer consultas, ofrece ayuda en línea y puede hacer representaciones gráficas de la jerarquía del conocimiento. Así mismo soporta preguntas de ¿cómo? y ¿por qué? a modo de explicación. Puede ser consultado en lenguaje natural gracias a su módulo NLS, en inglés, japonés y otros idiomas.

Lenguaje Nativo:

Pascal

Interface a Bases de Datos:

DB2, SQL/DS, VSAM.

Plataformas computacionales:

IBM PC/AT,PS/2 en modo de ejecución,IBM sistemas mayores (mainframes) para desarrollo y ejecución.

Sistema operativo:

MS-DOS, MVS/XA, MVS/TSO, VM/CMS, MVS/CICS,MVS/IMS.

Requerimientos de memoria:

En MVS y VM: 50 cilindros del DASD3380, 4MB para consulta y 6MB para desarrollo en TSO, en PC"s (sólo consulta):

10 MB en disco duro y 640K de RAM

Precios:

de $24,000 a 60,000 USD

4.5.3 FRAMEKIT

Comentarios:

FrameKit es una herramienta de representación de conocimiento basada en marcos. Está escrita en CommonLisp y provee los mecanismos básicos de los marcos: herencia, demonios, y perspectivas. Fue desarrollado por Jaime Carbonell e implementado en FranzLisp para una computadora VAX. Ha sido utilizado en diversas aplicaciones que abarcan desde sistemas expertos hasta sistemas de captura y almacenamiento de información médica. Posteriormente FrameKit fue transportado a CommonLisp (Robert Joseph) el cual ya podía ser instalado en máquinas IBM PC compatibles.

Los marcos en FrameKit se escriben como simples listas anidadas. Los demonios se pueden disparar indirectamente por alguna acción en un nicho de un marco y pueden ser descritos con cualquier expresión válida en Lisp. Las perspectivas se utilizan para almacenar la misma información sobre el mismo marco, para resolver diferentes problemas.

Representación de Conocimiento:

Marcos

Mecanismo de inferencia:

Herencia y relaciones

Lenguaje Nativo:

CommonLisp

Plataformas computacionales:

Vax e IBM PC compatibles

Sistema operativo:

VMS y MS-DOS

4.5.4 GOLDWORKS II

Comentarios:

Es una de las herramientas más grandes disponibles para computadoras personales. Corre en ambiente de desarrollo deCommon Lisp de GCLISP sobre plataformas 386 y en Macintoch II con Alegro Common Lisp y con 8 MB de RAM completo. GoldWorksII tiene varias características que lo colocan en la categoría de las herramientas avanzadas de desarrollo de sistemas expertos. Las nuevas utilerías del paquete incluyen gráficas dinámicas que permiten crear imágenes activas como válvulas y calibradores, además de interfaces dinámicas interactivas para el usuario como revisadores de marcos y reglas, y proposiciones orientados a gráficas. También exhibe una mejor integración con C, y un analizador lexicográfico de código ASCII, que puede leer archivos de texto formateados y crear objetos apropiados para guardar la información en el paquete. Otro factor importante es su velocidad de procesamiento, ya que puede ser compilado usando el compilador de GCLISP. Por tratarse de una herramienta híbrida es un buen candidato para casi cualquier tipo de aplicación avanzada.

Aplicaciones:

Análisis, Diagnóstico, calendarización, y clasificación.

Representación de Conocimiento:

Marcos, demonios, objetos con herencia, reglas bidireccionales, encadenamiento hacia adelante y hacia atrás.

Interface con el desarrollador:

Editor GMACS, módulos de explicación (¿porqué?, ¿como?), menús, pantallas revisadoras, árboles localizadores.

Lenguaje Nativo:

Common Lisp

Interfaces con Bases de Datos:

dBase III y Lotus 1-2-3

Plataformas computacionales:

Sun, Macintosh II, IMB PS/2, AT.386.

Sistema operativo:

MS-DOS y Macintosh

Requerimientos de memoria:

6MB RAM para compilar 4MB RAM enversión intérprete.

Precios:

De $ 7,500.00 a $ 10,000.00 USD.

4.5.5 KBMS

Comentarios:

La funcionalidad de KBMS está basada en 4 metodologías de razonamiento claves, lo cual permite atacar una gran variedad de problemas comúnes en las organizaciones. Además tiene absoluta compatibilidad con ambientes de procesamiento de sistemas mayores de IBM. KBMS puede operar bajo CICS, TSO, y otros ambientes comúnes a los sistemas mayores. Con su módulo INTELLECT, permite al usuario invocar y consultar a la base de conocimientos en inglés convencional.

Aplicaciones:

Análisis de inventarios, autorizaciones de crédito, calendadrización, planeación, análisis corporativo, etc.

Representación de Conocimiento:

Objetos temporales y permanentes, reglas de producción.

Métodos de inferencia:

encadenamiento hacia atrás y hacia adelante, demonios, razonamiento hipotético, herencia y enlances entre objetos (Programación Orientada a Objetos).

Interface con el desarrollador :

dirigido por menú, generación de pantallas y reportes, break, rastreo y dump, generador de interrogaciones SQL, conexión a base de datos (DB2), agrupamiento de reglas (paquetes).

Lenguaje Nativo:

Lenguaje C

Interface a Bases de Datos:

DB2, SQL/DB, IDMS, VSAM, ADAVAS, IMS.

Plataformas computacionales:

IBM PS/2, IBM Mainframes

Sistema operativo:

MS-DOS, OS2, MVS/XA, CICS, TSO, etc.

Requerimientos de memoria:

4 MB

Precios:

$ 9,500 USD. para PC's, de $90,000 a $225,000 USD para mainframe

4.5.6 KEE

Comentarios:

Corre en computadoras de IA avanzadas, es el ambiente de programación más comunmente usado para la construcción de sofisticados sistemas expertos. Los aspectos importantes de KEE son los multicaracterísticos ambientes de desarrollo y la interfaces de

usuario final, las cuales incorporan ventanas, menús y gráficas. KEE contiene un sistema sofisticado de marcos que permite el modelaje jerárquico de objetos y permite múltiples formas de herencia. También ofrece una variedad de métodos de razonamiento y análisis, incluyendo programación orientada a objetos, reglas de encadenamiento hacia adelante y hacia atrás, razonamiento hipotético, un lenguaje de lógica de predicados, y demonios. Tiene una arquitectura abierta que permite métodos de inferencia definidos por el usuario, ároles de herencia, operaciones lógicas, funciones y gráficas. Tiene tambien un conjunto grande de interfaces gráficas que son controladas por el usuario/desarrollador, que incluyen facilidades de simulación gráfica.

Aplicacinnes:

Diagnóstico, monitoreo, procesos de control en tiempo real, planeación, diseño y simulación.

Representación de Conocimiento:

Reglas de producción estructuradas, marcos con herencia, Objetos, lógica y procedimientos

Métodns de inferencia:

Encadenamiento hacia atras y hacia adelante, demonios, meta-control, cálculos matemáticos y contextos.

Interface con el desarrollador:

Editor de línea, menues, editor de base de conocimiemo, verificador de concistencia, reporte gráfico de la base de conocimiento, rastreo de herencias, utilerías gráficas para interfaces de usuario final, interfaces de explicación (¿porqué?, ¿como?), expansión de explicaciones, ayuda en línea, ayuda en sintaxis, extendibilidad de la herramienta.

Lenguaje Nativo:

Golden Common Lisp

Interface a Bases de Datos:

DB2, IMS/DB

Plataformas computacionales:

Symbolics, LMI, TI Explorer, VAX,386 PC.

Sistema operativo:

IBM/KEE:mainframe, MVS

Requerimientos de memoria:

10MB RAM, 100MB HD.

Precios:

$9,000.00 a $100,000.00 USD.

4.5.7 KNOWLEDGE CRAFT

Comentarios:

Es una herramienta híbrida basada en marcos, que soporta herencias definidas por el usuario. Es una integración de la versión de Carnegie Mellon de OPS5, Prolog y del lenguaje de representación de marcos KRL. Es un conjunto de herramientas de alta productividad para ingenieros del conocimiento experimentados y constructores de sistemas de IA. Los marcos son utilizados para el conocimiento declarativo; el conocimiento de procedimientos es instrumentado a través de demonios. KC es capaz de operar con razonamiento hipotético (no monotónico) cuando se utilizan contextos. La búsqueda es definida por el usuario. Una utilería de simulación gráfica se encuentra disponible (Simulation Craft). Fue diseñado para operar en tiempo real, para procesos de clasificación exclusiva es muy poderosa.

Aplicaciones:

Planeación, calendarización, control de procesos.

Representación de Conocimiento:

Reglas estructuradas, marcos conherencia, Objetos, lógica, procedimientos.

Métodos de inferencia:

Encadenamiento hacia atrás y hacia adelante, demonios, meta-control, lógica, cálculos matemáticos, contextos.

Interface con el desarrollador:

Editor de línea, Procesador de palabras, editor de base de conocimientos, representación gráfica de base de conocimiento, rastreo de herencias, utilerías para construcción de

interfaces de usuario final, interfaces de explicación.

Lenguaje Nativo:

Golden Common Lisp

Interface a Bases de Datos:

Oracle, dBase

Plataformas computacionales:

IBM 386, PS/2 en producción, IBM mainframes para desarrollo y producción, Sun MicroVAx, Symbolics, Explorer.

Sistema operativo:

UNIX, MS-DOS, MVS/XA, MVS/TSO, VM/CMS,MVS/CICS, MVS/IMS.

Requerimientos de memoria:

de 8 a 16 MB recomendados

Precios:

de $10,000 a 100,000 USD.

4.5.8 M1

Comentarios:

Es una herramienta basada en PC's enfocada a solución de problemas. Es básicamente un sistema de encadenamiento hacia atras diseñado para clasificación. Incluye capacidad de ordenamientos de meta-nivel para dirigir razonamiento hacia adelante. Esta escrito en C, lo que le permite ser integrado fácilmente a sistemas convencionales. Su principal inconveniente es que no permite una verdadera descripción de objetos y por lo tanto no puede manejar modelos profundos. Sin embargo, M1 tiene un buen conjunto de herramientas de desarrollo, interfaces de usuario final, e interfaces para el desarrollador.

Aplicaciones:

Clasificación

Representación de Conocimiento:

Reglas de producción (limitado a 1000), factores de certidumbre

Método de inferencia:

Encadenamiento hacia atras, razonamiento hacia adelante, demonios, meta-control, cálculos matemáticos.

Interface con el desarrollador:

Procesador de palabras, menú,editor de base de conocimiento, utilerías para formateo de pantallas, interface de explicación (¿porque?, ¿como?), expansión de explicación, ayuda en línea, ayuda en sintaxis

Lenguaje Nativo:

Lenguaje C

Plataformas computacionales:

IBM PC

Sistema operativo:

MS-DOS

Precios:

$ 5000.00 USD.

4.5.9 NEXPERT OBJECT

Comentarios:

Es una poderosa herramienta basada en reglas, codificado en C, capaz de correr en una Macintosh con 512K deRAM, en una Mac Plus o en una IBM PC AT. Tiene posibilidades de edición comproables a las de herramientas diseñadas para correren sofisticadas máquinas de IA. El sistema permite al desarrollador agrupar reglas por categorías, de manera que pueden ser llamadas sólo cuando se necesitan. Nexpert Object, soporta reglas, variables y combinaciones de encadenamiento hacia atrás y adelante. El sistema puede automáticamente generar representaciones gráficas de las redes de reglas, las cuales indican como unas reglas se relacionan con otras. Se pueden generar redes similares para mostrar que reglas se han disparado en respuesta a una consulta en particular. Nexpert Object incluye facilidades de representación tanto de marcos con múltiples herencias, como de apareamiento de patrones de reglas, facilitando razonamientos profundos. Se trata de una herramienta sofisticada con enfoque hacia la representación gráfica, tanto de las bases de

conocimiento, como del proceso de razonamiento, lo que hace que las interfaces resulten comprensibles y naturales para el usuario final y para el desarrollador.

Aplicaciones:

Clasificación, diseño, planificación, calendarización, control de procesos.

Representación de Conocimiento:

Reglas estructuradas (limitado a 2000), marcos con herencia, procedimientos.

Métodos de inferencia:

Encadenamiento hacia atrás, hacia adelante, demonios, meta-control, cálculos matemáticos, razonamiento no monotónico.

Interface con el desarrollador:

Editor de línea, editor de base de conocimiento, menús, verificador de consistencia, representación gráfica de la base de conocimiento, rastreo de herencias, utilerías gráficas para construcción de interfaces de usuario final, utilerías para formateo de pantallas, capacidad de simulación gráfica, interfaces de explicación (¿porqué?, ¿cómo?), expansión de explicación, ayuda en línea, ayuda en sintáxis, extensibilidad.

Lenguaje Nativo:

Lenguaje C

Interface a Bases de Datos:

Oracle, Sybase, Ingres, Informix,Lotus 1-2-3, dBase III, SQL, RDB.

Plataformas computacionales:

Macintosh, IBM PC AT, VAXstations,Apollo, HP, IBM RT.

Sistema operativo:

MS-DOS, VMS

Requerimientos de memoria:

IBM: 1MB memoria expandida, MAC 2 MB

Precios :

de $5,000.00 a $8,000.00 USD.

4.5.10 PERSONAL CONSULTANT PLUS

Comentarios:

Es una herramienta avocada a proveer muchas de las características que se encuentran en sistemas más sofisticados, tales como KEE. PC Plus utiliza marcos, con herencia de atributos y reglas. Aún que el fabricante clama que PC+ tiene marcos, en realidad lo que maneja son formas de agrupación de reglas que son capaces de heredar valores a través de parámetros y variables, pero ciertamente no es posible hacer una clara definición de objetos como tradicionalmente se supondría utilizando marcos. Utiliza el encadenamiento hacia atrás derivado de Emycin. También incluye encadenamiento hacia adelante sin instanciación de variables. Además tiene un extenso conjunto de herramientas tanto para su desarrollo como para su ejecución, que lo hacen realmente "amigable". La versión (2.0) opera hasta con 2 MB de memoria extendida ó expandida; además soporta monitores IBM EGA y tiene acceso a DBASE III en IBM PC's. Existe incluso una versión para la TI Explorer y MicroExplorer.

Aplicaciones:

Clasificación

Representación de Conocimiento:

Reglas estructuradas, factores de certidumbre, Marcos con herencia, procedimientos.

Métodos de inferencia:

Encadenamiento hacia atrás y hacia adelante, demonios, meta-control, cálculos matemáticos.

Interface con el desarrollador:

Entrada en línea, editor de basede conocimiento, menúes, verificador de consistencia, rastreo de herencias, facilidades gráficas de format,o de pantallas, interface de explicación (¿porqué?, ¿cómo?), expansión de explicación, ayuda en línea, ayuda en sintáxis, expandibilidad.

Lenguaje Nativo:

Scheme (dialecto de Lisp)

Interface a Bases de Datos:

dBase, Lotus 1-2-3.

Plataformas computacionales:

IBM PC/AT/XT/PS/2, Explorer,MicroExplorer

Sistema operativo:

MS-DOS

Requerimientos de memoria:

2MB de memoria expandida o extendida

Precio:

$3,000 USD.

4.5.11 PICON

Comentarios:

Esté diseñado como un shell Orientado a Objetos, para desarrollo de sistemas expertos en tiempo real, así como para automatización de procesos industriales y otros procesos que son monitoreados con sensores, tales como los sistemas de navegación y los procesos bursátiles y financieros. PICON trabaja en máquinas Lisp como LMI Lambda Plus y la TI Explorer, las cuales combinan el poder del procesamiento simbólico del procesador LISP con la alta velocidad de procesamiento numerico y adquisición de datos del procesador MC68010. Ambos procesadores operan simultáneamente, lo que permite a PICON monitorear al sistema en tiempo real, detectar eventos sisgnificativos del proceso, diagnosticar problemas, y decidir la acción apropiada. El editor de iconos y gráficos de la herramienta permite al desarrollador con mínimo entrenamiento en IA construír y representar modelos profundos del proceso a ser automatizado. Las reglas son creadas a base de menús y con interfaces de lenguaje natural.

Aplicaciones:

Clasificación y control de procesos

Representación de Conocimiento:

Reglas estructuradas, marcos conherencia, programación orientada a objetos

Métodos de inferencia:

Encadenamiento hacia adelante y hacia atrás, demonios, modelación en tiempo., meta-control, cálculos matemáticos.

Interface con el desarrollador:

Procesador de palabras, ingreso de mandatos por línea, editor de base de conocimientos, menues,verificador de concistencia, representación gráfica de la base de conocimientos, rastreo de la inferencia, utilerías gráficas para construcción de interfaces de usuario final, simulación gráfica, formateador de pantallas, módulo de explicación (¿Cómo?, ¿Porqué?), ayuda sintáctica, extensibilida de la herramienta.

Lenguaje Nativo:

Zeta-Lisp, C (compilable)

Plataformas computacionales:

LMI, TI Explorer

Sistema operativo:

Requerimientos de memoria:

Precios:

$60,000 USD.

4.5.12 GOLDEN COMMON LISP DEVELOPER

Comentarios:

Es una variante de LISP y no puede ser accesado en 640K de RAM. La versión 2.6 tiene algunas diferencias con LISP en el manejo de las variables; sin embargo, versiones más recientes si parecen ser 100% compatibles. Puede manejar arreglos de una dimensión y funciones " struct " , equivalente a los registros en Pascal. Así mismo puede hacer grupos de Stacks (Pilas). Los componentes de GCLISP son el interprete de LISP, el ambiente de GCLISP, el editor GMACS y un tutorial de LISP.

Aplicaciones:

Ideal para quienes se inician en la Programación en Lisp.

Interface con el desarrollador:

Editor GMACS

Plataformas computacionales:

IBM PS/2, AT, XT o compatibles, DEC Rainbow, WANG

Sistema operativo:

MS-DOS, VMS.

Requerimientos de memoria:

2MB RAM, 4MB HD

Precios:

$495.00 USD

4.5.13 OPS5

Comentarios:

Es un intérprete de reglas; una herramienta para solución de problemas que utiliza una colección de reglas para actualizar datos en la memoria de trabajo, tal cual como se describe al respecto en la sección que trata a los sistemas de producción. OPS5 es un lenguaje usado para expresar pares condición-acción y el contenido de la memoria de producción. Existen opciones comercialmente más robustas como OPS83 que básicamente opera bajo la misma filosofía, solo que provee de algunas vetajas en cuanto a sus interfaces con lenguajes externos y bases de datos.

Aplicaciones:

Diseño (ejemplo XCON)

Lenguaje Nativo:

C o LISP (depende de la computadora)

Interface con el desarrollador:

Acceso a editores externos (versión en UNIX), exploración de la memoria de trabajo y del conjunto de reglas que pueden disparar, regreso a soluciones parciales anteriores.

Plataformas computacionales:

IBM PS/2, AT, XT o compatibles,SunII, Macintosh, VAX y MicroVax (VaxOPS5)

Sistema operativo:

MS-DOS, VMS.

Requerimientos de memoria:

640K RAM, 1MB HD

Precios :

$ 1,800.00 USD. VaxOPS5 $ 7,500.00 USD.

Conclusiones

En las secciones anteriores se han descrito brevemente algunas de las herramientas que se encuentran en el mercado. En realidad, si fuese necesario es posible conceptualizar un problema de manera que se pueda resolver con cualquier herramienta. Algunos estudiosos de la IA y los SE's dicen que,"torturando el conocimiento, éste puede ser metido en prácticamente cualquier representación", sin embargo es claro que mientras más adecuada sea la herramienta al problema en cuestión, mucho más expresiva y más eficiente sea la solución o soluciones. Generalizando se pueden hacer dos grandes clasificaciones de los criterios a considerar para seleccionar una herramienta:

- En relación al ambiente de producción.
- En relación al ambiente de desarrollo.

En la siguiente sección se discuten las características de las plataformas computacionales y como afectan al desempeño de las herramientas de programación.

4.6 SELECCIÓN DE HARDWARE PARA SISTEMAS EXPERTOS

4.6.1 Introducción

En teoría, IA podría hacerse en un ábaco. El ábaco sería dolorosamente lento y el operador estaría propenso a cometer errores, pero se podría hacer. Después de todo, la computadora lee los mismos ceros y unos del código binario independientemente de si procesa lenguaje natural o una nómina. Recientemente se podría decir que IA es software; el hardware sólo facilita la rapidez de ejecución y abarata los costos .

Sistema operativo:

MS-DOS, VMS.

Requerimientos de memoria:

640K RAM, 1MB HD

Precios :

$ 1,800.00 USD. VaxOPS5 $ 7,500.00 USD.

Conclusiones

En las secciones anteriores se han descrito brevemente algunas de las herramientas que se encuentran en el mercado. En realidad, si fuese necesario es posible conceptualizar un problema de manera que se pueda resolver con cualquier herramienta. Algunos estudiosos de la IA y los SE's dicen que,"torturando el conocimiento, éste puede ser metido en prácticamente cualquier representación", sin embargo es claro que mientras más adecuada sea la herramienta al problema en cuestión, mucho más expresiva y más eficiente sea la solución o soluciones. Generalizando se pueden hacer dos grandes clasificaciones de los criterios a considerar para seleccionar una herramienta:

- En relación al ambiente de producción.
- En relación al ambiente de desarrollo.

En la siguiente sección se discuten las características de las plataformas computacionales y como afectan al desempeño de las herramientas de programación.

4.6 SELECCIÓN DE HARDWARE PARA SISTEMAS EXPERTOS

4.6.1 Introducción

En teoría, IA podría hacerse en un ábaco. El ábaco sería dolorosamente lento y el operador estaría propenso a cometer errores, pero se podría hacer. Después de todo, la computadora lee los mismos ceros y unos del código binario independientemente de si procesa lenguaje natural o una nómina. Recientemente se podría decir que IA es software; el hardware sólo facilita la rapidez de ejecución y abarata los costos .

La IA es fundamentalmente un proceso simbólico. En los Estados Unidos el procesamiento simbólico esta ampliamente asociado con el lenguaje de programación Lisp. Arquitecturas computacionales especializadas han sido creadas para optimizar el desarrollo del lenguaje Lisp. Otros lenguajes comúnmente usados en IA son Small-Talk y PROLOG,y también el lenguaje C .

En las siguientes secciones se analizarán estrategias y detalles involucrados en la selección de la plataforma computacional hardware para su aplicación en IA. Como es sabido las estaciones de trabajo y las PC's más poderosas están dirigiendo el desarrollo de IA. Estas máquinas de 32 bits se están convirtiendo en máquinas cada véz más poderosas, sofisticadas, y baratas. Las plataformas para IA, fluctuan desde los mainframes hasta los controladores especializados (máquinas LISP), que son mucho más versátiles que las PC's y las estaciones de trabajo. Sin embargo, pocas instituciones u organizaciones comprarán uno de estos controladores, o un mainframe para apoyar la distribución o desarrollo de una aplicación de IA.

4.6.2 Factores que afectan la selección de Hardware

4.6.2.1 Arquitectura

Cualquiera puede comprar una plataforma de alto rendimiento por $80,000 USD. pero, ¿Se puede conseguir un buen desempeño por $20.000 USD?. El arte de la Ingeniería yace en el balance de los elementos para hacer coincidir un precio objetivo con la capacidad de ejecución. Los elementos importantes para los diseñadores de plataformas incluyen el CPU, la memoria dinámica, el almacenamiento masivo, los dispositivos de E/S, y la programación software.

El corazón de cualquier sistema es el CPU. El CPU trabaja en base a datos según las instrucciones de su programa. Aquí es donde el compilador trabaja en el código fuente del programador para producir instrucciones de maquina. Típicamente, el CPU también actúa como el director para todo el sistema, controlando cual subcomponente ejecuta cuales funciones y cuando.

Se han desarrollado tres filosofías de diseño de CPU's. El diseño más común para propósitos generales se llama "Complexinstruction Set Computer" CISC. Las máquinas de arquitectura de etiquetas (tagged architecture machines) que corren LISP son el segundo tipo. Y el tercero, es un concepto de diseño en plataformas comerciales llamado arquitectura RISC (computadoras de conjunto reducido de instrucciones, del inglés Reduced Instruction Set Computer).

El Intel "PENTIUM " y Motorola 68040 son ejemplos guías de la filosofía de CISC . Un tipo de circuitería especial es construida dentro del chip (encapsulado) para desempeñar casi cualquier función que un programador podría desear. La complejidad del juego de instrucciones CISC produce una operación interna más lenta y ocupa un espacio adicional en el *chip* para descifrar las instrucciones y la lógica de control que se podía utilizar para características de optimización del rendimiento como cachés más grandes y más registros.

Las PC's IBM, las series P/S 2 y compatibles, Apple Macintuch, y las workstations basados en el 68040 ó en el "PENTIUM PRO" son máquinas CISC.

La tagged architecture CPU es usada en las Máquinas clásicas de LISP. Ciertas funciones especiales por hardware son construidas en el interior para apoyar a LISP, insertando una etiqueta por hardware para cada dirección de memoria. Cuando el CPU asigna información a una dirección de memoria, al mismo tiempo escribe en la etiqueta tag el tipo de información que es. La etiqueta puede ser de tipo número entero integer, punto flotante floating point, ocaracter character data. Este apoyo de hardware hace verificaciones "libres" de tipo. (efectuados sin castigar la ejecucion).

RISC, el tipo más nuevo de CPU, delega parte de la responsabilidad de la ejecución al software, especialmente a los compiladores optimizadores. En realidad, la mayoría de las instrucciones del CPU son simples; agregar, mover y almacenar. La arquitectura RISC rebalancea el diseño del sistema ejecutando solo instrucciones básicas comunes pero haciéndolas más rápido que CISC. Los *chips* tipo RISC usualmente tiene menos de 128 instrucciones, en comparación a los 200 a 300 en los *chips CISC* típicos. Los *chips* tipo RISC ofrecen menos formatos de instrucciones y modos de acceso a memoria que los *chips* CISC, lo que resulta en un hardware de control más simple.

En teoria una maquina RISC debería correr una tarea más rápido que una CISC , porque cada instrucción requiere sólo un ciclo de CPU o clock para ejecutarse. Los criticos de RISC señalan que un compilador generará más instrucciones para manejar menos instrucciones complicadas.

4.6.2.2 Memoria

Aún el mejor diseño de CPU es inservible sin memoria, y la IA es reconocida por demandar grandes cantidades de memoria. Los símbolos estan fuertemente conectados a otros símbolos. Las conecciones entre símbolos requiren memoria, mientras más rica sea la conexión requeriré más memoria y mayor será la demanda en el sistema.

La memoria no sólo tiene que ser rápida, sino también abundante y a precio razonable. Los CPU's van a la cabeza en la competencia de velocidad por dólar. Una memoria rápida llamada static RAM que puede competir con los CPU's de hoy en día, ya se encuentra disponible, pero es demasiado costosa para ser utilizada en usos generales. El común de memoria llamado dynamic RAM, es usada casi exclusivamente para hacer un mejor balance de la relación precio/ejecución.

El espacio de direcciones de la máquina puede ser expandida más allá de la RAM asignada, utilizando memoria virtual. Esta, copia regiones de la memoria RAM llamadas páginas, hacia y desde el disco duro de la máquina.

Los diseños avanzados de sistemas de memoria utilizan los 3 elementos: RAM estático, RAM dinámico, y discos duros. El veloz RAM estático puede funcionar como memoria cache o buffer que permite al CPU tener acceso a los datos e instrucciones a la misma velocidad a la que los esté procesando. El RAM dinámico puede ser muy grande y económico. El disco duro puede rápidamente permutar páginas de memoria. También es posible usar un encapsulado (chip) independiente para el control de memoria, y para coordinar los 3 elementos.

Teniendo en cuenta que se requieren 64 MB en RAM para los "PENTIUMS PRO" y 16MB para el "PENTIUM" (como ejemplo). La RAM adicional por si sólo añade $2000 USD al precio de un "PENTIUM" a 166 MHz. (Asumiendo $40 USD por Megabyte). El

fabricante es quien decide qué combinación incluir para alcanzar la meta de precio/ejecución .

Los CPU's y los sistemas de memoria no deberían ser considerados como inteligencias separadas.

4.6.2.3 Interface con el Usuario

Los sistemas computacionales deben de comunicarse con seres humanos, con otras máquinas, o con ambos. Los usuarios esperan una interface humano-maquina sofisticada, intuitiva, y fácil de usar. Abrir ventanas en un monitor grande con pantalla de alta resolución es común: el colorido multiple ayuda a vender proyectos a ejecutivos y usuarios.

Manejar pantallas grandes implica un gran esfuerzo. La ejecución se ve afectada si el sistema pone toda la carga en el CPU para producir la pantalla. Un coprocesador de punto-flotante o matemático puede ayudar especialmente para dibujar figuras no rectangulares como diagonales y círculos. El hardware del CPU puede dibujar cuadrados simples usados para ventanas de texto con enteros aritméticos, pero dibujar círculos requiere computación con seno y coseno, la cual se hace mejor con un coprocesador matemático, con circuitería de apoyo en su interior. Mejor aún es, un coprocesador gráfico dedicado para el manejo de todas las tareas del despliegue visual display.

4.6.2.4 Conectividad

La frase "Ningún hombre es una isla" se aplica también a las computadoras. La habilidad de conectarse a las redes existentes es un prerrequisito para muchos compradores de computadoras. Algunos fabricantes incluyen capacidades de comunicación en sus sistemas operativos, y/o en su Hardware, otros requieren de mayor inversión para infraestructuras de redes locales. En algunas aplicaciones departamentales y para usuarios más sofisticados, no es extraño pensar en conectarse al mainframe (macrocomputadora) de la compañía.

4.6.2.5 Software Disponible

La mejor computadora del planeta sería inservible sin software. A menos que se desarrolle el propio, los costos del software implican construcción sobre el trabajo de programación de otros. Los criterios de selección de plataformas incluyen disponibilidad de sistemas operativos, utilerías, herramientas de desarrollo, lenguages de programación, y aplicaciones de oficina de propósito general. En un mercado activo el software desarrollado por terceros es una gran ventaja para cualquier plataforma. La competencia lleva al vendedor hacia nuevos logros de ejecución, pero entonces también los precios disminuyen. Nada puede ser más desolador que una buena máquina sin software.

4.6.2.6 Sistema Operativo

Elegir un sistema operativo es muy importante en la selección de una plataforma. El sistema operativo dominante para los workstations es UNIX, un sistema muy poderoso, pero con el cuál no mucha gente está habituada a trabajar. Sin embargo UNIX corre algunos de los softwares más potentes. En caso de que se intentara introducir UNIX en un nuevo ambiente de trabajo hay que disponer de un presupuesto para entrenamiento. Otra opción es el sistema operativo de la Macintosh System Tools 6.0, el cuál es muy fácil de usar. El más extensamente disponible es el MS-DOS. En en las plataformas PS/2 de IBM se encuentran disponibles tres sistemas operativos: MS-DOS, OS/2 que es un SO multitareas, con mandatos similares a DOS y, AIX un SO multiusuario basado en UNIX.

4.6.2.7 Ambientes de Distribución y Desarrollo

El desarrollo es considerado como una de las tareas más demandantes para el hardware, tal vez porque quienes desarrollan son quienes compran, y tienden a consentirse con grandes pantallas, ambientes de desarrollo muy lujosos, y los más rápidos compiladores.

Sin embargo, dado el costo de los Ingenieros del Conocimiento en estos días , cualquier gasto que se traduzca en un aumento de su productividad no es muy difícil de justificar.

Al elegir un ambiente de desarrollo, hay que asegurarse primero de la portabilidad del código final, hacia la máquina del usuario. En segunda, se examinan las ventajas respecto a la relación precio/ejecución de la máquina y, cual sería el valor en el aumento de la productividad del programador.

La sabiduría convencional dice que las máquinas Lisp (tagged processor machines) tienen los mejores ambientes de desarrollo, la facilidad de hacer verificaciones de tipo, durante la ejecución (run-time type-checking), favorece las etapas más tempranas del desarrollo del programa. Como el programador está más libre, otros aspectos del trabajo reciben mayor atención, siendo la circuitería (hardware) quien se encarga de los tipos de datos. Después de que el programa ha sido escrito y depurado, las declaraciones de tipo pueden ser agregadas. Estas máquinas se han distinguido también, como líderes industriales en sofisticadas interfaces humano/máquina.

Las plataformas para producción deben ser pensadas en función de las necesidades del usuario y el ambiente, ya que son muy sensibles al precio, en la medida en que los usuarios son más en número que los que desarrollan. La conectividad y la capacidad instalada pueden ser las mayores restricciones.

La extensa distribución de las plataformas productivas, ha cubierto casi cualquier lugar de trabajo de los usuarios. Las PC's de escritorio pueden parecer demasiado restringidas, pero quienes logren hacer aplicaciones útiles para máquinas con capacidad de memoria de 640k RAM encontrarán una capacidad instalada gigantesca.

4.6.3 Las Maquinas del Mercado

Hasta ahora, el mercado de las estaciones de trabajo ha estado dominado por las máquinas de (entre otros) Digital Equipment Corp., Hewlett-Packard Co., IBM Corp., Silicon Graphics y Sun Microsystems, todas basadas en UNIX.

La familia RS/6000 de IBM ofrece grupo de computadoras, basadas en tecnología RISC. Al parecer, esta familia muestra un sorprendente rendimiento en su amplia gama de

configuraciones. La filosofía de su diseño las hace apropiadas para ambientes de estaciones de trabajo, servidores de archivos y equipos multiusuarios. Esto permite que sean utilizadas en aplicaciones comerciales, científicas, orientadas a gráficas o de cómputo intensivo.

Las estaciones de trabajo (workstation) se hacen cada vez más accesibles. Una gran variedad de modelos se basan en el motorola 68040 con velocidades de operación que estan por ensima de los 25MHz .

El mejor competidor para las workstations y las máquinas LISP, es el "PENTIUM PRO". Intel y Motorola compiten intermitentemente por el primer lugar en relación al diseño de los circuitos (chips).

El "PENTIUM" puede ser encontrado en un número creciente de máquinas muy sofisticadas. El volumen y la competencia del mercado de DOS hace que estas máquinas se mantengan a precios algo atractivos, aunque los precios de los sistemas continuarán descendiendo conforme los CPUs más rápidos sean introducidos (existen rumores de que modelos a 180 Mhz y a 200Mhz ya están a la venta). Las PCs "PENTIUM" seguirán siendo la mejor opción durante algún tiempo. Los sistemas de memoria han mejorado, y los sistemas con memoria cache L2 de 256K ya están disponibles .

El software puede ahora tomar ventaja de la capacidad del "PENTIUM". Un ejemplo significativo es el ambiente de programación KEE/386 de IntelliCorp.

Existen también, buenos compiladores de LISP disponibles para el "PENTIUM" como el de GOLD HILL que corre bajo DOS, y LUCIDCOMMON LISP que corre bajo UNIX (se podría utilizar el "PENTIUM PRO").

4.7 ¿Qué Comprar?

A veces se insiste en comprar lo último y lo mejor que existe, pero esto sólo garatiza un gasto de dinero. Si se quiere la máquina más rápida y de tecnología de punta tal vez habrá que preparese para una escasez de software, aún cuando se pueda localizar algo.

Por ejemplo, el precio de una Sun 4 simple, es de alredor de $40,000. Sun ofrece un paquete especialmente para LISP que vale casi $80,000 incluyendo el Software. La familia RS/6000 de IBM ofrece una poderosa configuración desde $18,000.00 USD. Con estos parámetros una RS puede resultar una mejor compra.

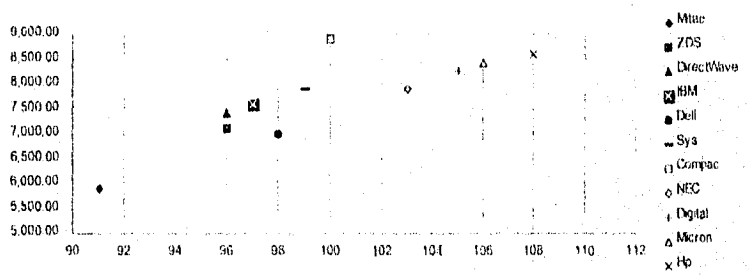
La Sun 4 es más rápida , pero tiene el inconveniete de operar con UNIX; un sistema operativo, muy diversificado. Un sistema de desarrollo completo incluyendo la Mac II podría costar al rededor de $25,000.

Symbolics ofrece un sistema de desarrollo por alrededor de $30,000. La Macintosh II puede parecer la opción ideal para el aficionado, pero en realidad es mucho más que eso. Las pruebas de ParkPlace con Smalltalk-80 muestran que es capaz de superar a algunas estaciones de trabajo profesionales, tales como Apollo y Sun. Se puede decir que la HP Apollo 9000 es la máquina de mejor rendimiento, el precio más elevado de una estación de trabajo de nivel intermedio, en comparación con una PC basada en Intel, se puede justificar para las aplicaciones científicas y de Ingeniería.

Si se esta buscando una estación de trabajo de alto rendimiento Intel tiene una propuesta que se ajusta a estas necesidades, se trata de estaciones de trabajo basadas en el microprocesador "PENTIUM PRO". Estamos en 1996 y el software de 32 bits esta a un año o dos de ser ampliamente adoptado.

La simple rentabilidad dicta que un sistema Pentium Pro debería estar equipado con el sistema operativo apropiado. Para lograrlo el sistema no es nada barato: es su pura implementación inicial a 150 MHz con 256K de cache L2 cuesta $974 USD por parte en cantidad de 1000 unidades.

En la figura se muestra una gráfica de la relación Indice-Precio/Desempeño de las estaciones de trabajo basadas en el "PENTIUM PRO".



Las líneas divisorias entre la Apple Macintosh, las PC's basadas en UNIX que usan RISC están desapareciendo cada día más. La PC tradicional ahora puede ejecutar sistemas operativos robustos y las estaciones de trabajo tradicionales ahora pueden usar sistemas operativos que ejecutan las operaciones de negocios populares.

4.7.1 Macintosh y MicroExplorer.

Por años, Apple Computer Company fue conocida por su Apple II, una computadora personal muy popular por sus usos "caseros" y fines educacionales. A principios de los 80's, justo cuando la IA comenzó a ganarse un lugar en el mercado, el co-fundador de Apple Steve Jobs, decidió que su compañía debería prepararse para incorporarse a la IA. En 1984 fué presentada en el mercado la primera Mac. Esta era más un concepto que una computadora. Tenía solo 128 KBRAM y no contaba con disco duro. Sus ventanas y su dispositivo apuntador (mouse), permitió que la gente se involucrara con las interfaces gráficas que ya se habían utilizado, desde hacía mucho tiempo, en IA. La Mac fué exitosa, sólo en la manera que permitió apoyar un proceso evolutivo hacia la Macintosh II, la cual fué presentada en marzo de 1987. A la gente de IA le gustó la Mac, pero no podía usarla. Con la Macintosh II la historia fué diferente. Ahora presentaba una capacidad de 8 Mb de RAM y mayor, si se hacía trabajar la memoria virtual, además de un enorme disco duro, y un canal de datos de arquitectura abierta (NuBus). La Mac II es muy adecuada para muchas de las aplicaciones comerciales [Harmon, 1989]. No es tan poderosa como la Sun, pero es más barata, fácil de interconectar en redes muy grandes, y corre toda clase de aplicaciones.

La Mac II está construida en torno al CPU 68020 de Motorola, el cual se basa en una arquitectura de 32 bits, con una velocidad de ejecución de 16.5 MHz, el tamaño básico de su memoria RAM es de 1Mb, la cual puede ser aumentada hasta 8Mb sobre la tarjeta principal (motherboard) y hasta 2Gb a través de las ranuras de expansión del NuBus.

La familia de procesadores 680x0 de arquitectura CISC es la usada por las máquinas Macintosches de Apple. El microprocesador utilizado es el 68040. Es un microprocesador de 32 bits con una unidad de números reales, unidad de administración de memoria y 8K de cache en chip. El 68040 presenta caches dobles (4K de instrucciones y 4K de datos) y

unidades de administración de memoria dobles. Esto le da al 68040 una ventaja en sus operaciones de manejo de instrucciones por que el acceso a los datos y a las instrucciones puede ocurrir al mismo tiempo. Este microprocesador tiene un cache de conjunto asociativo de 4 vías con 16 bytes en cada línea del caché.

El procesamiento numérico explota el poder cuantitativo de las computadoras, y así, ha revolucionado la productividad humana. Hoy en día el procesamiento simbólico comienza a explotar el poder cualitativo de las computadoras, y promete ofrecer mejores resultados, al proveer de conocimiento al mismo tiempo que datos. El procesamiento simbólico aumenta el potencial de la computación convencional. Esta avanzada tecnología utiliza símbolos para representar objetos del mundo real, conceptos y relaciones. Trata con problemas poco estructurados, interpreta información, aplica reglas y maneja problemas complejos, más allá de la capacidad de las más poderosas computadoras numéricas .

Existen dos co-procesadores LISP disponibles para la Mac II. Texas Instruments en acuerdo comercial con Apple, vende la MicroExplore, una Mac II equipada con un procesador Lisp. TI también vende una tarjeta co-procesadora Lisp, para quienes ya poseen una Mac II. Esta tarjeta cuesta alrededor de $10,000.00 USD.

Symbolics vende el sistema MacIvory, que consiste también de una Mac II equipada con un procesador Lisp. Este sistema incluye el ambiente Genera. Los usuarios de este equipo pueden adquirir , el paquete "Symbolics Joshua" para desarrollo de SE's, y Statice, una base de datos orientada a objetos. Este sistema esta disponible por $21,000.00 USD.

Ambos co-procesadores LISP, que han sido presentados, pueden correr software diseñado para las estaciones de trabajo ExplorerII y Symbolics 3600. Los dos sistemas son tan poderosos como sus antecesores, sin embargo, ambas compañías los ofrecen como plataformas de distribución, ya que no son tan rápidas como las originales. De acuerdo al precio, algunas compañías y grupos académicos han decidido que tambien funcionan como excelentes plataformas de desarrollo.

4.7.2 Plataformas IBM PS/2 PC's y compatibles

4.7.2.1 PS/2

En abril de 1987 IBM lanzó al mercado la familia PS/2 de computadoras de escritorio. La motivación de IBM era fundamentalmente, proveer al usuario de una plataforma que le garantizara una permanencia tecnológica y una mejor relación precio/rendimiento, facilidades de conectividad por su arquitectura abierta y el manejo de múltiples dispositivo através de un solo puerto (dispositivos SCSI), entre otras características.

Definitivamente, el avance más significativo de esta familia es, la implementación de la arquitectura de MicroCanal, la cual constituyó uno de los puntos polémicos en la presentación de las PS/2. La MCA (Micro Chanel Arquitecture) consiste básicamente en un canal de datos, inspirado en la tecnología de los sitemas 370 de IBM, que permite la utilización de procesadores independientes. Desde luego el aprovechamiento de tales características, no sería posible con el sistema operativo DOS, de ahí que, IBM desarrollara paralelamente su sistema operativo OS/2, el cual convierte a las PS/2 en computadoras multi-tareas (para mayor información sobre la arquitectura de MicroCanal ver[IBM,1990]). Este es un punto que favorece al desarrollo de sistemas de IA, ya que la mayoría de las aplicaciones requieren de cómputo intensivo, ya sea por el manejo de memoria o por la utilización de gráficas, etc.

La familia PS/2 incluye 8 computadoras de escritorio, 2 torres de piso y un modelo portatil.

4.7.2.2 RS/6000

La familia RS/6000 es la más nueva de las plataformas IBM. Estos sistemas están basados en la tecnología RISC de segunda generación, creados alrededor de una

arquitectura de sistema abierto con capacidad para multitareas. Todos los integrantes de la familia RS/6000 incorporan la arquitectura POWER IBM, implantada bajo tecnología de muy alta escala de integración (CMOS-VLSI). Esta arquitectura aprovecha al máximo diversas inovaciones de diseño. La existencia de unidades separadas que funcionan en forma simultanea para el procesamiento de punto fijo y punto flotante, junto con memorias cache separadas para datos e instrucciones, permite que el procesador central pueda realizar hasta cinco operaciones en un solo ciclo de reloj . Otra característica importante es la nueva modalidad de operación de formato "streaming" para un bus MicroCanal de 32 bits y otro bus separado de 64 o 128 bits para transferencia de información entre el procesador central y la memoria.

El sistema operativo de estas estaciones de trabajo es el AIX (Advance Interactive System), gracias al cual se puede obtener un máximo aprovechamiento del hardware. Aun que no es tan exuberante la lista de aplicaciones construidas para AIX, como las construidas para DOS o OS/2, ya hay disponibilidad de algunos paquetes en el mercado. IBM ofrece por consistencia y compatibilidad que estos sistemas pueden correr bajo OS/2, aunque el rendimiento y aprovechamiento del sistema no es el mismo.

4.7.2.3 PC Compatibles

La gama de computadoras personales PC compatibles es inmesa. Las marcas disponibles en el mercado mexicano van desde las modestas Printaform, hasta las costosas Hewlett-Packard. Para no quedarse atrás, DEC presenta su familia de computadoras personales PCSA (Personal Computing Architecture), que es una extención de la arquitectura de redes "digital", que mezcla los ambientes de PC y VMS. Además de incorporar algunos módulos de comunicación con plataformas Macintosh y las características que facilitan su interconexión en red. Las PCSA's ofrecen configuraciones similares a las de otras marcas. Para las aplicaciones de IA, como se ha mencionado en secciones anteriores, se requieren plataformas de computo intensivo, y de preferencia aquellas que tengan procesamiento simbólico, algo que hasta ahora ninguna de las compatibles ofrece. Por otra parte, no se sabe con certeza cuál ser el futuro del DOS. Sin embargo, las características que privan en

el mercado nacional, indican que la permanencia de las PC's podría estar garantizada en por lo menos 5 años. Por lo tanto, si se trata de escoger una herramienta de desarrollo de aplicaciones de IA y SE's, no es desafortunado considerar a estas plataformas, sobre todo considerando sus precios. Es probable que para sistemas no muy grandes y presupuestos reducidos, no exista mejor opción.

CAPITULO 5

PRESENTACION DEL LABORATORIO DE SISTEMAS EXPERTOS EN LA ENEP ARAGON

5.1 INTRODUCCION

Este capítulo trata de la conjunción de conceptos teóricos y prácticos revisados y analizados en los capítulos anteriores. Lo que en las siguientes secciones se discute, es una propuesta formal de los objetivos, la infraestructura, la operación y las actividades que debe llevar acabo un Laboratorio de Desarrollo de Sistemas Expertos. Los próximos párrafos están dedicados a resumir y aterrizar una serie de criterios que orientaron el presente estudio. La proposición estructural está enfocada desde la óptica de los requerimientos de un proceso de enseñanza encaminado a cubrir las necesidades del mercado.

Para la propuesta final se utilizó una metodología de diseño apoyada en la metodología de Crosby para la descripción e implementación de procesos industriales. Para ello, se asumieron similitudes entre el proceso de enseñanza y los procesos industriales. Sentado lo anterior, se separaron y clasificaron los elementos que intervienen en el proceso y se incluyeron en cada una de las secciones respectivas. La utilización de este modelo facilitó la descripción y la claridad de la propuesta final. Contrario a lo que pudiera parecer, el modelo ofrece características de estructuración, más no afecta la calidad humana de la enseñanza, sino que por el contrario, le ofrece más posibilidades gracias a su organización.

5.2 METODOLOGIA DE DISEÑO.

La propuesta estructural del Laboratorio de Desarrollo de Sistemas Expertos, debe hacerse considerando a éste como un elemento en el proceso de enseñanza. Para tal efecto, hay que comenzar por definir qué es un proceso. En principio, cualquier trabajo que se realice puede ser considerado como un proceso. Esto quiere decir que haya una sucesión de acciones que produzcan un resultado. Partiendo de esta idea, se reconoce que para que un proceso funcione, se requiere de ciertos recursos. Al tratar de describir los elementos que interviene en un proceso, lo que se obtiene es un modelo de proceso. La claridad y

simplicidad que muestra el modelo de proceso de Crosby [Crosby, 1990], permite utilizarlo en el presente trabajo, guardando algunas diferencias con el modelo original.

Los elementos que Crosby propone son:

- Objetivos (Alcances)
- Resultados
- Insumos transformables
- Materiales de información
- Insumos de control
- Procedimientos
- Equipo e instalaciones
- Habilidades y Conocimientos
- Estándares de realización

En las siguientes secciones se definen los elementos que describen el modelo de proceso y se indican aquellos que particularmente atañen a la propuesta del laboratorio.

5.3 OBJETIVOS.

El funcionamiento del laboratorio, debe estar dirigido por objetivos bien definidos. Cualquier actividad humana requiere de una definición clara de las metas a alcanzar y de los mecanismos de medición del éxito. En algunas instituciones se definen "estandares de realización", como una forma de medir los requerimientos de excelencia en la obtención del objetivo final. A continuación se proponen los objetivos del Laboratorio de Desarrollo de Sistemas Expertos y las aportaciones que deberá ofrecer a los alumnos de la carrera de Ingeniería en Computación y la comunidad universitaria en general.

Auxiliar en la selección del campo de trabajo del alumno.

A través de las prácticas y proyectos realizados en el laboratorio, los alumos serán capaces de reconocer sus habilidades personales y orientar sus intereses hacia las diferentes áreas de aplicación de los SE's (Diseño,Clasificación, Toma de decisiónes, análisis, etc). De

igual forma serán capaces de evaluar cuáles de estas aplicaciones presentan una mejor perspectiva en nuestro país, mediante la elaboración de proyectos empresariales conjuntos.

Orientar en la utilización de herramientas.

La experiencia que el alumno adquiera al manipular las diferentes herramientas que el laboratorio le ofrezca, le permitirá evaluar con mayor certeza, que tipos de herramienta, desde el punto de vista práctico, son más recomendables de acuerdo al tipo de problema que se pretende resolver.

Apertura absoluta a los alumnos de Ingeniería en Computación.

Los alumnos del Departamento de Ingeniería en Computación, deben encontrar en el laboratorio un apoyo a la realización de sus proyectos académicos y de titulación, siempre y cuando el coordinador del laboratorio, previa evaluación, haya encontrado que el proyecto en cuestión presenta un problemática digna de ser atacada con ténicas de IA y SE's.

Proveer una base sólida para seleccionar el área de desarrollo profesional.

A partir de la experiencia académica adquirida en el laboratorio, el alumno deberá obtener los elementos de juicio necesarios para seleccionar el área de desarrollo profesional que le resulte más interesante. Para lograrlo, el alumno trabajará en proyectos que le muestren conceptualmente las diferencias entre las áreas de Investigación en Ciencias Básicas, Investigación Aplicada y Desarrollos Empresariales.

Generará un catálogo de aplicaciones.

La documentación de todos los proyectos realizados dentro del laboratorio, debe formar una métodología de trabajo en el alúmno, así como integrar una biblioteca de "casos" a manera de acervo tecnológico, que permita a los alumnos de generaciones posteriores tener acceso a un "banco de experiencia". De esta forma los subsecuentes desarrollos se verán enriquecidos, con mejores tiempos de desarrollo y mayor calidad en el contenido de sus soluciones.

Generar tecnología (Proponer metodologías de trabajo).

A pesar de que los fundamentos tecnológicos y científicos de la IA y SE's han sido expuestos desde el final de la década de los 50'as, la experiencia acumulada y documentada de casos prácticos es muy reciente. Algunas metodologías aún se encuentran en proceso (por ejemplo, Ingeniería del Conocimiento) y el consenso entre los científicos y expertos

del área todavía no se logra consolidar. De ahí que los desarrollos elaborados dentro del laboratorio, tengan la oportunidad de contibuir en la proposición de una metodología de trabajo con SE's, que se adecue a la realidad nacional y al papel de la universidad dentro de la sociedad. Asimismo podrá contribuir a los programas de excelencia académica de la universidad.

Proporcionar un foro a expositores externos y promover las relaciones universidad-industria.

Los tópicos académicos y temas de estudio a tratarse en el laboratorio, pueden involucrar distintos y muy diversos aspectos del conocimiento científico. El complementar estos temas con actividades tales como ciclos de conferencias, mesas redondas, simposiums, exposiciones y otras formas de intercambio académico, ayudarán a normar el criterio del alumno y ofrecerán una oportunidad de exponer su trabajo ante la comunidad universitaria y los sectores empresariales interesados en el área, así como fomentar el crecimiento del acervo tecnológico del Depto. de Ingeniería en Computación y consecuentemente de la universidad.

El intercambio de experiencias con otras instituciones académicas y empresariales, eleva la calidad de la enseñanza y mantiene actualizado el contenido de la misma.

5.4 RESULTADOS

Todo proceso ofrece un resultado. Otra forma de visualizar un proceso, es modelándolo como una caja negra, a la cual a la entrada se le ingresa cierta información, materia prima e insumos. La salida entrega información enriquecida como resultado. Quienes ingresan la información se les conoce como proveedores y quienes la reciben son los clientes. Estos resultados deben tener ciertas características, que se llaman requisitos, para que sean de utilidad al cliente. Asimismo, los proveedores deben cubrir ciertos requisitos en su materia prima, para que el proceso se lleve a cabo.

5.4.1 Clientes

Los principales clientes del laboratorio serán los alumnos, quienes a su vez servirán a las empresas e industrias que los contraten. Desde luego, para que la cadena no se interrumpa, los alumnos deberán cubrir ciertos requisitos. De acuerdo a la observación realizada a través de entrevistas y encuestas, a continuación se menciona una lista de conocimientos y habilidades deseables en los profesionistas dedicados a trabajar en el campo de los SE's. Aunque algunas características son comunes, a continuación se separan los perfiles según las áreas de trabajo clasificadas en el capítulo 3.

Investigación Académica.

- lenguajes de programación de IA (LISP, PROLOG).
- Arquitecturas de SE's.
- Mecanismos de inferencia y control.
- Métodos Generales de Solucion de Problemas en IA.
- Lógica.
- Representación de conocimiento.

Investigación Aplicada

- Herramientas de Programación.
- Areas de Aplicación.
- Formas de Representación de Conocimiento.
- Visión general del área.

Desarrollos de Sistemas Comerciales.

- Facilidad de palabra.
- Capacidad de abstracción.
- Interfaces de usuario.
- Visión modular de los problemas.

Estas caracteristicas son recomendaciones recopiladas del estudio de una muestra de los diferentes sectores que componen el mercado de los SE's, sin embargo, su definición final está supeditada a los cursos y temarios que el Depto. de Ingeniería en Computación

determine. En el apéndice "A" se muestran los temarios de las asignaturas relacionadas con el tema y cursos impartidos por el Depto. de Ingeniería en Computación.

5.4.2 Proveedores

Fundamentalmente los proveedores del laboratorio, serán los maestros e instructores que trabajen con los alumnos. También se consideran proveedores de experiencias y conocimientos a los conferencistas y expositores externos que participen en las diversas actividades que más adelante se describen. De igual forma en la que se determinaron requisitos o características de los egresados, se tendrán que determinar para los proveedores. La selección de profesores, instructores y demás personal académico, debe corresponder a los criterios que ha aplicado el Depto. de Ingeniería en Computación en otras áreas. Por lo que respecta a los colaboradores externos y eventuales, es prudente también marcar algunos criterios de selección de acuerdo con su reputación académica y profesional.

5.5 INSUMOS TRANSFORMABLES

Los insumos transformables puede ser de dos tipos: Materiales y de Información. En las siguientes secciones se mencionan los insumos de información que habrán de alimentar al laboratorio.

5.5.1 Actividades

Hoy en día no resulta extraño hablar del vertiginoso ritmo con el que cambia la tecnología. De ahí que el intercambio de experiencias y conocimientos sea un método adecuado para cualquier institución o persona que pretenda mantenerse actualizada. Por ello es importante que el Laboratorio de Desarrollo de Sistemas Expertos, cuente con apoyos didácticos como los que a continuación se proponen:

Seminarios:

Impartidos por profesionales del área, sobre temas diversos en torno a SE's e IA.

Simposiums:

Foros donde se expongan diferentes trabajos de investigación y desarrollo y donde se promueva la participación activa del alumnado.

Exposiciones:

Donde se muestren equipos y herramientas que mantengan al tanto a profesores y alumnos de las tendencias del mercado.

Ciclos de conferencias:

Dictadas por reconocidos académicos y profesionistas (de nuestro país, y en medida de lo posible del extranjero).

Círculos de lectura:

Donde los alumnos puedan recopilar e intercambiar artículos, libros y revistas sobre diversos temas y bajo la dirección de un coordinador calificado. El coordinador debe ordenar los temas de acuerdo a las preferencias de los participantes.

Intercambios con otras instituciones:

Es recomendable el buscar convenios con empresas, industrias y otras instituciones académicas para apoyar los proyectos que se desarrollen en la Universidad y para dar proyección a los alumnos, ya sea para seguir en programas académicos (postgrados) o para incorporarse al sistema productivo.

5.5.2 Acervo Bibliográfico.

Entre los apoyos didácticos que se requieren para el laboratorio, se encuentran publicaciones de diversa índole. El objeto de tener acceso a dicha literatura, permitirá a los alumnos y profesores mantenerse al tanto de los acontecimientos mundiales en el área de los SE's y la IA. Esto da la oportunidad de que los temarios se sometan a revisión constante y que se incorporen conceptos de nuevas tecnologías. Incorporar este tipo de apoyos, es necesario no por el simple hecho de conocer los últimos adelantos en la materia, sino porque capacita al laboratorio y a su personal para asimilar, probar e instrumentar soluciones más

creativas y con mejor oportunidad académica y comercialmente hablando. A continuación se listan las publicaciones recomendadas, de acuerdo a las diferentes áreas de desarrollo profesional detectadas en nuestro país.

5.5.2.1 Investigación en Ciencias Básicas

Artificial Intelligence editada por North Holand es una revista en la que se encuentran artículos de muy alto nivel académico con autores de la calidad de Newell y Simon, Feigenbaume investigadores europeos de alto nivel, provenientes de universidades como la de Edimburgo (Una de las universides con mejor reputación mundial en el área de IA) y otras de reconocido nivel académico. La suscripción anual a esta revista tiene un costo de $1,000.00 USD.

Proceedings of The International Join Conference of Artificial Intelligence contiene trabajos de gran relevancia, ya que presentan tecnologías y conocimientos obtenidos en los laboratorios e instituciones de investigación más importantes del mundo. En esta publicación se presentan con frecuencia, diferentes alternativas de solución a los problemas clasicos de la IA.

La revista mensual de la AAAI presenta además de artículos científicos de alta calidad, calendarios de eventos diversos que lleva acabo la AAAI. Dentro de estos eventos se encuentran conferencias, exposiciones, programas de intercambio, programas de becas que ofrecen diferentes universidades en los EUA, concursos, etc., los cuales pueden contribuir a la motivación de los alumnos y profesores del laboratorio.

5.5.2.2 Investigación Aplicada.

AI Expert. es una revista muy accesible al lector. La mayoría de sus artículos versan sobre soluciones a problemas reales, que se han presentado en empresas de diversa índole. Aquí tambien aparecen con frecuencia evaluaciones, pruebas y editoriales sobre nuevas

herramientas software, hardware y tecnologías . Además no faltan los artículos que hablan acerca del impacto de la IA y los SE's en la cultura organizacional de las empresas y cuáles son las alternativas que han presentado los directamente afectados. La suscripción anual cuesta al rededor de $45.00 USD.

PC AI es una revista modesta pero que tiene la peculiaridad de centrarse en el mundo de las PC's. Sin hacer alarde de grandes recursos, esta publicación también presenta algunas evaluaciones sobre las herramientas más populares de IA y SE's para PC's. Sus artículos son bastante accesibles y a menos, algunos de ellos puden ser buenas introducciones de temas complejos y sofisticados.

5.5.2.3 Desarrollos comerciales

Intelligent Software Strategies es una publicación de tipo boletín que presenta de manera muy clara las tendencias del mercado en los E.U.A.. Sus artículos comprenden análisis y evaluación de herramientas bajo perspectiva de costos y beneficios, análisis de SE's en funcionamiento y comparaciones de acuerdo a utilidades reportadas, simplicidad de operación y parámetros sobre su construcción como herramientas de programación, tiempo de desarrollo, etc. Suscribirse a este boletín por un año cuesta cerca de $400.00 USD.

5.5.2.4 Registro De Casos

El registros de caso de cada semestre (Documentación), permitirá al laboratorio formar una acervo tecnológico propio. Este registro de casos debe estar bibliográficamente ordenado para facilitar su consulta y almacenamiento. Es probable que no se puedan guardar y clasificar todos los casos resueltos en un semestre, por lo que se recomienda al instructor guardar los mejores, con lo que además se puede estimular a los alumnos que participen en los proyectos.

Para efectuar el registro, se propone crear una estructura estandarizada del reporte, que conste de secciones como: presentación, descripción, solución y apéndices técnicos donde se indiquen las herramientas utilizadas, diagramas, árboles de decisión, esquemas de

representación, base de conocimiento y técnicas de adquisición de conocimiento utilizadas en el proyecto. En el apéndice "C" se muestra un ejemplo de reporte mostrando una estructura que se podría utilizar.

5.6 INSUMOS DE CONTROL.

Los insumos de control permiten que el proceso funcione al mismo tiempo que lo controlan. Esto indica la utilización de materias primas que no han de ser transformadas por el proceso, sino que se le agregan a éste, para garantizar su óptimo funcionamiento.

5.6.1 Procedimientos

En cualquier centro de trabajo debe existir una forma de administrar los recursos disponibles, por eso el establecer políticas de uso de las herramientas, tanto computadoras como paquetes y programas (software), es indispensable. Estas políticas deben contemplar formas de trabajo en equipo, es decir: se debe determinar el número de alumnos que participan en un proyecto, de acuerdo a su magnitud y complejidad. Esta determinación será el resultado de un análisis conjunto entre los alumnos y el instructor. La asignación de recursos procurará ser lo más económica posible, de manera que los alumnos puedan trabajar con herramientas adecuadas a la dimensión del proyecto en turno. La organización de horarios de trabajo debe apegarse a la disponibilidad que las autoridades universitarias decidan otorgar al laboratorio. Una propuesta podría contemplar horarios especiales para alumnos de cada asignatura, alumnos en proyectos de cooperación (proyectos con alguna empresa) y tesistas. Desde luego esta organización dependerá del equipo y herramientas que sean asignados al laboratorio.

El cambio continuo de la tecnología, obliga a formalizar políticas de actualización de herramientas, para garantizar la vigencia de los cursos y prácticas del laboratorio. Para elaborar estas políticas, se propone recopilar información acerca de nuevos productos en el mercado (esta labor puede apoyarse en algunas de las publicaciones propuestas), monitorear

las áreas de desarrollo de SE's en nuestro país y revisar presupuestos, convenios, y donaciones con las que cuenta la ENEP.

Es recomendable estudiar propuestas de financiamiento e inversión de empresas e industrias interesadas en el área, para tratar de lograr un presupuesto propio del laboratorio. Las posibilidades de los proyectos de cooperación dependen de los resultados que ofrezca el laboratorio, es conveniente empezar a publicitarlo entre las empresas, para que en el mediano plazo se encuentren propuestas formales de inversión.

5.6.2 Instalaciones y Equipo

En esta sección, a diferencia de lo que se hizo en el capítulo 4, se discuten los criterios desde un punto de vista práctico para seleccionar las herramientas en las que se apoya la operación del laboratorio. Los parámetros que aquí se toman encuenta, no contraponen la disertación teórica realizada en el capítulo mencionado. Por el contrario, a continuación se presenta un panorama de la infraestructura que actualmente posee el Depto. de Ingeniería en Computación, y la cual sirve como punto de partida a la presente propuesta.

Una vez revisados los conceptos teóricos sobre la construcción de SE's y el horizonte nacional respecto a esta tecnología, es posible plantear las características idóneas de un laboratorio de desarrollo de SE's. Esta perspectiva muestra una clara interdependencia entre las herramientas de programación, las plataformas computacionales que se utilizarán y los requerimientos del mercado nacional. En los siguientes párrafos se tratan los puntos importantes al respecto.

5.6.2.1 Hardware

Primeramente vale la pena hacer una descripción de lo que podría ser la plataforma computacional ideal para desarrollar aplicaciones de IA y SE's. Dada la complejidad y dinamismo de las estructuras de datos que estas aplicaciones requieren, resulta claro que el primer recurso que será exhaustivamente explotado, es la capacidad de almacenamiento dinámico (RAM). Por supuesto, para que existan resultados, se requiere de un

procesamiento de estos datos. Por la forma y contenido de los mismos, el procesador idóneo para esta tarea es aquel que pueda ejecutar procesamiento simbólico (procesador LISP).

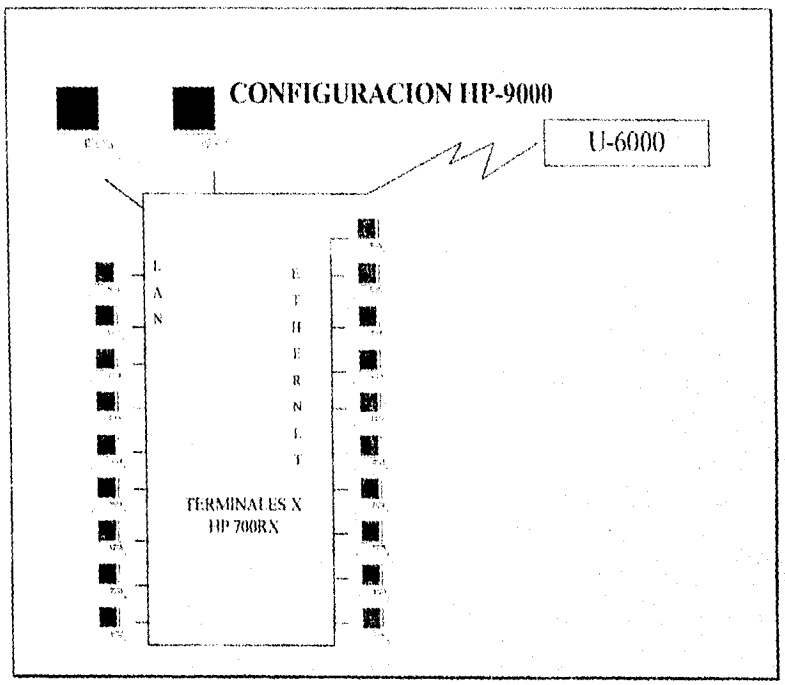
Otro punto importante a considerar es la manipulación de gráficas. Sin que se trate de un punto vital para el usuario final o para el desarrollador, el que una aplicación presenta gráficos resulta de gran ayuda para ambos. Si bien es cierto, bajo ciertas circunstancias pudiese considerarse un lujo, más que una necesidad. Por otra parte, no se puede negar que la tendencia generalizada en el desarrollo de sistemas, es el de crear interfaces más amigables para el usuario, mismas que se basan en el uso exhaustivo de menus de multinivel, íconos y gráficos. Por supuesto, el manejo de gráficos tiene asociado un costo computacional considerable. Se requiere de CPU's (Unidad Central de Proceso) de alto rendimiento, y a veces hasta de procesadores dedicados a manipulación de gráficos.

Por último, se debe tomar en cuenta la carga de trabajo que implican los procesos de entrada y salida. Si se considera una estación de trabajo de desarrollo para atender a un solo usuario a la véz, la carga de trabajo de E/S (Entrada/Salida) no será importante, pero si se piensa en una plataforma multiusuario, entonces el número de procesos de E/S puede ser un factor que afecte significativamente el desempeño de la estación.

Ya que se han establecido los criterios de selección se puede proponer equipo disponible en México y útil al laboratorio. El primer paso es ofrecer alternativas para aprovechar el equipo que ya se encuentra en la Universidad. Bajo la administración del Depto de Ingeniería en Computación, se cuentan los siguientes equipos: **configuración HP 9000, configuración U 6000, PC's, Sun, y esta conectada a la red UNAM.** Este equipo apoya a las actividades académicas de diversas asignaturas y su disponibilidad para el laboratorio no sería de tiempo completo. De acuerdo a la forma de asignación de recursos, que sigue el Depto, se debe proponer un equipo que además de favorecer el desarrollo de SE's, brinde apoyo al resto de las actividades académicas. Bajo un punto de vista estricto, esto no es deseable, pues se espera que el laboratorio a mediano plazo trabaje a toda su capacidad, sin embargo, las circunstancias indican que a corto plazo, compartir equipo para diferentes funciones es una política que no cambiará.

5.6.2.1.1 CONFIGURACION HP 9000.

La configuración **HP 9000** esta formado por dos equipos HP900 de la serie 700 (720 y 730) y 17 terminales X modelo 700/RX. Todos los equipos están enlazados en una red tipo "bus lineal"mediante tarjetas Ethernet. Están utilizando configuración Cliente servidor; en el cual, los clientes (Terminales X) realizan peticiones de trabajo que son atendidas y realizadas por el servidor (HP 9000). Está configuración forma parte de una red mayor en la que se encuentra conectado un Equipo U 6000 y PW^2 de UNISYS.



CARACTERISTICAS DEL EQUIPO HP 9000 .

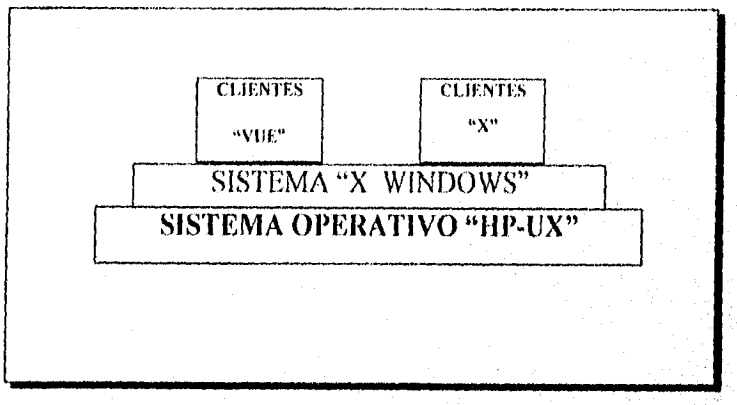
HP-9000/ 730

El equipo está basado en la arquitectura PA-RISC que es la implementación mejorada de HP. La velocidad del reloj es de 66Mhz y tiene instalada 80MB de memoria RAM; La cual

se puede extender hasta 128 MB. La capacidad en disco es de 64 GB, teniendo actualmente sólo 664 MB en un disco SCSI. Además contiene una unidad de cassette DDS SCSI, cuya capacidad de almacenamiento es de 1.3 GB por cassete; unidad de CD ROM SCSI de 599 MB y una unidad de diskette de 3.5 pulgadas.

El equipo tiene interconstruido un conector Thin LAN y Thinck LAN, que le permiten conetarse a una red Ethernet; una para video al que se encuentra conectado un monitor de 19'' monocromático; uno para dispositivos HP-HIL que tiene conectados el teclado y un ratón; un puerto paralelo, dos seriales RS232 y uno para tarjetas EISA.

El sistema operativo instalado es el HP-UX 8.07; que es la versión UNIX de HP. Sobre el sistema operativo se encuentra corriendo el ambiente gráfico X Windows y sobre éste, esta HP-VUE (HP Visual User Environment), que es el conjunto de programas (clientes) que permiten ejecutar comandos mediante iconos.



Actualmente cuenta con los lenguajes de programación Fortran 77, Pascal, Cnbol, C y C++. Próximamente será instalada paquetería, entre la que se encuentra LOTUS 123, la librería gráfica Startbase y el ambiente de desarrollo SoftBench.

HP-9000/720.

El hardware es parecido al mencionado anteriormente; pero con menos capacidad. Consta de la arquitectura PA-RISK corriendo a 50 Mhz, con 80 MB en RAM y dos discos SCSI de 420 MB. cada uno. Al igual que el equipo 730 , tiene conectores Thin y Think LAN para video y dispositivos HP-HIL, así como dos puertos seriales y uno paralelo. Además cuenta con un monitor monocromático de 19'', teclado y ratón.

En relación al software tiene el mismo que el modelo 730.

TERMINALES X.

Una estación de trabajo X esta compuesta por un monitor monocromático de 19'' con una resolución de1280x1024; un teclado y un ratón de tres botones HP-HIL; una unidad de proceso, la cual carga el software necesario para manejar el ambiente gráfico y realizar las comunicaciones desde el servidor, en lo que se encuentrán coriendo todos los procesos. Cada unidad tiene instalados 4 MB de RAM, un conector Thin y Thinck LAN, uno para los dispositivos HP-HIL, uno para video, así como uno para puerto serial y uno paralelo.

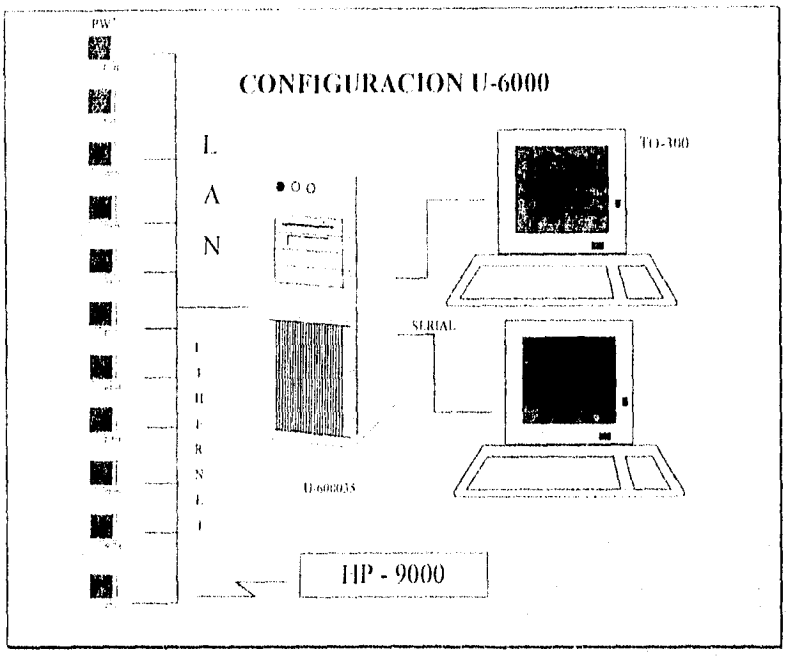
Este equipo no tiene software propio; el que se necesita para funcionar lo lee del sevidor a través de la red.

5.6.2.1.2 CONFIGURACION U-6000.

Este sistema esta formado por un equipo U-6000 modelo 35 como servidor y 10 PW^2 386 trabajando como terminales X; todos enlazados en red mediante tarjetas Ethernet. En forma adicional se encuentran conectadas al U-6000, dos terminales TO-300 mediante una tarjeta de entrada /salida de 8 puertos.

Esta configuración forma parte de una red mayor en la que se encuentran conectados dos HP-9000 (modelo 720 y 730) y 17 terminales X 700/RX de HP.

Las terminales X y las PW^2 pueden entrar a sesión a la U-6000 o cualquiera de las HP-9000 y establecer cominicación con cualquier equipo en la red.



CARACTERISTICAS DEL EQUIPO U-6000.

U-6000/35

Cuenta con un microprocesador Intel 80486 a 33 Mhz, 24 MB de memoria en RAM, 1 tarjeta E/S de 8 puertos , una tarjeta Ethernet, una unidad de cartucho de 1/4'' de 150 MB, una unidad de disco flexible de 5 ¼ de 1.2 MB. 1 disco duro SCSI de 425 MB y otro de 660.

El sistema operativo que tiene instalado es el UNIX system V Release 4.1.1, que es una versión mejorada del UNIX V.4 de AT&T; además, cuenta con el software de comunicación para la red como es TCP/IP, los servivios de ARPA, Telnet, etc. que permiten la comunicción con las computadoras PW[2], los lenguajes de programación FORTRAN, pascal y un paquete denominado "Ambiente de Desarrollo de Software" que permite crear aplicaciones con el lenguaje C; así como el ambiente gráfico "Primary

Graphical Environment (PGE)", el cual esta basado en X Windows, la interface gráfica de usuarios de MOTIF, utilerías gráficas como X . Desktop. Esto permite trabajar en un ambiente gráfico, y mediante iconos, realizar tareas como copiar y borrar archivos, cambiarse de directorio , ejecutar programas, etc. Entre las utilerías se encuentra un emulador de terminal que permite ingresar comandos propios de UNIX; de tal forma que se puedan tener vcarias terminales virtuales dentro de una estación de trabajo.

PW^2 modelo 3252.

Se tiene 10 microcomputadoras PW^2 , las cuales cuentan con un microprocesador 80386 SX a 25 Mhz , 6 MB de RAM, un disco duro SCSI de 40 MB, una unidad de discos flexibles de 3 ½ , de 1.4 MB , un monitor Super VGA de color , teclado en Español y ratón. Tiene instalada una tarjeta Ethernet y el software de comunicación se encuentra el llamado Xsigth, que permite el acceso al ambiente gráfico de la U-6000 (PGE) y de los equipos HP-9000 (HP VUE).

Terminales TO-300

Las dos terminales monocromáticas con que cuenta actualmente el equipo, se encuentran conectadas a través de cables seriales a la tarjeta de E/S de la U-6000; esto permite entrar a sesión y trabajar con el sistema UNIX. estos equipos no pueden trabajar en ambiente gráfico; sino en el modo de texto como cualquier terminal . Tienen un conjunto de herramientas como son una calculadora , un reloj con alarma y un calendario.

5.6.2.1.3 RED UNAM.

A principios de 1994 la ENEP Aragón se integró a la RED UNAM y actualmente en el centro de computo de este plantel ya se cuenta con el acceso a todos sus servicios.

INTERNET.

A su vez, RedUNAM está conectada a *internet,* la cual es una red o conjunto de redes de computadoras interconectadas entre sí a nivel mundial para la comunicación de datos. Esta

es la red más grande del mundo y su proposito principal es el de comunicar entidades gubernamentales, universidades, centros de iovestigación y organizaciones industriales.

Todos los equipos y redes en Internet, usan el protocolo TCP/IP (Treansport Control Protocolo/Internet Protocol), que es un lenguaje de comunicación entre casi todas las marcas y modelos de computadoras.

Red UNAM tiene acceso a la diversidad de servicios y herramientas que provee internet.

Entre estos servicios destacan:

MAIL (CORREO ELECTRONICO).

Este servicio permite al usuario mandar y recibir mensajes de correo electrónico a individuos o grupo de individuos.

Los programas que manejan el correo, aceptan y almacenan mensajes y los distribuyen al usuario al cual van dirigidos. La mayoria de los usuarios tienen un buzon muy personal de correo, donde todos sus mensajes recibidos son almacenados.

FTP (TRANSFERENCIA DE ARCHIVOS).

La tranferencia de archivos permite mover urchivos desde y hacia otras computadoras, aún si cada computadora tiene un sistema operativo y un formato de almacenamiento diferente. Los archivos pueden ser datos, programas, reportes, etc, y con cualquier tamaño.

TELNET(ACCESO REMOTO).

Esta herramienta permite a un usuario establecer sesiones interactivas en otras computadoras remotas. Una vez establecida la sesión con la máquina remota, el usuario puede correr programas, capturar datos, o hacer cualquier otra operación como si la terminal remota fuera local.

ARCHIE.

Archie localiza información disponible en la red. Es en realidad una lista que contiene los nombres de todos los archivos de dominio público almacenados en miles de computadoras conectadas a internet. Las redes con frecuencia ofrecen bases de datos

centrales, que pueden consultarse desde cualquier terminal que este conectada a la red. Estas bases de datos pueden directorios que contengan información sobre temas o artículos sobre un tópico en paricular, así como información de los usuarios de la red.

GOPHIER.

El Gopher de Internet Combina las características de los boletines electrónicos y las bases en datos de un sistema de información de distribución jerárquica, lo que permite reastrear información que se necesite utilizando indices de texto completo. Los Gophers almacenan una gran cantidad de información, incluyendo documentación de computadora, directorios, noticias, meteorología, bases de datos de bibliotecas, libros, sonidos, información por medio del comando finger, etc.

5.6.2.1.4 PC's.

Las plataformas RISC son también dignos candidatos para el desarrollo de SE's. Sin embargo, hoy en dia, no es claro para el mercado nacional qué, portabilidad podrían ofrecer los sistemas desarrollados en estas plataformas. Actualmente en México, las mejores opciones para adquisición de plataformas RISC son la RS6000 de IBM y la familia DEC stations de Digital. La primera puede operar bajo OS/2 o bajo AIX, un nuevo sistema operativo de IBM que incluso opera en las PS/2 más grandes. Las segundas son plataformas de muy alto rendimiento que operan bajo ULTRIX, una versión de UNIX para estaciones de trabajo. Para las máquinas DEC ya existen herramientas de programación de IA y SE's, para la máquina IBM los productos al respecto son menos. La factibilidad de desarrollar proyectos en cualquiera de estas dos plataformas depende del presupuesto del laboratorio, ya que son equipos muy caros y las herramientas que operan en ellos también lo son. Como una herramienta formativa puede ser útil. De acuerdo a la realidad nacional es posible adquirir valiosos conceptos básicos, tabajando en herramientas menos sofisticadas.

Como se mencionó en el capítulo 3, dadas las características del mercado mexicano, una de las áreas de oportunidad más importante para los SE's, se encuentra en las aplicaciones

de PC's. Por ello, es deseable proponer herramientas orientadas a satisfacer ésta demanda. Gracias a los sorprendentes avances de la tecnología de semiconductores, hoy en día, se puede contar con increíbles capacidades de cómputo y enormes volúmenes de almacenamiento sobre cualquier escritorio a costos increíblemente bajos. A pesar de que todavía las herramientas de construcción de SE's son grandes consumidoras de recursos, sus demandas ya empiezan a sentirse razonables. De acuerdo a las herramientas revisadas en el capítulo 4, se puede decir que la aplicación más sencilla que se desee desarrollar, requerirá de una capacidad de memoria dinámica (RAM) de por lo menos 2MB. Si tomamos en cuenta que la mayoría de los fabricantes de PC's instalan de fábrica 1MB en sus plataformas, resulta razonable pensar en agregar 1MB más, sobre todo considerando el abaratamiento de los chips (encapsuladas semiconductores) de memoria, principalmente desde la aparición de los SIMS.

De acuerdo a las tendencias del mercado, que constantemente se discuten en las revistas especializadas, parece que la permanencia del CPU de Intel 80X86 ó "pentium" está garantizada por varios años. Es indudable que al hablar de un procesador se debe hablar del sistema operativo que lo gobierna. En este caso, pese a todos los pronósticos, también parece estará garantizada la permanencia de DOS como sistema operativo líder en el mundo de las PC's. A raíz de la aparición de la familia PS/2 de IBM y de su implementación del MicroCanal, parecía ser que el mercado de PC's se vería gobernado por las aplicaciones de OS/2, sin embargo, dada la enorme cantidad de recursos que demanda para ser ejecutado, el mercado volvió la mirada hacia el supuesta "descontinuado" DOS.

Para efectos de la presente propuesta se deben considerar factores como los descritos en el párrafo anterior para adquirir una plataforma que cumpla con los requerimientos de construcción de SE's y que además garantice cierta vida útil.

Hoy en día, cualquier fabricante de PC's compatibles (clones de IBM), que se precie de ser de primer nivel, debe ofrecer la posibilidad de hacer crecer la capacidad de memoria dinámica hasta 4 MB en la tarjeta principal, y hasta 8 MB con tarjetas auxiliares, que además puedan operar en modo extendido (XMS) o en modo expandido (EMS). Definitivamente se requiere de CPU's muy rápidos para lograr una buena capacidad en cómputo intensivo. Esto sólo es posible con plataformas cuyo procesdor central sea como

mínimo un 80386 o 80386sx, los cuales puedan operar de 16Mhz hasta33Mhz. Curiosamente, en las versiones que hoy se usan de DOS, no se obtiene una explotación máxima del conjunto de instrucciones de estos procesadores, aunque su velocidad ha probado ser un parámetro fundamental, incluso en el diseño de compiladores. Es sabido que algunas aplicaciones que hacen uso exhaustivo de gráficos, no pueden ejecutarse en plataformas que no contengan un coprocesador matemático integrado. Intel ofrece en su procesador 486 esta opción dentro del mismo encapsulado del CPU.

Un factor importante a considerar es el costo de estas máquinas. En el caso de las estaciones de trabajo en las que se pueden atender a 8 o 10 usuarios, es fácil dividir su costo entre ellos y saber que sólo se están gastando alrededor de $600.00 USD por usuario. Para las PC"s todavía es un punto en contra, a menos que se les compare con lo que costaría desarrollar en una macroplataforma (mainframe). Según datos recabados con los distribuidores mexicanos, una PS/2 modelo 70 A-21 cuesta alrededor de $14,000.00 USD; sin embargo, se trata de una plataforma de alto rendimiento. Su procesador central es un 80386 que opera a 25Mhz y dentro de su arquitectura cuenta con una memoria cache de 128k que incrementa considerablemente su desempeño. Así mismo, cuenta también con circuitería integrada a la tarjeta principal para manejo de monitores VGA de alta resolución y un disco duro de 120 MB. Estas características muestran un gran apego a las solicitadas para desarrollar SE's. Desde luego cualquier plataforma basada en procesadores Intel que cumplan estas características, puede ser utilizada en el laborarterio; incluso es sabido que algunos clones llegan acostar considerablemente menos que las máquinas IBM; sin embargo, es difícil conseguir clones cuya calidad de construcción garanticen operación libre de errores a velocidades mayores de 20Mhz. A estas frecuencias, el aislamiento electromagnético juega papeles definitivos y sólo los fabricante más finos lo consiguen.

El laboratorio de cómputo de la ENEP adquirió nuevo equipo. El equipo que arribó son 25 computadoras 486sx a 33Mhz. marca UNISYS con disco duro de 200 MB y monitor a color SuperVGA. Estas máquinas están ubicadas en el salón 7 donde antes se encontraba la minicomputadora HP-1000.

La configuración dispuesta para administrar el equipo de nueva adquisición es una red Ethernet con sistema operativo Netware ver 3.11, también se adiciona una nueva

configuración el equipo 386. Acer del salón 5 (cuya memoria se ha incrementado por medio de una red arcnet).

5.6.2.1.5 Sun.

El equipo Sun está constituido por dos estaciones de trabajo que funcionan como servidores y se integran por los elementos que a continuación se describen.

Un SPARCclassic Server con 48 MB. en memoria RAM, disco interno de 1.05 GB. FAST SCSI-2 Monitor color 16 ''.

Dos SPARCclassic X con 8 MB en memoria RAM y monitor de 16''.

Un SPARCstation 20 Modelo 51 TurboGX con monitor Color 17'', tarjeta de aceleración gráfica TurboGX, 1-MB Frame buffer, 32 MB de memoria RAM , procesador SuperSPARC A 50 MHz., 8 Kb de cache de datos y 16 kb de instrucciones 1 MB de superCache, puerto Ethernet par trenzado a 10 MB/seg, dos puertos seriales RS-232/RS-423 síncronos. Un puerto paralelo compatible Centronics. Bocina interna de 16-bits a 48 kHz , cuatro slots de expansión SBus (32-bits). Disco interno de 1.05 GB Fast SCSI-2.

Es conveniente puntalizar que el término SPARC arquitectura de rendimiento graduable se refiere a una computadora RISC de 32 bits creada por Sun microsystems.

El ambiente de usuario Solaris es un paquete integral de soluciones que permiten eficientizarel uso de la computadora, está basado en el uso de la interface gráfica OPEN LOOK.

Solaris está integrado por :

- Sistema operativo SunOS
- El ambiente OpenWindows.
- Un Conjunto de Herramientas gráficas.
- Manuales en línea.

Sistema Operativo:

Solaris 2.3 incluye el sistema operativo SunOS que es la versión UNIX de Sun Microsystems.

Ambiente OpenWindows:

Open windows 3.3 está incluido en solaris 2.3 y es el software que permite el manejo de gráficos y ventanas en el equipo. incluye el área de trabajo con sus menus de útilerias, programas y propiedades.

Herramientas Gráficas (Deskset) :

Estas son un conjunto de programas diseñados para trabajar y aprovechar el ambiente gráfico. Entre estas se encuentrán:

File Manager	El cual permite el manejo de archivos y directorios en forma gráfica.
Text Editor	Permite la creación y modificación de archivos de texto.
Mail tool	Herramienta gráfica através de la cual puede enviarse correo a otra persona en la red.
Calendar	Calendario que permite llevar citas y Manager notas.
Clock	Programa que despliega un reloj en la pantalla con todas sus opciones.
Calculator.	Se despliega una calculadora para poder hacer operaciones.
Paint tolt	permite imprimir archivos.
icon Editor	Permite editar iconos.
AnswerBook.	Libro de respuestas. Manuales.

5.6.2.2 Software

Al igual que en la sección anterior, se habrán de marcar los atributos requeridos para sugerir una o varias herramientas de programación para construcción de SE's. Asímismo, se indicarán herramientas que den la posibilidad de alcanzar los objetivos a cumplir por el laboratorio.

Tal y como se describió en el capitulo 4, existen diferentes tipos de herramientas a diferentes niveles. Por lo que respecta a los lenguajes de programación, resulta

indispensable tener en el laboratorio por lo menos un intertprete de LISP. Esta herramienta permitirá trabajar al alumno en conceptos fundamentales de IA y SE's , tales como métodos de búsqueda, métodos de resolución de problemas, etc. Aunque LISP es un lenguaje que apareció hace ya muchos años, todavía no existe un estándar como en FORTRAN o Pascal, de ahí que seleccionar ¿que intérprete o complidador adquirir?, no es una tarea simple. Las cualidades de los compiladores e intérpretes varían de proveedor a proveedor, por lo que resulta mas simple escoger el software de acuerdo a la plataforma en la que se desea ejecutar.

Como se ha mostrado, las características que debe tener una herramienta son varias, entre ellas quizá tres de las más importantes sean las interfaces externas, el método de inferencia y el esquema de representación de conocimiento. La herramienta idónea, es aquella que ofrezca un esquema híbrido de representación de conocimiento. Al respecto, se planteó que el dominio de un problema real, se puede representar como un globo inflado con aire y el esquema de representación como un recipiente en cual se pretende guardar dicho globo. La geometría de tal recipiente marcará que tan fácil es de guardar el globo. Si se asocia cada esquema de representación con una geometría específica, quizá la herramienta híbrida ideal se vea como un cilindro, mientras que la que presenta marcos y reglas se vea como un cubo y la que sólo opera con reglas, se vea como un tetraedro. Así, se necesitará "torturar" más o menos el conocimiento que describe el problema, de acuerdo al el esquema de representación que se escoja.

Otra característica importante que se busca en una herramienta, es su portabilidad. Afortunadamente existen compiladores de LISP, versiones de OPS5 y shells que operan en las tres plataformas computacionales propuestas.

En cuanto a PROLOG se refiere DEC, no reporta ningún producto al respecto, lo cual no es raro dado que la tradición americana en procesamiento simbólico se basa en LISP, mientras que en Europa los investigadores prefieren trabajar en PROLOG.

Una herramienta que resulta muy interesante son los intérpretes de reglas como OPS5. DEC tiene diponible el VAX OPS5, un intérprete de reglas con una estrategia de inferencia de encadenamiento hacia adelante, que permite un desarrollo muy rápido de prototipos de SE's y sistemas heurísticos de apoyo para la toma de desiciones. Su sintáxis es muy similar

a la de LISP y su potencial ha sido probado en aplicaciones tan exitosas como R1 (hoy XCON), un sistema experto configurador de VAX ese que sigue dando dividendos importantes.

Por lo que a shells se refiere, DEC presenta el VAX Decision Expert una heramienta con un esquema de representación basado en reglas, que además permite la construcción gráfica de árboles de decisión, selección de la estrategia de inferencia (Encadenamiento hacia atrás y hacia adelante) y facilidades para la construccion de interfaces de usuario.

Para las palatafomas basadas en el Intel 80X86 ó (PC's y PS/2), también existe una amplia gama de herramientas. Si se trata de intérpretes de LISP, el candidato indiscutible es GoldenCommon LISP, una Heredera directa de intérpretes originales de macroplataformas la cual ha podido conservar gran parte de su potencialidad para trabajar en PC's. Por supuesto estas cualidades tiene asociadas un costo, el cual se refleja en la necesidad de instalar 4MB de RAM para poder correr la versión intérprete y 6 MB de RAM para la version que compila. Este mismo LISP sirve de base para uno de los más poderosos shells que existen en PC, el GoldWorks II.

Para cubrir los requerimientos de programación lógica, existen disponibles varios intérpretes de PROLOG para PC's. Los más fáciles de reconocer en las revistas especializadas son el Turbo PROLOG y el Arity Prolog. El paquete de Borland, prácticamente ha desaparecido del mercado y su versión más reciente fué la 2.0. Al igual que en LISP, no existe un estandar en PROLOG, de ahí que según el proveedor se encuentren unas u otras facilidades. Para turbo PROLOG, las limitaciones son similares a las que había en Turbo Pascal 3.0. El editor tiene un tamaño máximo de 64K de buffer. Su interface al exterior es siempre a través de "C", con quien entabla una comunicación natural. Para los expertos, Turbo Prolog es un "juguete interesante", sobre todo, tomando en cuenta que no cuesta más de $120 USD. La opinión generalizada indica que para hacer desarrollos serios Arity PROLOG contiene apenas lo indispensable.

Uno de los shells más comentados entre los expertos es el Nexpert Object. Esta es una herramienta que se acerca considerablemente a la descripción ideal. Entre sus cualidades se encuntran una serie de interfaces con bases de datos y hojas de cálculo, de acuerdo a la plataforma en la que opere, ademas de exelentes facilidades gráficas de depuración y para

construcción de interfaces de usuario final. También puede operar con las dos estrategias de inferencia conocidas para sistemas de reglas. Asimismo, su esquema de representación de conocimiento se ve complementado con la utilización de objetos con funciones muy similares a la de los marcos. Otra caraterística muy importante es que existe una versión para cada una de las plataformas computacionales que se han sugerido.

Para las plataformas Macimosh, gracias a la utilización de la tarjeta de Texas Instruments, se cuenta prácticamente con el mismo software que para las estaciones Explorer y MicrolExplorer. Desde luego, la recomendación natural es Personal Consultant Plus, un shell que también corre en PC's y que ha probado ser un constructor serio de aplicaciones. A modo de referencia, se puede mencionar que los desarrollos realizados por el CETEC del ITESM Campus Monterrey, en su mayoría, han sido hechos en PC Plus. Un valor agregado a este producto es que contiene un intérprete de Scheme, un dialecto de LISP, el cual es un excelente auxiliar del shell, ya que muchas funciones que resultan tediosas de desarrollar a base de reglas, pueden simplificarse escribiendolas en Scheme.

En la figura siguiente se presenta un cuadro de las herramientas recomendadas para cada plataforma que se encotraría en el laboratorio, de acuerdo a las recomendaciones citadas.

HERRAMIENTAS DE DESARROLLO DE SISTEMAS EXPERTOS PARA ORDENADORES PERSONALES.

Programa	Publicado por	Oredenador	tipo
CxPERT	Software Plus	IBM PC	Marcos, Reglas.
DAISY	Lithp Systems	IBM PC	Marcos, Reglas, Redes Semanlicas.
ESI	Abacus Programming	IBM PC	Reglas.
EXPERT EASE	Expert systems.	IBM PC	Inducción.
EXPERT EDGE	Human Edge software	IBM PC	Reglas.
EXSYS	Exsys, Inc.	IBM PC	Reglas.

Programa	Distribuidor	Ordenador	Tipo
1st Class.	Programs in Motion, Inc.	IBM PC	Reglas o Inducción.
Methods	Digitalk	IBM PC	Basado en caracteres.
Micro Expert.	McGraw-Hill.	IBM PC	Reglas.
OPS5	ArteIligence, Inc.	IBM PC	Reglas.
NEXPERT	Neuron Data Inc.	IBM PC/AT, Apple, Macintosh	Reglas.
VP-EXPERT	Peperhack Software Inc.	IBM PC	Reglas o inducción.
DUCK	Smart Systems Technology	Sun, Xerox, Symbolics.	Cálculo de predicados
TWAICE	Logicware Inc.	IBM. Sun. Apollo.	Reglas.

5.6.3 Habilidades y Conocimientos.

Las asignaturas que se apoyen en el laboratorio, deberán tener sus correspondientes temarios y objetivos, sin embargo, el laboratorio debe proporcionar los recursos para desarrollar prácticas y prototipos bajo diferentes perspectivas. Para ello se propone que los temas a tratar preferentemente sean:

- Métodos de Solución de Problemas.

- Introducción a LISP.
- Introducción a PROLOG.
- Representación de Conocimiento y Esquemas de Representación.
- Arquitectura de SE's (Mecanismos de inferencia, Control y Módulos Auxiliares.)
- Proceso de Construcción y Ciclo de Vida de los SE's.
- Criterios de Selección de Herramientas.

Algunos temás adicionales de interes son:

- Procesamiento de lenguaje Natural.
- Reconocimiento de Patrones.
- Redes Neuronales.
- Algoritmos Genéticos.
- Lógica Borrosa.
- Programación Orientada a Objetos.

5.6.4 Estándares de realización.

Los estándares de realización se refieren a la calidad de trabajos que se realizan en el laboratorio. Para poder establecer estas referencias, hay que distinguir primero las áreas en las que se trabajar .

5.6.4.1 Tesistas.

Los alumnos que trabajen en su opción terminal, tiene la oportunidad de atraer proyectos al laboratorio. Estos pueden funcionar como los cimientos de anteriores trabajos, que formalmente queden establecidos como proyectos de cooperación. En el depto. de Ingeniería en Computación, existen ya marcadas las exigencias de calidad en los trabajos de tesis, aunque estas pueden ser revisadas de acuerdo al proyecto y a la empresa que lo patrocine.

En los proyectos orientados a la investigación básica, su calidad estará acorde a las políticas y normas establecidas por el depto. de Ingeniría en Computación. Es recomendable que los alumnos que deseen seguir a postgrados, trabajen en proyectos introductorios a su futura actividad académica. Esta orientación se puede conseguir através de intercambios con otras instituciones dentro y fuera del país.

5.6.4.2 Alumnos de asignatura.

Para las prácticas y proyectos que realicen los alumnos que están cursando alguna de las asignaturas del área, los estándares de calidad y niveles de exigencia serán establecidos por los instructores y profesores en turno, de acuerdo a los objetivos establecidos en los temarios y en concordancia con las políticas y normas del depto. de Ingeniería en Computación.

5.6.4.3 Formas de medir el éxito del Laboratorio.

Existen diversas formas de medir el éxito o la eficiencia de un proceso, pero cuando se trata de un proceso de enseñanza, los múltiples elementos de carácter humano que intervienen, pueden dotar de cierta subjetividad a la evaluación. Una forma de evaluar un proceso, es determinar el grado con el que se han obtenido los objetivos generales. Para ello se proponen las siguientes formas de evaluación:

Encuestas.

La elaboración de encuestas al finalizar los cursos o trabajos en el laboratorio, tiene por objeto conocer si:

Los alumnos saben distinguir los diferentes tipos de problemas que son apropiados de resolver con SE's, con lo que se cumpliría el primer objetivo.

Los alumnos saben reconocer qué, herramienta es la apropiada para un problema en cuestión, y porqué, trabajaron con ella, con lo cual se probaría el segundo objetivo.

Los alumnos reconocen las tres áreas básicas de desarrollo profesional (Investigación en Ciencias Básicas, Investigación Aplicada y Desarrollos Comerciales) y en cual de ellas se ubica el trabajo que acaban de realizar, probando así el cuarto objetivo

Estadísticas.

Para el resto de los objetivos generales, resulta recomendable elaborar estadísticas que informen:

cuántos proyectos han sido realizados en el laboratorio; cuántos proyectos han logrado la calidad suficiente para incluirse en la biblioteca de casos, y cuantos han quedado fuera, cuántas exposiciones, simposiums, ciclos de conferencias, etc, se han realizado en la ENEP con temas afines a IA y SE's; cuántos artículos han sido publicados o enviados a congresos, exposiciones, simposiums o publicaciones de otras instituciones. Asimismo, es bueno saber cuántas empresas se han acercado al laboratorio para realizar proyectos, cursos o donaciones y qué difusión se le ha dado a los trabajos del laboratorio en el medio empresarial.

Establecer las características de cada una de las propuestas implica una labor interdisciplinaria, dado lo cual se sugiere la asesoría de los departamentos de Psicología y Didáctica. Además se propone como proyecto para alumnos del Servicio Social, la elaboración y aplicación de los mecanismos descritos.

CAPITULO 6

CONCLUSIONES

Este capítulo se ha seccionado en tres partes para dar claridad a las ideas finales del trabajo. La primera parte se refiere a las aportaciones que se generaron con la realización de este estudio y por qué se consideran como tales. En la segunda parte se mencionan cuáles fueron las limitaciones a las que se sujetó el estudio y cuáles fueron los motivos que originaron los ajustes. En la tercera parte se hacen los comentarios finales, en los que aparece una opinión personalizada sobre la elaboración del trabajo y las experiencias y los conocimientos recibidos.

6.1 Contribuciones.

Las contribuciones principales de este trabajo se pueden describir como sigue:

Se ha obtenido un documento que plantea los requerimientos para la instalación de un Laboratorio de Desarrollo de Sistemas Expertos para los alumnos de ingeniería en Computación, en cuyo plan de estudios se incluyen asignaturas al respecto. Así, es posible asimilar el presente trabajo como un proyecto formal y base de la infraestructura definitiva para la creación de una disciplina académica que las autoridades universitarias decidan constituir en su momento. Este planteamiento describe objetivos, así como procedimientos recomendables para la operación del laboratorio. Al mismo tiempo se mencionan los recursos materiales y humanos que han de apoyar los trabajos desarrollados y cuáles son las actividades extra-académicas que pueden complementar el aprendizaje del alumno. Por otra parte, también se marcan algunas recomendaciones para atraer recursos hacia el laboratorio y lograr a mediano plazo financiamientos extra presupuestales através de proyectos de cooperación y asesorías. De igual forma, se proponen mecanismos para garantizar el robustecimiento tecnológico del laboratorio y garantizar la superación académica.

Igualmente, se ha realizado una revisión de los conceptos teóricos sobre construcción de sistemas expertos. Su disposición a manera de antología puede constituir una guía de estudio para introducir al alumno al tema.

Se presenta también en este trabajo, un panorama del mercado que tienen los SE's en nuestro país, mostrando cuáles son los requerimientos y necesidades de los diversos sectores que intervienen en el proceso productivo, así como las perspectivas académicas y de empleo de los egresados con esta especialidad. Los pronósticos que se muestran, cumplen con una trayectoria de origen histórico. A través de una semblanza, se ofrecen explicaciones de las circunstancias actuales y cual sería el escenario futuro a mediano plazo, de acuerdo al ritmo de trabajo que hoy existe.

Respecto a la revisión bibliográfica que se realizó, se pudo constatar la disponibilidad de algunos títulos importantes en nuestro país y a los cuales los alumnos pueden acudir a profundizar y revisar temas complementarios al presente trabajo. También algunos de los análisis presentados, que originalmente se extrajeron de fuentes bibliográficas, fueron asimilados a través de experiencias personales que permitieron resaltar el carácter práctico de los mismos.

En conclusión, el trabajo ofrece una síntesis panorámica del acontecer nacional en el ámbito de los SE's, con lo cual se logró una propuesta concreta y recomendaciones para la instalación de un laboratorio de desarrollo que ofrezca una formación académica que reduzca la distancia entre la Universidad y los sectores productivos, en beneficio del desarrollo nacional.

6.2 Limitaciones

Todos los trabajos de investigación, de acuerdo a su nivel se encuentran delimitados por sus objetivos y alcances originalmente plasmados en el anteproyecto o propuesta inicial. Durante la elaboración de este trabajo, se requirió de hacer ajustes en el procesamiento de la información para garantizar el enfoque central. En los siguiente párrafos, se describen algunos de éstos ajustes y cuáles fueron las causas de su revisión.

En los últimos años, se ha escrito un gran volumen de libros, artículos y bibliografía en general, acerca de SE's. La mayoría de estos trabajos son producto de varios años de investigación. Por ende, profundizar en algunos de los temas aquí tratados no hubiese reportado una contribución significativa y si hubiesen desviado el presente trabajo de su objetivo central. Por ello, en repetidas ocasiones se prefirió remitir al lector a las fuentes originales. Así mismo, es importante afirmar que los conceptos que aquí se presentaron no

describen totalmente la tecnología de SE's, sin embargo, muestran los principios para dirigir un estudio posterior, más estructuralmente detallado y más profundo.

Es innegable que el planteamiento general de los sistemas de almacenamiento, procesamiento de información y procesamiento de conocimiento, implica una interrelación estrecha entre diferentes tecnologías. Hace unos diez años, los SE's eran el objeto central de estudio de algunas disciplinas científicas. Hoy en día, éstos se han convertido en un módulo más de los complejos sistemas de información. Es por eso que al hablar de construcción de SE's, es inevitable hablar de procesamiento de lenguaje natural, sistemas Expertos conexionistas, procesamiento de imágenes, visión, programación orientada a objetos, bases de datos inteligentes, redes neuronales, algoritmos genéticos, conjuntos difusos (fuzzysets) etc.. Por eso, el presente trabajo plantea una visión global, con enfoque sistémico invitando al lector a profundizar en los temas pertinentes.

La recopilación de información a través de entrevistas, permite conocer con cierto detalle las opiniones de los entrevistados, pero no facilita la recolección masiva de información, por lo que resulta más efectivo la elaboración de estadísticas. Sin embargo, los resultados obtenidos tiene una confiabilidad significativa, considerando el tipo de trabajo, las empresas y/o centros de investigación y los proyectos en los que están involucrados los entrevistados.

Asimismo, no fue posible evaluar todas las herramientas de programación y computacionales que se mencionaron. De ahí que varios de los criterios de selección, se fundamentaran en bibliografía especializada y no con base en la experiencia personal. Sin embargo, los reportes obtenidos de revistas ofrecen un alto grado de confiabilidad. Por eso, las conclusiones obtenidas a partir de éstos, se han utilizado extensamente en este estudio.

El hecho de presentar un estudio cualitativo y no cuantitativo de los beneficios que puede reportar la utilización de SE's en nuestro país, obedece fundamentalmente a dos factores:

Primero: lo novedoso de esta tecnología no ha permitido que las aplicaciones abandonen los centros de investigación y los ámbitos académicos para incorporarse totalmente al sistema productivo y por consiguiente, no existen registros contables de la rentabilidad de éstos.

Segundo: aún en el caso de aquellas empresas pioneras que ya han utilizado SE's, la información sobre sus proyectos guarda una considerable confidencialidad, ya que se asume como verdad general, que los SE's son un arma competitiva de asombroso alcance. Sin embargo, la información que aquí se presento, deja vislumbrar la imperiosa necesidad de que los científicos, especialistas y profesionistas del ramo, se involucren seriamente con una tecnología que regirá a los sistemas de información más adelante.

6.3 Extensiones

En esta sección se proponen temas de estudio y proyectos que pudiesen tomar a este trabajo como punto de partida, así como tópicos complementarios a los conceptos que aquí se trataron.

La recopilación de información acerca de investigadores y profesionales a quienes consultar, proyectos, centros de investigación, Universidades, proveedores de equipo, etc., hubiesen podido estructurarse más finamente de haber diseñado cuestionarios específicos, de acuerdo al sector encuestado. Asimismo, la consulta a los bancos de información y el establecimiento de mejores canales de comunicación con organismos como el INEGI (Instituto Nacional de Estadística Geografía e Informática), el CONACyT (Consejo Nacional de Ciencia Y Tecnología), la SEP (Secretaría de Educación Pública), la SMIA (Sociedad Mexicana de Inteligencia Artificial), el IIE (Institutode Investigaciones Eléctricas de la Comisión Federal de Electricidad), etc., hubiese enriquecido la información. Una contribución muy importante, puede ser la generación de un Catálogo Nacional de Entidades Relacionadas con SE's, donde se distingan cada uno de los sectores arriba citados y se establezcan las formas de entablar comunicación con cada uno de ellos. Desde luego, dada la envergadura del proyecto se propone un trabajo interorganizacional, con una considerable aportación de recursos por parte de los involucrados.

Respecto a la evaluación de herramientas de programación y computacionales es posible proponer criterios para diseñar "pruebas de banco" y aplicarlas a algunas de las herramientas que ciertas organizaciones ya tienen a su disposición en nuestro país. Para tal efecto, hace falta establecer los canales de comunicación apropiados. El presentar un proyecto debidamente respaldado por la ENEP, puede facilitar su consecución. Aún así, la evaluación no será completa, pero servirá para establecer parámetros de comparación con

los mostrados en la bibliografía. Además, la experiencia adquirida en dicho trabajo permitirá aclarar conceptos prácticos de gran utilidad. (Como ¿qué tanto afecta a la evolución del proyecto y a la productividad del programador las facilidades de edición-compilación, etc.?)

Por lo que al contenido académico del presente estudio corresponde, se propone profundizar en la definición de criterios de selección de arquitecturas de construcción de SE's, de acuerdo a la tipificación de problemas que existe y mostrar cómo un esquema de representación de conocimiento puede resultar más apropiado que otro al tratar de resolver el mismo problema. Al respecto, cabe señalar que de acuerdo al enfoque sistémico que se pretende dar a la tecnología de SE's, probablemente una de las arquitecturas que se incorpore más naturalmente a este proceso sea la arquitectura de pizarrón, por lo cual se considera pertinente elaborar un estudio más detallado al respecto.

Un tema también relevante en la construcción de SE's de tamaño medio y mayor, es el de análisis de consistencia y mantenimiento de la verdad en la base de conocimiento. De la misma manera como en la programación tradicional, se han propuesto metodologías como la programación estructurada. Todavía se buscan soluciones que permitan depurar y mantener de una manera estándar las grandes bases de conocimiento.

Otro aspecto que resulta fundamental en la aceptación de la tecnología de SE's, es el que se refiere a la construcción de interfaces de usuario final. Utilizando técnicas de IA como procesamiento de lenguaje natural, reconocimiento de patrones, visión etc. y conceptos como el de implementar SE's como administradores de bases de datos tradicionales, se facilita la construcción de interfaces mas "humanas". Estas se han convertido en la piedra angular para la colocación de dichos sistemas en las actividades cotidianas de las empresas. Las interfaces con carácter "humano" han probado reducir la resistencia al cambio y el impacto sobre la cultura organizacional de los usuarios finales. Por eso profundizar en estos tópicos puede redituar considerables beneficios.

6.4 Comentarios Finales

El presente trabajo brindó la oportunidad de constatar que el estudio de la Inteligencia Artificial y los Sistemas Expertos no sólo es fascinante por su carácter interdisciplinario, sino porque además constituye parte importante de la tecnología que gobernará los

desarrollos de sistemas en un futuro cercano. En general, podemos decir que los sistemas automáticos de almacenamiento, procesamiento y análisis de información han sido propuestos con la idea de permitir al hombre desarrollar tareas mas creativas y enriquecedoras, dejando a las computadoras el trabajo "rutinario". Tener a nuestro servicio "máquinas inteligentes " es una responsabilidad de grandes dimensiones.

Tomo Peters menciona en el prologo del libro The Raise of the Expert Company de E. Feigenbaum, P. McCorduck y H.Penny, que " los estandares de vida han subido no por que la gente trabaje más, sino porque trabaja más inteligentemente." Esta frase deja entre ver que la utilización de la tecnologia de Sistemas Expertos por si misma, no ofrecerá importantes beneficios. Es la utilización racional de la misma, la que proveerá de logros importantes.

Nuestro pais por consecuencias históricas, ha participado casi siempre con retraso en los avances tecnológicos. A decir de los investigadores a quienes se les consultó, el reto de los profesionistas que trabajen con la citada tecnología, no es el de aventajar los descubrimientos realizados en países con mayores recursos que el nuestro, sino el de comprender y dominar una tecnología que pueda contribuir notablemente al desarrollo de los sectores productivos de la nación, asi como el de formar recursos humanos que nos alejen sensiblemente de la dependencia tecnológica.

A diferencia de lo que pudimos apreciar en los ámbitos empresariales, la gente de la academia esta ávida de compartir experiencias e información. Sin embargo, algunos esfuerzos como los de la SMIA no han mostrado resultados más contundentes por la falta de integración con los sectores productivos, una circunstancia que se ha presentado en otros países y que se repite en el nuestro; como lo ha señalado el Dr. Simon en diversas ocasiones, "es la notable carencia de recursos humanos dedicados a conjuntar los esfuerzos de los investigadores en ciencias básicas con las necesidades de las empresas e industrias."

En este momento y de acuerdo a lo novedoso de los Sistemas Expertos, la Universidad Nacional Autónoma tiene la oportunidad de participar abiertamente en el desarrollo de los recursos que acorto plazo nuestro país ha de necesitar para mantener su crecimiento. El Laboratorio de Desarrollo de Sistemas Expertos brinda la oportunidad de formar profesionales comprometidos con su realidad para dirigir los destinos tecnológicos del

país, pero fundamentalmente, debe ser un apoyo decisivo y unificador del esfuerzo que otras instituciones de enseñanza superior ya han iniciado.

A lo largo del trabajo se detectaron ciertas áreas de especialización en cada una de las Instituciones de Educación Superior que se encuestaron. Nuestro laboratorio se puede ubicar en alguna de las áreas no cubiertas y contribuir así notablemente a complementar el esfuerzo nacional.

Cabe aclarar que el tiempo que nos dedicaron los científicos, investigadores, profesores, profesionistas, proveedores y vendedores de equipo que fueron consultados, constituyó uno de las principales fuentes de información para éste trabajo. Durante su ejecución, pudimos constatar el promisorio interés que todos y cada uno de ellos mostraron hacia el presente estudio, debido a que consideran que cualquier esfuerzo por retratar y mostrar la labor de la comunidad de Inteligencia Artificial en nuestro país, será importante para lograr una unidad más sólida y productiva, tal como se demostró en Iberamia 95 y en el Simposium Internacional de IA de ITESMCMTY.

APENDICE "A"
PLANES DE ESTUDIO RELACIONADOS CON SISTEMAS EXPERTOS E IA EN DIFERENTES UNIVERSIDADES

UNIVERSIDAD: Universidad Nacional Autónoma de México.

LICENCIATURAS

CARRERA: Ingenieria en Computación.
MATERIAS: Inteligencia Artificial, Sistemas Expertos, reconocimiento de patrones, Robótica, Procesamiento Digital de Imágenes.

MAESTRIAS

NOMBRE: Ciencias de la Computación.
MATERIAS: Procesamiento Digital de Imagenes, Reconocimiento de Patrones, Visión Computacional, Representación del Conocimiento, Construcción de Sistemas Expertos, Tratamiento del Leguaje Natural.

NOMBRE: Ingenieria Informatica.
MATERIAS: Inteligencia Artificial Distribuida, Algoritmos Genéticos, Lógica Borrosa, Neurocontrol y Robótica, Temas avanzados de neurocomputación.

NOMBRE: Electrica.
MATERIAS: Neurocomputación.

UNIVERSIDAD: Instituto Tecnológico y de Estudios Superiores de Monterrey.

LICENCIATURAS

CARRERA: Licenciatura en Sistemas de Computación Administrativa
MATERIAS: Sistemas para la toma de desiciones y Sistemas Expertos.

CARRERA: Ingenieria en Sistemas Computacionales
MATERIAS: Paradigmas orientado a objetos.

MAESTRIAS

NOMBRE: Ciencias Computacionales (especialidad en IA)
MATERIAS: Reconocimiento de Patrones, Ingeniería del Conocimiento, Sistemas Expertos, Inteligencia Artificial, Programación Orientada a Objetos y Simbólica

NOMBRE: Sistemas de Información
MATERIAS: Programación Orientada a Objetos, Sistemas Expertos.

UNIVERSIDAD: Instituto Tecnológico Autónomo de México

LICENCIATURAS
CARRERA: Ingeniería en Computación
MATERIAS: Inteligencia Artificial, Paradigma Orientado a Objetos.

DIPLOMADOS
NOMBRE: Inteligencia Artificial y Sistemas Expertos

UNIVERSIDAD: Iberoamericana.

LICENCIATURAS
CARRERA: Ingeniería Electrónica y de Comunicaciones
MATERIAS: Sistemas Expertos y Laboratorio

CARRERA: Licenciatura en Sistemas Computarizados e Informática
MATERIAS: Sistemas Expertos y Laboratorio

MAESTRIAS
NOMBRE: Sistemas, Planeación e Informática
MATERIAS: Representación de Conocimiento y Estrategias de Búsqueda, Sistemas Expertos, Inteligencia Artificial, Seminario de Sistemas Expertos.

UNIVERSIDAD: La Salle

LICENCIATURAS
CARRERA: Ingeniería Cibernética y en Sistemas Computacionales
MATERIAS: Inteligencia Artificial.

UNIVERSIDAD: Universidad Veracruzana LANIA A.C.

MAESTRIAS
NOMBRE: Inteligencia Artificial.
TEMAS . Sistemas basados en el conocimiento, lenguaje natural, aprendizaje,visión, Robotica, planificación, sistemas distribuidos, bases de datos deductivas, bases de datos orientadas a objetos, programación logica funcional.

TEMARIOS DE LAS ASIGNATURAS DE INTELIGENCIA ARTIFICIAL Y SISTEMAS EXPERTOS.

PROGRAMA DE LA ASIGNATURA: Inteligencia Artificial.
CLAVE: 0406
DURACION DEL CURSO: Semanas 16. SEMESTRE: Noveno.
horas: 64
HORAS A LA SEMANA: Teoria 04. CARACTER: Obligatoria
Practica 00

OBJETIVO DEL CURSO : El alumno conocerá en forma general el curso de Inteligencia Artificial y utilizará estas técnicas para resolver problemas básicos.

TEMAS

NUMERO	NOMBRE	HORAS
I	Lenguajes de programacion LISP y PROLOG.	16
II	Sistemas Expertos.	12
III	Representación de problemas en Inteligencia Artificial.	12
IV	Uso y Modificación del Conocimiento.	12
V	Aplicaciones Especificas.	12
		64

I LENGUAJES DE PROGRAMACION LISP

ANTECEDENTES: Compiladores.
OBJETIVO: El alumno realizará problemas de Inteligencia Artificial en lenguajes LISP y PROLOG.
CONTENIDO:

I.1 Estructuras básicas.
I.2 Programas en LISP y PROLOG.

I. SISTEMAS EXPERTOS.

ANTECEDENTES: Compiladores, Bases de datos.
OBJETIVO: El alumno conocerá las características generales y alcances de los sistemas denominados expertos.
CONTENIDO:

II.1 Caracteristicas Generales.
II.2 Objetivos.
II.3 Perspectivas.

OBJETIVO DEL CURSO: El alumno conocerá en forma general el curso de Inteligencia Artificial y utilizará estas técnicas para resolver problemas básicos.

TEMAS

NUMERO	NOMBRE	HORAS
I	Introduccion.	4
II	Construcción de Sistemas Expertos	12
III	Herramientas para sistemas expertos..	12
IV	Construcción de Herramientas para sistemas expertos.	20
V	Desarrollo de una aplicación de un sistema experto.	16
		64

I INTRODUCCION

ANTECEDENTES. Ingeniería de programación e Inteligencia Artificial.
OBJETIVO. El alumno explicará el desarrollo historico de los sistemas expertos. También identificará y explicará las faces del ciclo de vida de los mismos, así como sus partes funcionales
CONTENIDO:

I.1 Desarrollo Historico de los sistemas expertos.
I.2 El ciclo de vida de los sistemas expertos.
I.3 Estructura de un sistema experto.

II CONSTRUCCION DE SISTEMAS EXPERTOS.

ANTECEDENTES: Inteligencia Artificial y Estructuras Discretas.
OBJETIVO: El alumno explicará el proceso de desarrollo de los sistemas expertos.

CONTENIDO:

II.1 Identificación del problema.
II.2 Conceptualización.
II.3 Transformación de la representación del conocimiento.
II.4 Instrumentación.
II.5 Pruebas y evaluación.

III. HERRAMIENTAS PARA SISTEMAS EXPERTOS.

ANTECEDENTES: Inteligencia Artificial.
OBJETIVO: El alumno identificará y aplicará las herramientas para el desarrollo de sistemas expertos.
CONTENIDO:

III.1 Lenguajes para el desarrollo de sistemas expertos

III. REPRESENTACION DE PROBLEMAS EN INTELIGENCIA ARTIFICIAL

ANTECEDENTES: Compiladores.
OBJETIVO: El alumno utilizará diferentes técnicas de representación para resolver problemas.

CONTENIDO:

III.1 Métodos de Inteligencia Artificial para resolver problemas.
III.2 Representación de espacios de estado.
III.3 Métodos de Busqueda en un espacio de estado.
III.4 Representación reducida del problema.
III.5 Métodos de busqueda en representación reducida del problema.
III.6 Representación en forma de procedimientos o rutinas.
III.7 Redes Semánticas.
III.8 Cuadros de Minsky.

IV. USO Y MODIFICACION DEL CONOCIMIENTO.

ANTECEDENTES: Compiladores.
OBJETIVO: El alumno conoceralas caracteristicas y alcances de las técnicas de aprendizaje y resolución general de problemas.

CONTENIDO:

IV.1 Programa que resuelva problemas generales.
IV.2 Programa que resuelva problemas generales con premisas.
IV.3 Sistema de creencias.
IV.4 Deducción Automática.
IV.5 Planeación.
IV.6 Aprendizaje e indiferencia deductiva.

V. APLICACIONES ESPECIFICAS

ANTECEDENTES: Compiladores.
OBJETIVO. El alumno conocerá casos especificos de programas donde se utilizarán las técnicas de Inteligencia artificial en diferentes areas de aplicación.

CONTENIDO:

V.1 Visión. Algoritmos para visión.
V.2 Comprensión del lenguaje.
V.3 Ciencia.
V.4 Medicina.
V.5 Educación.
V.6 Programación.
V.7 Conocimientos.

PROGRAMA DE LA ASIGNATURA: Sistemas Expertos.
CLAVE: 2138.
DURACION DEL CURSO: Semanas 16. SEMESTRE: Noveno o Décimo.
horas. 64.
HORAS A LA SEMANA: Teoria 04. CARACTER: Optativa.
Practica 00.

III.2 Esqueletos de Sistemas Expertos
III 3 Maquinas Lisp
III.4 Herramientas para la adquisición del conocimiento

IV. DESARROLLO DE HERRAMIENTAS PARA SISTEMAS EXPERTOS.

ANTECEDENTES: Inteligencia Artificial.
OBJETIVO: El alumno construirá utilerías para una mejor explotación de los sistemas expertos

CONTENIDO:

Esta parte depende mucho de la experiencia e interes del profesor.
Ejemplos de utilerías para una mejor explotación de sistemas expertos puede ser:
a) Métodos y herramientas automatizados o semiautomatizados para la adquisición del conocimiento de los expertos; como pueden ser una interfaz de lenguaje natural, herramientas, gráficas, etc.
b) Consultas de caracter general de la base de conocimientos lo cual implica desarrollar una utilería que haga preguntas de caracter natural con sus respectivos informes.
c) Interfaz amistosa al usuario, lo cual puede implicar el desarrollo de utilerías para menus en líneas, únicos y desglosados, ventanas con o sin encabezado y con o sin pie, deslizamiento vertical y horizontal , así como pantallas de captura.

V. DESARROLLO DE UNA APLICACIÓN DE UN SISTEMA EXPERTO.

ANTECEDENTES: Inteligencia Artificial.
OBJETIVO: El alumno aplicará métodos y herramientas para el desarrollo de una aplicación práctica de un sistema experto.
CONTENIDO:

Esta parte es dependiente de los proyectos presentados por equipos de alumnos que fueron aprobados por el profesor

CURSOS

Los encargados de programar cursos, sobre computación en general es departamento de educación continua, en donde se nos informo que en no tenian programados cursos relacionados con el tema.

APENDICE "B"
EJEMPLO DE UNPROGRAMA EN PROLOG

```
DOMAINS

 lista=symbol*
 LIST = SYMBOL*

DATABASE

 info(symbol,lista).
 si(symbol).
 no(symbol).

PREDICATES

 maxlen(LIST,INTEGER,INTEGER)
 listlen(LIST,INTEGER)
 writelist(INTEGER,INTEGER,LIST)
 write_list(INTEGER,LIST)
 write_list2(LIST)
 append(LIST,LIST,LIST)
 repeat
 index(LIST,INTEGER,SYMBOL)

CLAUSES

 index([X|_],1,X):- !.
 index([_|L],N,X):- N>1,N1=N-1,index(L,N1,X).
 append([],L,L).
 append([Ah|At],B,[Ah|C]):-append(At,B,C).
 maxlen([H|T],MAX,MAX1):- str_len(H,LEN), LEN>MAX,!, maxlen(T,LEN,MAX1).
 maxlen([_|T],MAX,MAX1):-maxlen(T,MAX,MAX1).
 maxlen([],LEN,LEN).
 listlen([],0).
 listlen([_|T],N):-listlen(T,X), N=X+1.
 writelist(_,_,[]).
 writelist(LI,ANTKOL,[H|T]):- field_str(LI,0,ANTKOL,H),
                              LI=LI+1,writelist(LI,ANTKOL,T).
 repeat. repeat:-repeat.
 write_list(_,[]).
 write_list(_,[X]):-!,write(X).
 write_list(4,[H|T]):-!,write(H),nl,write_list(0,T).
 write_list(3,[H|T]):-str_len(H,LEN),LEN>13,!,write(H),nl,write_list(0,T).
```

```
write_list(N,[H|T]):-str_len(H,LEN),LEN>13,!,N1=N+2,writef("%-27
                    ",H),write_list(N1,T).
write_list(N,[H|T]):-N1=N+1,writef("%-13 ",H),write_list(N1,T).
write_list2([]).
write_list2([H|T]):-write(H,' '),write_list2(T).
```

PREDICATES.

```
start.
anadir(symbol,lista,lista).
preguntar.
analizar(symbol).
checarbaseno(lista).
intentar(symbol,lista).
procesar(symbol,symbol).
escribirlista(lista,integer).
purgar.
```

CLAUSES.

```
start if consult ("musica.txt"),fail.
start if preguntar,purgar.
anadir(X,L,[X|L]).
preguntar if write("Escribe el objetivo a verificar: "), readln(Obj), ! ,analizar (Obj), ! .
analizar(Obj) if info (Obj,A),
checarbaseno (A), intentar(Obj,A), nl.
write("El objetivo ", Obj, " existe").
analizar(Obj) if not(info(Obj,_)),nl,
  write("Existe ", Obj, " (s/n): ? "), readln(Q),!, procesar(Obj,Q).
analizar(Obj) if !, nl, write ("El objetivo ",Obj, " no existe"),fail.
checarbaseno([]).
checarbaseno([Cab|Cola]) if not (no(Cab)), ! , checarbaseno(Cola).
intentar(O,[ ]) if !
intentar(O, [X|T]) if si(X), ! , intentar (O,T).
intentar(O,[X|T]) if analizar (X), ! , intentar (O,T), ! .
procesar(O,s)  if asserta(si(O)), !.
procesar(O,n) if asserta(no(O)), ! , clearwindow,fail.
escribirlista([ ],_).
escribirlista([H|T],3) if write (H),nl, escribirlista(T,1).
escribirlista([H|T],I) if N = I + 1, write(H, " "), escribirlista(T,N).
purgar if retract (si(X)), fail.
purgar if retract (no(X)), fail.
purgar if ! .
```

```
DOMAINS

  ROW,COL,LEN = INTEGER
  KEY = cr ; esc ; break ; tab ; btab ; del ; bdel ; ins ;
    end ; home ; ffast(INTEGER) ; up ; down ; left ; right ;
    ctrlleft; ctrlright; ctrlend; ctrlhome; pgup; pgdn;
    chr(CHAR) ; otherspec

PREDICATES

  readkey(KEY)
  readkey1(KEY,CHAR,INTEGER)
  readkey2(KEY,INTEGER)

CLAUSES

  readkey(KEY):-readchar(T),char_int(T,VAL),readkey1(KEY,T,VAL).
  readkey1(KEY,_,0):-!,readchar(T),char_int(T,VAL),readkey2(KEY,VAL).
  readkey1(cr,_,13):-!.
  readkey1(esc,_,27):-!.
  readkey1(chr(T),T,_).
  readkey2(up,72):-!.
  readkey2(down,80):-!.
  readkey2(ffast(N),VAL):-VAL>58,VAL<70,N=VAL-58,!.
  readkey2(otherspec,_).

PREDICATES

  menu(ROW,COL,STRING,LIST,INTEGER)
  menu1(ROW,LIST,ROW,INTEGER,INTEGER)
  menu2(ROW,LIST,ROW,INTEGER,INTEGER,KEY)

CLAUSES

  menu(LI,KOL,TXT,LIST,CHOICE):-
              shiftwindow(21), maxlen(LIST,0,ANTKOL),
              listlen(LIST,LEN),ANTLI=LEN,LEN>0,
              HH1=ANTLI+2,HH2=ANTKOL+2,
              makewindow(3,7,7,TXT,LI,KOL,HH1,HH2),
              HH3=ANTKOL,
              writelist(0,HH3,LIST),cursor(0,0),
              menu1(0,LIST,ANTLI,ANTKOL,CH).
```

```
                CHOICE=-1+CH,removewindow, shiftwindow(22), shiftwindow(2).
 menu1(LI,LIST,MAXLI,ANTKOL,CHOICE):- field_attr(LI,0,ANTKOL,112),
cursor(LI,0),readkey(KEY), menu2(LI,LIST,MAXLI,ANTKOL,CHOICE,KEY).

 menu2(_,_,_,_,-1,esc):-!.
 menu2(LI,_,_,_,CH,fast(10)):-!,CH=LI.
 menu2(LI,_,_,_,CH,er):-!,CH=LI.
 menu2(LI,LIST,MAXLI,ANTKOL,CHOICE,up):-
                LI>0,!,
                field_attr(LI,0,ANTKOL,7),
                LII=LI-1,
                menu1(LII,LIST,MAXLI,ANTKOL,CHOICE).

 menu2(LI,LIST,MAXLI,ANTKOL,CHOICE,down):-
                LI<MAXLI-1,!,
                field_attr(LI,0,ANTKOL,7),
                LII=LI+1,
                menu1(LII,LIST,MAXLI,ANTKOL,CHOICE).

 menu2(LI,LIST,MAXLI,ANTKOL,CHOICE,_):-
                menu1(LI,LIST,MAXLI,ANTKOL,CHOICE).
```

PREDICATES

```
 mainmenu
 proces(INTEGER)
 menu33
```

GOAL menu33.

CLAUSES

```
 menu33:-
  clearwindow,
 graphics(4,24,120),nl,nl,beep,beep,
        makewindow(22,112,0,"",24,0,1,80),
        makewindow(2,7,7,"CLASIFICACION DE LOS INSTRUMENTOS
MUSICALES",0,0,24,80),
        mainmenu.

 mainmenu:-  repeat,purgar,
                menu(8,49,"MENU DE OPCIONES",
                 [ "BASE DE CONOCIMIENTO",
                "",
```

```
                "DOS SHELL",
                "",
                "PROCESO"],CHOICE),
              proces(CHOICE),
              CHOICE=0,!.
removewindow,removewindow.
proces(0):-clearwindow,write("\n"SALIR DEL PROGRAMA? (s/n): "),readchar(T),T='s'.
proces(1):-file_str("musica.txt",TXT),display(TXT),!.
proces(1):-write("=> musica.txt no se encuentra en el directorio actual\n").
proces(2).
proces(3):-makewindow(3,7,0,"",0,0,25,80),write("escribe EXIT para regresar\n\n"),
      system("").!,removewindow.
proces(3):-write("=> command.com no accesible, presiona cualquier
Tecla"),readchar(_),removewindow.
proces(4).
proces(5):-makewindow(3,7,112,"",11,5,14,75),purgar,start,purgar.
```

BASE DE CONOCIMIENTO

```
info("leon",["felino","carnivoro"])
info("carnivoro",["come carne","tiene dientes"])
info("tigre",["felino","rayado"])
info("caballo",["relincha","equino"])
info("felino",["mamifero","da zarpazos"])
info("da zarpazos",["tiene garras","lastima"])
info("leon",["ruge","tiene melena"])
info("equino",["tiene 4 patas","trota"])
info("burro",["equino","rebuzna"])
info("mamifero",["vertebrado","sangre caliente"])
info("jirafa",["manchado","alto"]);
info("leopardo",["carnivoro","manchado"])
info("equino",["pezuñas","herbivoro"])
info("oso",["tiene pelo","da zarpazos"])
info("camello",["rumiante","tiene joroba"])
info("herbivoro",["come hierba","no es agresivo"])
info("gato",["domestico","maulla"])
info("gorila",["vertebrado","fuerte"])
info("vertebrado",["tiene huesos","es animal"])
info("vaca",["da leche","rumiante"])

PRESIONE ESC PARA CONTINUAR
```

CORRIDA DEL PROGRAMA

ESCRIBE EL OBJETIVO A VERIFICAR: tigre.

Existe tiene huesos (s / n) : ? s.
Existe es un animal (s / n) : ? s.

El objetivo vertebrado existe

Existe sangre caliente (s / n) : ? s.

El objetivo mamifero existe

Existe tiene garras (s / n) : ? s.
Existe lastima (s / n) : ? s.

El objetivo felino existe

Existe rayado (s / n) : ? s.

El objetivo tigre existe

En este ejemplo podemos observar como se cumple el objetivo tigre debido a que se cumple contodos los objetivos anteriores..

ESCRIBE EL OBJETIVO A VERIFICAR: tigre.

Existe tiene huesos (s / n) : ? s.
Existe es un animal (s / n) : ? s.

El objetivo vertebrado existe

Existe sangre caliente (s / n) : ? s.

El objetivo mamifero existe

Existe tiene garras (s / n) : ? n.

El objetivo felino no existe

El objetivo tigre no existe

En este ejemplo podemos observar como no se cumple el objetivo tigre debido a que no se cumple con todos los objetivos anteriores..

APENDICE "C"
EJEMPLO DE LA ESTRUCTURA DE UN REPORTE A REGISTRAR

"CONDE": UN SISTEMA EXPERTO PARA RECONOCIMIENTO DE PATRONES DE SINTOMAS Y DIAGNOSTICO

J. Arellano, E. Ramirez, Y Galicia, G. San Roman.

Instituto de Investigaciones Electricas
Departamento de Energía Nuclear
Apartado Postal 475.
Cuernavaca, Morelos, México
62000

RESUMEN

En este trabajo se presenta el sistema experto "CONDE", el cual ha sido construido para ayudar a los operadores a reconocer patrones de síntomas y diagnosticar fallas en el sistema de condensado de una central neuroeléctrica con reactor tipo BWR. Para la construcción de este sistema experto se utilizó tecnología del Estado del Arte en cuanto a la adquisición del conocimiento, estructuración de la base del conocimiento, y mecanismos de inferencia se refiere. La principal herramienta utilizada en la creación de "CONDE" es el esqueleto ("Shell") GENESIS, especial para este tipo de aplicación.

Palabras Claves: Sistemas Expertos (SE), esqueleto (Shell) para construccion de SE, adquisición del conocimiento, mecanismos de inferencias, patrones de síntomas, conjunto mínimo de corte (CMC).

1.0 INTRODUCCION

La aplicación de la tecnología de Sistemas Expertos para ayudar a los operadores en el análisis de síntomas (Alarmas, medidores, indicadores, etc.) y dignóstico de las causas de un disturbio, se ha visto con creciente interes en la industria del nucleo eléctrico. Uno de los

momentos más críticos en la operación de las centrales nucleoeléctricas es cuando el operador se enfrenta a disturbios que involucran muchas alarmas, medidores, e indicadores activandose en cortos periodos de tiempo, en estos casos el operador tiene que reconocer rapidamente cuales de estos síntomas son importantes y diagnosticar la causas raíz del disturbio. Obviamente, esta labor resulta aún más difícil para el operador en condiciones de fuerte tensión. Por esta razón, se ha planteado desarrollar sistemas expertos que ayuden a los operadores a reconocer los patrones de síntomas relevantes (separando causas de efectos) y a diagnosticar las causas básicas de un disturbio.

El Sistema Experto CONDE ha sido desarrollada para ayudar a los operadores a reconocer patrones de síntomas y diagnosticar fallas en el sistema de condensado de una central nucleoeléctrica con reactor tipo BWR. CONDE utiliza tecnología del Estado del Arte en cuanto adquisición del Conocimiento, Estructuración de la Base del Conocimiento y Mecanismos de Inferencia se refiere. La herramienta que se utilizó en la construcción de CONDE es el esqueleto (Shell) GENESIS, especial para este tipo de aplicación, y también desarrollado por los autores de este trabajo.

Al perderse la función del sistema de condensado de la central CONDE es capaz de reconocer rápidamente los patrones de síntomas relevantes y relacionarlos con los eventos responsables de la falla del sistema, logrando con ésto un diagnóstico efectivo de la causa raíz del problema.

CONDE interactúa con el operador a través de menús, preguntando por el estado de alarmas e indicadores que ha concluido son relevantes, y presentando los diagnósticos correspondientes. Adicionalmente, incluye mecanismos gráficos y textuales de explicación y justificación. Actualmente, CONDE cuenta con un conjunto de estrategias de reconocimiento de patrones de síntomas y diagnóstico que pueden ser utilizadas en el futuro para implementar un versión en la que el diálogo no sea con el operador, sino directamente con un sistema de adquisición de datos (CONDE versión tiempo real).

El desarrollo e implantación de CONDE se hizo en una computadora personal 80386 con monitor VGA y utilizando el esqueleto de desarrollo GENESIS. El uso de este esqueleto permitió que la construcción de CONDE se realizara de manera rápida y sistemática. De la manera en que fue estructurado, CONDE permite a los expertos humanos poder actualizar

fácilmente la base de conocimientos y las estratégias de reconocimiento de patrones de síntomas.

2.0 BREVE DESCRIPCION DEL SISTEMA DE CONDENSADO

En la central genérica que estamos considerando, el sistema de condensados se incluye dentro de "Sistema de Condensado y Agua de alimentación", y se han establecido bases de diseño referentes a la funcionalidad, seguridad, confiabilidad de este último. El sistema de Condensado comprende desde el pozo caliente del condensador hasta la descarga de los calentadores de baja presión. El sistema de agua de alimentación del reactor comprende desde la descarga de estos últimos calentadores de baja presión hasta la válvula check de aislamiento "Cheque" de aislamiento fuera del pozo seco.

Las funciones del sistema de condensado incluyen.

- Proporcionar un suministro continuo de agua al sistema de agua, de alimentación al reactor
- Proporcionar los medios para el precalentamiento inicial del agua de alimentación al reactor
- Mantener el agua de alimentación dentro de las condiciones requeridas de pureza
- Poseer un margen de funcionamiento para poder llevar agua a la succión de las turbo bombas de agua de alimentación durante transitorios

Los principales componentes del sistema de condensado se listan a continuación;

1. El condensador principal lado carcaza
2. Tres bombas de condensado cada una del 50 % de capacidad
3. Un condensador de vapor de sellos
4. Tres condensadores del (Off-gas) lado tubos, cada uno del 100 % de capacidad
5. Dos condensadores de los eyectores de aire cada uno del 100 % de capacidad
6. Siete desmineralizadores de condensado. Frecuentemente estos desmineralizadores se incluyen más bien como parte del "Sistemas de tratamiento del condensado".
7. Tres bombas de refuerzo de condensado cada una del 50 % de capacidad
8. Dos trenes de calentadores de baja presion cada tren del 50 % de capacidad incluye cinco calentadores

Las funciones de todos ellos integran secuencialmente desde la succión del condensado hasta la descarga hacia las bombas de agua de alimentación.

Para el desarrollo del SE el reconocimiento de patrones de alarmas del sistema de condensado se definieron como fronteras del sistema las siguentes:

- La entrada del flujo decondensado al sistema es en el cabezal de succión de las bombas de refuerzo de condensado
- Del pozo caliente del condensador se consideran unicamente algunos medidores de nivel
- La descarga del sistema de condensado se considera en el cabezal de succión de las turbo bombas de agua de alimentación.
- Por considerarse parte del "Sistema de tratamiento de condensado" no se incluyen para el desarrollo del SE los desmineralizadores.

3.0 DESARROLLO DEL SISTEMA EXPERTO

La metodología que se siguió para la construcción de CONDE difiere de las tradicionales de la tecnología de sistemas expertos, ya que incorpora novedosas ideas acerca de la utilización de técnicas de Análisis Probabilístico de Seguridad en el área de diagnóstico. La metodología que se usó para CONDE permitio desarrollar este sistema experto de manera rápida y sistemática.

A continuación se presentan las características más relevantes de CONDE, haciéndose énfasis en su arquitectura y la descripción de cada uno de los módulos contenidos en ésta.

3.1 Arquitectura del Sistema Experto.

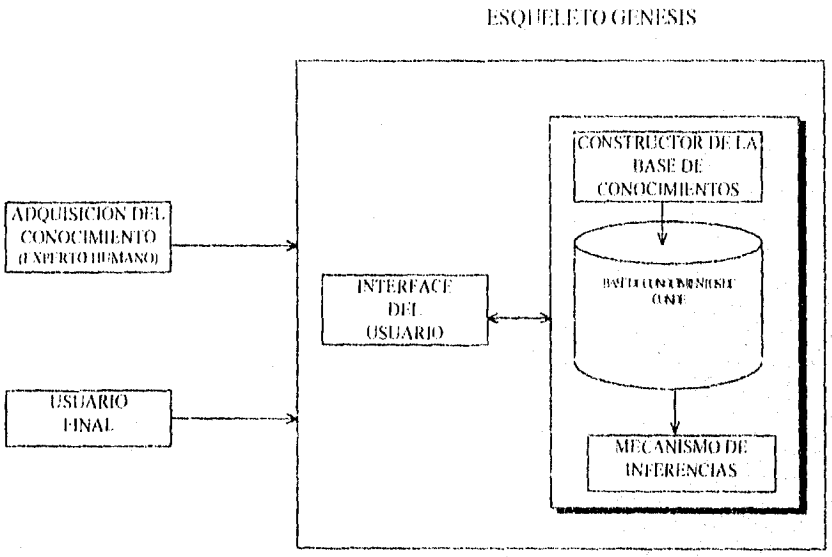
La figura 1 muestra la arquitectura del Sistema Experto CONDE . En está arquitectura se puede ver que el esqueleto ("shell") GENESIS es la principal herramienta de desarrollo de CONDE, GENESIS es un medio ambiente de desarrollo que proporciona una serie de programas y procedimientos especialmente diseñados para una construcción rápida y sistemática de sistemas expertos. La qrquitectura de CONDE, detallando los componentes de GENESIS, se muestraen la figura 2.

Para desarrollar CONDE, la adquisición del conocimiento se realizó, de acuerdo a lo que propone la metodología de GENESIS , mediante la construción, por parte de los expertos humanos, de modelos lógicos de falla - llamados árboles de fallas - del sistema condensado. Los expertos humanos también proporcionan, para cada uno de los eventos básicos de falla representados en el árbol, los síntomas que se presentan cuando esté evento ocurre, y las

posibilidades de ocurrencia de cada uno de éstos eventos. Con está información como entrada, GENESIS fué utilizado para generar automáticamente la base de conocimientos y las estrategias óptimas para el reconocimiento de patrones de síntomas. Además, el recorrido (ejecución) de las estrategias se realiza mediante el módulo de encadenamiento "hacia-adelante" del mismo GENESIS.

A continuación se detallan cada uno de los módulod en la arquitectura y el proceso de desarrollo de CONDE .

FIG. 1. ARQUITECTURA DEL SISTEMA EXPERTO CONDE.



3.2 Adquisición del conocimiento.

Para construir el sistema experto CONDE, la etapa de adquisición del conocimiento se realizó de acuerdo a los lineamientos que el esqueleto GENESIS recomienda. Estos lineamientos proponen que el proceso de adquisición del conocimeinto se divida en dos etapas.

- Construcción de árboles de fallas.
- Inclusión de síntomas.

Estas etapas son realizadas por los expertos humanos en diseño y operación del sistema.

Construcción de los Arboles de Fallas

Brevemente, un árbol de fallas es un modelo lógico-gráfico que describe las distintas formas en que eventos (normales y de falla) se combinan para ocasionar un evento indeseado - llamado "tope"- . Este evento tope puede representar una condición peligrosa, la pérdida de la función del sistema, etc. En el caso del sistema condensado, el evento tope que se seleccionó es "Bajo Flujo de Condesado en el Cabezal de Succión de Las Turbobombas de Agua de Alimentación". Este evento representa una condición en la que el Sistema de Condensado está enviando el 50 % o menos del flujo requerido al cabezal de descarga de los calentadores de baja presión . Una manera equivalente de describir este evento tope es "Flujo insuficiente en el cabezal de descarga del Sistema de Condesado".

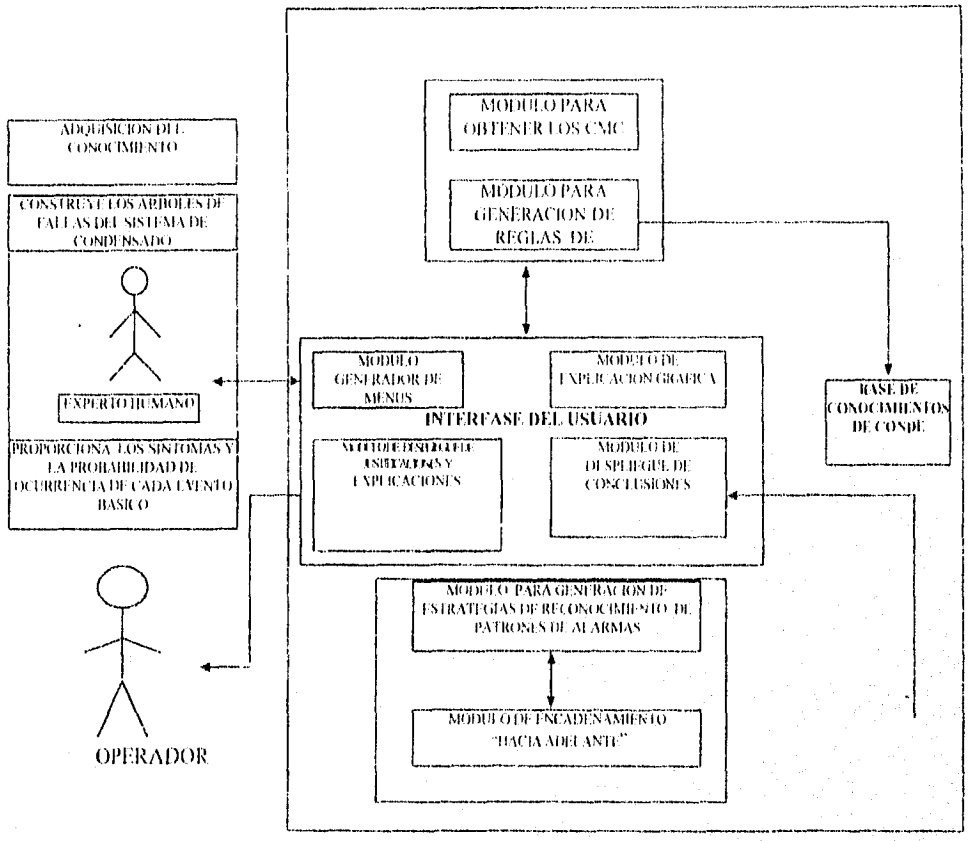
Es importante recalcar que los árboles de fallas son un instrumento que permite realizar una adquisición sistemática del conocimiento, ya que en los expertos humanos pueden plasmar, no únicamente su conocimiento formal acerca del diseño y operación del sistema, sino también su "experiencia" no formal acumulada. Además, por su naturaleza gráfica, el conocimiento representado en los árboles puede ser fácilmente revisado, incrementado y auditado por otros expertos humanos.

Para construir el sistema experto CONDE, los expertos humanos que desarrollaron el árbol fueron los autores de este documento, apoyándose en literatura tal como las descripciones del sistema , cursos de tecnología, etc. Se espera que versiones futuras de CONDE incorporen el conocimiento de operadores expertos.

Inclusión de Síntomas

Para cada evento básico en el árbol de fallas se incorporaron los síntomas que se presentan cuando cada uno de estos eventos ocurre. Una vez más, la fuente de conocimiento de estos síntomas fueron los investigadores autores de este reporte.

FIG 2. Arquitectura detallada del Sistema Experto CONDE.



3.3 Estrcuctura de la base de conocimientos.

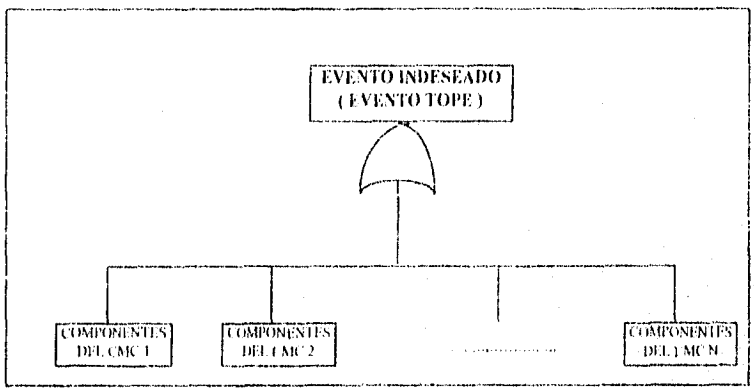
GENESIS incluye un módulo para codoficar y estructurar automáticamente el conocimiento de los expertos humanos de manera que pueda ser fácilmente utilizado por un sistema experto. Este módulo se denomina constructor de la base de conocimiento . Las entradas

requeridas para que el esqueleto realice esta tarea son una descripción alfanumerica del árbol y los síntomas asociados a cada evento básico.

Como primer paso, GENESIS redujo el árbol del Sistema de Condesado a un árbol de dos niveles en términos de sus Conjuntos Mínimos de Corte . Brevemente, un CMC es un conjunto mínimo de eventos básicos del árbol tal que, si todos los eventos en el conjunto se presentan, el evento tope ocurre.

Todo árbol de fallas reducido a dos niveles en términos de sus CMC tiene una estructura como la que se muestra en la figura 3. En ella se puede observar que el evento tope ocurre cuando cualquiera de los CMC bajo la compuerta "O" ocurre. Esta estructura de dos niveles simplifica enormemente el espacio de búsqueda. El orden de un CMC es el número de eventos básicos que lo componen.

FIG. 3. Árbol en dos niveles.



GENESIS obtiene los conjuntos mínimos de corte de cualquier árbol de fallas llamando a ejecución al código FTAP el cual utiliza, la ellección del usuario, métodos de sustitución "arriba-abajo" y/o "abajo-arriba".

Para el sistema de condensado se generarón un total de 1579 conjuntos mínimos de corte de orden uno, dos y tres.

Una vez que obtiene el arbol en dos niveles, GENESIS, genera, para cada CMC, los grupos de patrones de síntomas que pueden presentarse cuando cada CMC ocurre (reglas de producción).Para ésto , se incorporan a cada evento básico de cada CMC los síntomas correspondientes y se realizan las reducciones booleanas que se requieran. Por ejemplo, supongamos que se desean obtener los patrones de síntomas del CMC de orden dos compuesto por los eventos básicos:

EVENTO	DESCRIPCION_EVENTO
B100	Motoválvula MV-0908 Falla a permanecer Abierta.
D61	No Hay Potencia del Bus Para la Bomba P001A Durante Operación.

cuyos síntomas individuales son:

EVENTO	SINTOMAS
B100	PI-0912_BAJ
	LG-1015_NOR
	LG-1016_NOR
	A3-(1-8)_NOR
	MV-0908_CERR
	PI-0906_ALT
	PI-0907_ALT
D61	PI-0912_BAJ
	LG-1015_NOR

LG-1016_NOR

PI-0903_BAJ

PI-0902_ALT

A3(1-6)_ON

Para generarlas reglas (patrones) GENESIS asume que los dos eventos básicos han ocurrido, activa los síntomas de cada uno de los eventos, y realiza minimizaciones Booleanas. En este ejemplo, la única ley del algebra Booleana que tiene que aplicarse es A.A = A, con lo que el patrón de síntomas resultante para el CMC queda como sigue:

PI-0912_BAJ
LG-1015_NOR
LG-1016_NOR
A3 (1-8)_NOR
MV-0908_CERR
PI-0906_ALT
PI-0907_ALT
PI-0903_BAJ
PI-0902_ALT
A3 (1-6)_ON

Así, se habrá generado una regla de producción con la siguiente estructura:

IF

PI-0912_BAJ
LG-1015_NOR
LG-1016_NOR
A3 (1-8)_NOR
MV-0908_CERR
PI-0906_ALT
PI-0907_ALT
PI-0903_BAJ
PI-0902_ALT
A3 (1-6)_ON

THEN

NO HAY POTENCIA DEL BUS PARA LA BOMBA P00LA DURANTE OPERACION

(Y)

MOTOVALVULA MV-0908 FALLA A PERMANECER ABIERTA.

GENESIS puede generar más de una regla para CMC, cuando uno o varios de los eventos incluidos en éste tiene más de un patron de síntomas posible. Para el sistema experto CONDE, GENESIS generó automáticamente un total de 1579 reglas de producción.

3.4 Mecanismos de inferencia.

Las 1579 reglas previamente generadas constituyen la base de conocimientos de CONDE a partir de ellas el operador podría, siguiendolas una a una, reconocer el patron de alarmas que se está presentando cuando falla el sistema de condensado. Sin embargo, este es un procedimiento ineficiente, por lo que GENESIS incorporá un nódulo para generar estrategias óptimas de reconocimiento de patrones que garantiza que únicamente unas cuantas de ellas tendrán que ser checadas antes de realizar algún diagnóstico efectivo de las causas de falla del sistema.

Suponiendo que el evento indeseado se está presentando la idea general del algoritmo implementado en el módulo de construcción de estrategias es que, si al revisar un síntoma se detecta que éste no ha ocurrido, puede entonces deducirse que ningún patron en los que aparece este síntoma ha ocurrido, y como consecuencia pueden estas reglas eliminarse del espacio de búsqueda. Las estrategias que se generan consisten de una lista inicial y un conjunto de sublistas. La lista inicial incluye a un grupo de síntomas ordenados por orden descendente de sus valores de importancia probabilística (*), además de la ruta de búsqueda que se recomienda seguir en caso de que el síntoma se haya presentado . Si el síntoma es el único elemento de una regla, entonces por sí mismo puede indicar qué eventos básicos han ocurrido. Si el síntoma está ligado a otros en una o varias reglas, la estrategia nos remitirá a una sublista en la que aparecen algunos otros síntomas comunes que aparecen en reglas junto con el síntoma referido y, a su vez, estos dirigirán la búsqueda si es preciso. Así, una estrategia para reconocimiento de patrones de síntomas construida de esta forma optimiza la búsqueda, puesto que solo tendrá que considerar los patrones que contengan alarmas que hayan ocurrido.

Otro punto que es muy importante recalcar es la estrategia generada realmente representa la manera en que un operador trataría de realizar el diagnóstico al analizar los síntomas. La búsqueda de las causas de falla por subsistema es lo que muy probablemente haría un operador humano experto, cosa que concuerda muy bien con la labor automática "inteligente" realizada por el sistema experto al construir la estrategia de reconocimiento de síntomas; de hecho, lo más probable es que el operador, tal vez con base en el entrenamiento, cheque los mismos síntomas que la estrategia propone. La estrategia generada puede ser fácilmente recorrida "hacia adelante" por una rutina de computadora para poder realizar diagnóstico basado en síntomas de manera muy eficiente. En CONDE, esta labor la lleva acabo el módulo de encadenamiento "hacia adelante" incluido en GENESIS, el cual es encargado de ir preguntando al operador o a un sistema de adquisición de datos, el estado de las alarmas e indicadores que van apareciendo en el recorrido de la estrategia, y a partir de las respuestas realizar las siguientes preguntas tendientes a diagnosticar las causas de falla del sistema. La presente versión de CONDE, interactúa directamente con el operador, quien es el que va respondiendo las preguntas realizadas por el sistema experto.

3.5 Interface del usuario.

CONDE utiliza la misma interfase del usuario de GENESIS para comunicarse con el operador y/o experto humano. Las funciones principales de esta interface son generar menús, desplegar diagnósticos, y explicaciones gráficas y textuales.

CONDE incorpora los mímicos del sistema de condensado en dos niveles: de sistema y de subsistema. Una vez que CONDE ha realizado un diagnóstico de las probables causas de falla del sistema presenta al operador el mímico de todo el sistema de condensado, señalando el sistema en el que se identificóel problema y los síntomas clave que lo llevaron a esta conclusión; despues de esto, presenta en detalle la subsección en cuestión y los síntomas que justifican el diagnóstico.

4.0 USO Y VALIDACION DEL SISTEMA EXPERTO.

Esta primera versión del sistema experto CONDE ha sido implementada en una computadora personal 80386 con monitor VGA y diseñada para interaccionar con los usuarios (expertos humanos y operadores) a través de menús. Estos últimos van dirigiendo las diferentes operaciones que los usuarios pueden llevar acabo sobre el sistema experto.Se contempla que los operadores pueden únicamente hacer consultas, actualizar la base de conocimientos y estrategias de reconocimiento.

Durante el proceso de desarrolllo se hicieron validaciones extensivas del árbol de fallas, conjuntos mínimos de corte, reglas de producción, y estrategias de reconocimiento.

El proceso de validación de un sistema experto es en general muy largo, y requiere de pruebas extensivas por parte de los usuarios finales y expertos humanos; tal es el caso de la presente versión de CONDE, la cual, aunque ha sido simplemente probada por los expertos humanos por los que fue desarrollada, en su siguiente etapa requiere una revisión detallada del conocimiento.

5.0 CONCLUSIONES.

Se ha desarrollado un sistema experto -CONDE- en reconocimiento de patrones de síntomas y diagnóstico de fallas para el sistema de condensado de una Central Nucleoeléctrica tipo BWR. CONDE ayuda a los operadores en la tarea de analizar alarmas, medidores para diagnosticar las causas de la falla del sistema.

El esquema de clasificación y síntesis de síntomas que utiliza CONDE para reconocer patrones es similar al que desarrollaría un operador humano experimentado cuando trata de reconocer las alarmas, medidores e indicadores importantes para diagnosticar las causas de falla del sistema . Esta característica se logró para CONDE gracias a los esquemas que utiliza el esqueleto GENESIS para generar automáticamente estrategias "inteligentes" de reconocimiento de patrones .

La presente versión de CONDE interactúa con los operadores a través de menús; sin embargo, llevarla a una versión que dialogue con los sistemas de aquisición de datos es una tarea que simplifica enormemente apoyándose en el conocimiento ya adquirido, la actual

estructura de la base de conocimientos, y las estrategias de reconocimientos, y las estrategias de reconocimeinto ya generadas.

Las pruebas que se le han prácticado a CONDE demuestran que es una poderosa herramienta para ayudar a los operadores en el proceso de síntesis de alarmas , medidores e indicadores del sistema de condensado y que su uso contribuye enormemente con los operadores, ya que éstos logran un diagnóstico más rápido y exacto de las causas del disturbio.

GENESIS, el medio ambiente de desarrollo que se utilizá para la construcción de CONDE, demostró ser una herramienta muy avanzada y útil que permitió que las tareas involucradas se realizaran de manera rápida y sistemática.

6.0 REFERENCIA.

(*) USNRC (1984), "Probabilistic Risk Assessment (PRA) Reference Document", NUREG-1050.

APENDICE "D"
REFERENCIAS DE LAS PERSONAS CONSULTADAS EN LA INVESTIGACION

CHRISTIAN LEMAITRE:

Es Doctor en Infomática en la Universidad de Paris VI, Francia. Actualmente es investigador Titular de LANIA. Sus áreas de interés son Informática e Inteligencia Artificial. Pertenece a la Sociedad Mexicana de Inteligencia Artificial.

AGUSTIN PEREZ:

Es Ingeniero en Sistemas Computacionales. Egresado del Instituto Tecnologico y de Estudios Superiores de Monterrey. Actualmente es Director del área de Inteligencia Artificial en SofTek.

CARLOS ZOZAYA GOROSTIZA:

Es Doctor en Ingeniería Civil de la Universidad de Carnegi Mellon, EUA, Trabajando en el área de Sistemas Expertos. Actualmente es el Director General de la División Académica de Computación del ITAM, sus áreas de interés son Inteligencia Articial, Optimización e Impacto de la Tecnologia de Información en las Organizaciones. Pertenece a la Sociedad Mexicana de Inteligencia Artificial.

APENDICE "D"
REFERENCIAS DE LAS PERSONAS CONSULTADAS EN LA INVESTIGACION

OSWALDO CAIRO BATISTUTTI:

Es candidato a Doctor en Ciencias en Ingenieria Eléctrica, CINVESTAV-IPN. Actualmente es profesor de tiempo completo del departamento academico de computacion y coordinador del diplomado de sistemas expertos en el ITAM. Sus áreas de interes son: Inteligencia Artificial Sistemas, Expertos e Ingenieria del Conocimiento y Groupware. Pertenece a la Sociedad Mexicana de Inteligencia Artificial.

HORACIO CARVAJAL:

Es Ingeniero en Sistemas. Actualmente es Jefe de Sistemas Computacionales del Banco Nacional de México (BANAMEX).

Pertenece a la Sociedad Mexicana de Inteligencia Artificial ocupando el cargo de Tesorero.

SILVIA GUARDATI BUENO:

Es candidato a Doctor en Ciencias en Ingenieria Eléctrica, CINVESTAV-IPN. Actualmente es Directora de la Ingeniería en Computación del ITAM. Sus áreas de interes son: Inteligencia Artificial, Sistemas Expertos y Mecanismos de Razonamiento Aplicados a los Sistemas Expertos. Pertenece a la Sociedad Mexicana de Inteligencia Artificial.

APENDICE "E"
REFERENCIAS BIBLIOGRAFICAS

- [Carrico 89] Carrico. M.A., Girard, J.E.,& Jones.J.P.,Building Knowledge Systems. Deloping and Managing rule-basedapplications. New York. McGraw-Hill,1988.

- [Crosby 90] Crosby,Phillip., Manual para el Sistema de Eduación en Calidad, Phillip Crosby & Associated ,1980.

- [DEC 89] Digital Equipment Corporation., A Technical Summary of Digital's Workstations. DEC, 1989.

- [Feigenbaum 88] Feigenbaum. E.A., y McCorduck. P., y The Rise of the Expert Company, New York Times Books. 1988.

- [Forgy 80] Forgy. C. L. & McDermott. J. Production System Conflict Resolution Strategies. Eds. Waterman & Hayes-Roth. 1978.

- [From 55] Fromm. Eric., Psicoanálisis de la Sociedad Contemporanea Fondo de Cultura Económica. México, 1956.

- [Fujitsu 87] Fujitsu Microelectronics, Inc., IMB86900 RISC Processor Architecture Manual., 1987.

- [Graham 88] Graham. Paul.,Anatomy of a LISP Machine. AI Expert Magazine, vol 3. No.12, Diciembre 1988.

- [Harmon 89] Harmon, Paul , Expert Systems on the Macintosh ,Expert Systems Strategies, vol 5, NO.3, Cutter Information Corp.,1989.

- [Hart 86] Hart Anna, Knowledge Acquisition for Expert Systems, McGraw-Hill. New York, 1986.

- [Hwang 88] Hwang K., & Briggs F.A., Arquitectura de Computadoras y Procesamiento Paralelo. McGraw-Hill, España,1988.

- [IBM 90a] International Bussines Machine, Ficha Tecnica del modelo 7013 serie RS/6000, México 1990.

- [IBM 90b] International Business Machine, IA closer look at IBMPS/2 MicroChanel Architecture, IBM Publications, 1990.

- [IBERAMIA 90] 2o. Congreso Iberoamericano de Inteligencia Artificial, Memorias, Limusa, Morelia, Michoacán, México 1990.

- [IBERAMIA 92] 3er. Congreso Iberoamericano de Inteligencia Artificial, Memorias, Limusa, La Habana, Cuba 1992.

- [IEEE 88a] IEEE MICRO, vol 8 No.2, The Computer Society, Abril 1988.

- [IEEE 88a] IEEE MICRO, vol 8 No.3, The Computer Society, junio1988.

- [IDT 88] Integrated Devices Technology, Inc., Reduce Instruction Set Computer (RISC) Processors, High Performance CMOS Databook.

- [ITESM 88a] ITESM, Catálogo, Programas de Graduados, ITESM, Mty, México, 1988.

- [ITESM 88b] ITESM, Memorias, 1er Simposium Internacional de Inteligencia Artificial. ITESM, MTY. México, 1988.

- [ITESM 88c] ITESM, Tutorial, 1er Simposium Internacional de Inteligencia Artificial., ITESM, Mty. México, 1988.

- [ITESM 90d] CETEC, Programa de Tecnología Avanzada para la Producción., ITESM, Mty. México, 1990.

- [ITESM 89e] Centro de Investigación en Informática, Sistemas Basados en el Conocimiento. ITESM, Mty. México, 1989.

- [ITESM 90d] CETEC, Centro de Inteligencia Artificial., ITESM,Mty. México, 1990.

- [Jakson 86] Jackson Peter, Introduction to Expert Systems, Addison-Wesley Publishing Company, Workingham, 1986.

- [Kane 88] Kane, Gerry., MIPS R2000 RISCK Architecture, 1987.

- [Lehrer 74] Lehrer K., Knowledge, Oxford University Press, 1974.

- [Levine 87] Levine R.I., Drang D.E., Edelson B., IA Comprehensive Guide to AI & Expert Systems, McGraw-Hill, New York, 1987.

- [McDermott 82] McDermott J., R1: A Rule-Based Configurer of Computer Systems, Artificial Intelligence, 19:39-88, 1982.

- [Milner 88] Milner W.L.,Common LISP, a Tutorial., Prentice-Hall, New Yersey, 1988.

- [Mishkoff, 86] Mishkoff, Henrry C., Understanding Artificial Intelligence, Radio Shack, Texas Instruments Information Publishing Center, 1986.

- [Mishkoff, 88] Mishkoff, Henrry C., A fondo: Inteligencia Artificial, Ediciones Anaya Multimedia, Madrid, 1988.

- [Murray 88] Murray J.T., Murray M.J., Expert Systems in Data Processing, McGraw-Hill, New York, 1988.

- [Negrete 88] Negrete M.J., Inteligencia Experimental en Computadoras., LIMUSA, México D.F., 1988.

- [Nilson 71] Nilson, N.J., Problem-Solving Methods in Artificial Intelligence, McGraw-Hill, USA.

- [Osborn 53] Osborn A.F., Applied Imagination Scribner.,New York, 1953.

- [Ritch 88] Ritch.E., Inteligencia Artificial, Gustavo Gili, Barcelonana, 1988.

- [Roberts 87] Roberts.R., The Power of Turbo Prolog. The Natural Language of Artificial Intelligence, Tab Books, U.S.A.,1987.

- [Rolston 88] Rolston David W. Principles of Artificial Intelligence & Expert Systems Development, McGraw-Hill, New York,1988.

- [Scheel 88] Scheel C., Ingeniería de Sistemas Basados en el Conocimiento, ITESM, Monterrey Mexico, 1988.

- [SMIA 88a] SMIA, Memorias, V Reunión Nacional de Inteligencia Artificial., SMIA-LIMUSA, México D.F., 1988.

- [SMIA 89a] SMIA, Memorias, VI Reunión Nacional de Inteligencia Artificial., SMIA-LIMUSA, México D.F., 1989.

- [SMIA 91a] SMIA, Memorias, VIII Reunión Nacional de Inteligencia Artificial., SMIA-LIMUSA, México D.F., 1991.

- [SMIA 92a] SMIA, Memorias, IX Reunión Nacional de Inteligencia Artificial., SMIA-LIMUSA, México D.F., 1992.

- [SYMPOSIUM 91] Symposium on natural and artificial intelligence: a meeting between neuroscience and artificial intelligence, LANIA, Xalapa, Veracruz, México 1991.

- [Schutzer 87] Schutzer, Daniel., Artificial Intelligence., Van Nostrad Reinhold Company Inc., Ontario Canada, 1987.

- [Stock 88] Stock, Michael., AI Theory & Applications in the VAXEnvironment., McGraw-Hill Book Company, New York, 1988.

- [Taoffler 72] Toffler, Alvin., El Shock del Futuro, Fondo de Cultura Económica, México D.F., 1972.

- [Winston 87] Winston P.H., Berthold K.P.H., LISP, Addison-Wesley, Massachusetts, 1987.

- [Parsey 88] Parsey k., Mark C., Expert Systems for Experts,Wiley & Sons, New York, 1988.

- [Von Bertalanffy 86] Von Bertalanffy L., Teoría General de Los Sistemas. Fondo de Cultura Económica, México D.F., 1986.

- [Whitehead 10] Whitehead A.N., & Russell B., Principia Mathematica, Cambridge: University Press, 1910.

- [William 78] Expert William, G. Gervarter., The Nature & Evaluation of Comercial Systems Building Tools., Computer Magazine, Mayo1987.

- [Zozaya 89] Press, Zozaya-Gorostiza C., Hendrickson C., Rehak D.R., Knowledge-Base Process Planning for Construction & Manufacturing, Academic Boston, 1989.