

## 2.4. Desarrollo histórico de los sistemas expertos

Los Sistemas Expertos como tales, surgen a mediados de los años sesenta; en esos tiempos, se creía que bastaban unas pocas leyes de razonamiento junto con potentes ordenadores para producir resultados brillantes<sup>16</sup>. Los primeros investigadores que desarrollaron programas basados en leyes de razonamiento fueron Alan Newell y Herbert Simon, quienes desarrollaron el GPS (General Problem Solver). Este sistema era capaz de resolver problemas como el de las torres de Hanoi y otros similares, a través de la criptoaritmética. Sin embargo, este programa no podía resolver problemas más “cotidianos” y reales, como, por ejemplo, dar un diagnóstico médico.

Entonces algunos investigadores cambiaron el enfoque del problema: ahora se dedicaban a resolver problemas sobre un área específica intentando simular el razonamiento humano. En vez de dedicarse a computarizar la inteligencia general, se centraron en dominios de conocimiento muy concretos<sup>17</sup>. De esta manera nacieron los Sistemas Expertos.

El primer SE que se aplicó a problemas más reales fue desarrollado en 1965 con el fin de identificar estructuras químicas: el programa se llamó DENDRAL. Lo que este Sistema Experto hacía, al igual que lo hacían los expertos de entonces, era tomar unas hipótesis relevantes como soluciones posibles, y someterlas a prueba comparándolas con los datos<sup>18</sup>. El nombre DENDRAL significa árbol en griego. Debido a esto, el programa fue bautizado así porque su principal tarea era buscar en un árbol de posibilidades la estructura del compuesto.

El siguiente SE que causó gran impacto fue el Mycin, en 1972, pues su aplicación era detectar trastornos en la sangre y recetar los medicamentos requeridos. Fue tal el éxito

---

<sup>16</sup> SAMPER Márquez, Juan José. **Sistemas Expertos: El conocimiento al poder.**  
URL(<http://www.psycologia.com/articulos/ar-jsamper01.htm>).

<sup>17</sup> *Ibid.*

<sup>18</sup> ROOS, Rita. **Historia de los Sistemas Expertos.**  
URL(<http://www.monografias.com/trabajos10/exper/exper.shtml>).

---

de Mycin que incluso se llegó a utilizar en algunos hospitales<sup>19</sup>. Para 1973, se creó Tieresias, cuya función era la de servir de intérprete o interfaz entre los especialistas que manejaban el Mycin cuando introducían nuevos conocimientos<sup>20</sup>. Tieresias entraba en acción cuando Mycin cometía un error en un diagnóstico, por la falta de información o por alguna falla en el árbol de desarrollo de teorías, corrigiendo la regla que generaba el resultado o bien destruyéndola.

Para 1980 se implantó en la Digital Equipment Corporation (DEC) el primer Sistema Experto: el XCON. Para esto se tuvieron que dedicar dos años al desarrollo de este SE. Y valió la pena para la compañía, pues aún cuando en su primer intento al implantarse en 1979 consiguió sólo el 20% del 95% de la resolución de las configuraciones de todas las computadoras que salieron de DEC, volvió al laboratorio de desarrollo otro año más, y a su regreso le resultó en un ahorro de 40 millones de dólares a dicha compañía.

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

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

Desgraciadamente para DEC, el SE que antes fuera un gran ahorro dejó de ser rentable en 1987, debido a que al actualizarlo llegaron a gastar más de dos millones de

---

<sup>19</sup> HURTADO Vega, José de Jesús. *Op. cit.*

<sup>20</sup> ROOS, Rita. *Op. cit.*

dólares al año para darle mantenimiento, por lo que su beneficio quedó en duda, y por lo tanto fue descontinuado; algo similar ocurrió con DELTA21.

A continuación la Tabla N°.1 presenta de manera resumida los primeros Sistemas Expertos y sus aplicaciones.<sup>22</sup>

**Tabla N° 2.1 Primeros Sistemas Expertos y Sus Aplicaciones**

Sistema	Fecha	Autor	Aplicación
DENDRAL	1965	Stanford	Deduce información sobre estructuras químicas
Máxima	1965	MIT	Análisis matemático complejo
HearSay	1965	Carnegie Mellon	Interpreta en lenguaje natural un subconjunto del idioma
Mycin	1972	Stanford	Diagnóstico de enfermedades de la sangre
Tieresias	1972	Stanford	Herramienta para la transformación de conocimientos
Prospector	1972	Stanford	Exploración mineral y herramientas de identificación
Age	1973	Stanford	Herramienta para generar Sistemas Expertos
OPSS	1974	Carnegie Mellon	Herramientas para desarrollo de Sistemas Expertos
Caduceos	1975	University of Pittsburg	Herramienta de diagnóstico para medicina interna
Rosie	1978	Rand	Herramienta de desarrollo de Sistemas Expertos

<sup>21</sup> ROOS, Rita. *Op. cit.*

<sup>22</sup> ROLSTON, David. *Op. cit.*

R1	1978	Carnegie Mellon	–	Configurador de equipos de computación para DEC
----	------	-----------------	---	---

## 2.5 Generalidades de los sistemas expertos

### 2.5.1 ¿Qué son los sistemas expertos?

Desde su aparición, a mediados de 1960, los Sistemas Expertos se han definido como aquellos programas que se basan en el conocimiento y tratan de imitar el razonamiento de un experto para resolver un problema de un tópico definido. Su comportamiento se basa generalmente en reglas, es decir, se basa en conocimientos previamente definidos, y mediante estos conocimientos, los SE son capaces de tomar decisiones. Sería ilógico pensar que solo existe una definición de Sistemas Expertos, ya que tanto los SE como la propia IA han ido evolucionando a la par a través de los años. <sup>23</sup>

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

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

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

Un SE es un sistema informático que simula los procesos de aprendizaje, memorización, razonamiento, comunicación y acción de un experto humano en una determinada rama de la ciencia, suministrando, de esta forma, un consultor que puede sustituirle con unas ciertas garantías de éxito.<sup>24</sup>

---

<sup>23</sup> DE ÁVILA Ramos, Jorge. *Op. cit.*

<sup>24</sup> *Ibid.*

Podemos agregar otro concepto actual, dado por la Asociación Argentina de Inteligencia Artificial:

Los Sistemas Expertos permiten el desarrollo de otros sistemas que representan el conocimiento como una serie de reglas. Las distintas relaciones, conexiones y afinidades sobre un tema pueden ser compiladas en un Sistema Experto pudiendo incluir relaciones altamente complejas y con múltiples interacciones.<sup>25</sup>

Las características mencionadas en las definiciones anteriores le permiten a un Sistema Experto almacenar datos y conocimiento, sacar conclusiones lógicas, ser capaces de tomar decisiones, aprender, comunicarse con expertos humanos o con otros Sistemas Expertos, explicar el razonamiento de su decisión y realizar acciones como consecuencia de todo lo anterior.<sup>26</sup>

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

Una solución del problema tiene una rentabilidad tan alta que justifica el desarrollo de un sistema, pues las soluciones son necesidades del área y no se ha trabajado en otros métodos para obtenerla.

El problema puede resolverse sólo por un conocimiento experto que puede dar forma a los conocimientos necesarios para resolver el problema, y la intervención de experto dará al sistema la experiencia que necesita.

El problema puede resolverse solamente por un conocimiento experto en vez de usar algoritmos particulares.

---

<sup>25</sup> ASOCIACIÓN ARGENTINA DE LA INTELIGENCIA ARTIFICIAL. **Glosario.**  
URL([http://www.lafacu.com/apuntes/ingenieria/asocacion\\_argentina\\_de\\_inteligencia\\_artificial/asocacion\\_argentina\\_de\\_inteligencia\\_artificial.html](http://www.lafacu.com/apuntes/ingenieria/asocacion_argentina_de_inteligencia_artificial/asocacion_argentina_de_inteligencia_artificial.html))

<sup>26</sup> DE ÁVILA Ramos, Jorge. *Op. cit.*

---

Se tiene acceso a un experto que puede dar forma a los conocimientos necesarios para resolver el problema. La intervención de este experto dará al sistema la experiencia que necesita.

El problema puede o no tener una solución única. Los Sistemas Expertos funcionan mejor con problemas que tienen un cierto número de soluciones aceptables.

El problema cambia rápidamente, o bien el conocimiento es el que cambia rápidamente, o sus soluciones son las que cambian constantemente.

El desarrollo de un Sistema Experto no se considera que está acabado una vez que funciona este, sino que continúan desarrollando y actualizando tanto el conocimiento del sistema como los métodos de procesamiento, quedando reflejados los progresos o modificaciones en el campo, área o sistema.<sup>27</sup>

### **2.5.2 Los expertos Humanos**

Un experto humano es una persona que es competente en un área determinada del conocimiento o del saber. Un experto humano es alguien que sabe mucho sobre un tema determinado y que puede dar un consejo adecuado. Esta experiencia sólo se adquiere tras un largo aprendizaje y a base de mucha experiencia.<sup>28</sup> Los expertos humanos tienen las siguientes características generales:

- Son personas raras, tanto por su escaso número como su comportamiento poco “ortodoxo” y e “incomprensible” frente a los problemas con los que se enfrentan.
- Son caros por dos motivos: por su escaso número y por necesitar un largo periodo de aprendizaje.

---

<sup>27</sup> GONZÁLEZ Ayala, Luis Enrique. **Sistemas Expertos.**  
URL (<http://www.prodigyweb.net.mx/enrayala/Sistemas.htm>)

<sup>28</sup> BRIAN Keith y Steven Brain, **Inteligencia artificial en el dragón.**

- No están siempre disponibles, pues son humanos y cuando se jubilan o mueren se llevan con ellos todos sus conocimientos. Es por eso que tradicionalmente están acompañados de un “aprendiz”.
- Hay expertos que tienen mal carácter, son informales o poco comunicativos, lo que a veces les hace antipáticos.
- La forma más rápida de formar a un experto es mediante el aprendizaje formal o académico (“conocimiento profundo”) en un principio, y posteriormente un aprendizaje informal o práctico (“conocimiento informal”).<sup>29</sup>

### 2.5.3 Diferencias entre un experto y un no experto humano

**Tabla N° 2.2 Diferencias entre un experto y un no experto humano.**

	Experto	No experto
Tiempo de Resolución	Pequeño	Grande
Eficacia Resolutiva	Alta	Baja
Organización	Alta	Baja
Estrategias y Tácticas	Sí	No
Búsqueda de Soluciones	Heurística	No Heurística
Cálculos Aproximados	Sí	No

El uso de heurísticas contribuye grandemente a la potencia y flexibilidad de los SE y tiende a distinguirlos aún mas del software tradicional.<sup>30</sup>

### 2.5.4 Diferencias entre un SE y un programa tradicional

---

<sup>29</sup> SÁNCHEZ y Beltrán, Juan Pablo. **Sistemas expertos: Una metodología de programación.**

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

	Sistema Experto	Programa Tradicional
Conocimiento	En programa e independiente	En programa y circuitos
Tipo de datos	Simbólicos	Numéricos
Resolución	Heurística	Combinatoria
Def. problema	Declarativa	Procedimental
Control	Independiente. No secuencial	Dependiente. Secuencial
Conocimientos	Imprecisos	Precisos
Modificaciones	Frecuentes	Raras
Explicaciones	Sí	No
Solución	Satisfactoria	Optima
Justificación	Sí	No
Resolución	Área limitada	Específico
Comunicación	Independiente	En programa

### 2.5.5 Diferencias entre un sistema experto y un experto humano

**Tabla N° 2.4 Diferencias entre un Sistema Experto y un Experto Humano.**

	Sistema experto	Experto humano
Conocimiento	Adquirido	Adquirido + Innato
Adquisición del conocimiento	Teórico	Teórico + Práctico
Campo	Único	Múltiples
Explicación	Siempre	A veces
Limitación de capacidad	Sí	Sí, no valuable
Reproducible	Sí, idéntico	No
Vida	Infinita	Finita

<sup>30</sup> ROLSTON, David W. *Op. cit.*

## 2.5.6 Ventajas de los sistemas expertos

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

- Están siempre disponibles a cualquier hora del día y de la noche, y de forma interrumpida.
- Mantiene el humor.
- Pueden duplicarse (lo que permite tener tantos SE como se necesiten).
- Pueden situarse en el mismo lugar donde sean necesarios.
- Permiten tener decisiones homogéneas efectuadas según las directrices que se les fijen.
- Son fáciles de reprogramar.
- Pueden perdurar y crecer en el tiempo de forma indefinida.
- Pueden ser consultados por personas o otros sistemas informáticos (SAP).<sup>31</sup>

## 2.5.7 Forma en que los usuarios interactúan con los SE.

El usuario de un SE puede estar operando en cualquiera de los siguientes modos:

- Verificador. El usuario intenta comprobar la validez del desempeño del sistema.
- Tutor. El usuario da información adicional al sistema o modifica el conocimiento que ya está presente en el sistema.
- alumno. El usuario busca rápidamente desarrollar pericia personal relacionada con el área específica mediante la recuperación de conocimientos organizados y condensados del sistema.
- cliente. El usuario aplica la pericia del sistema a tareas específicas reales.

---

<sup>31</sup> SÁNCHEZ y Beltrán, Juan Pablo. *Op. cit.*

El reconocimiento de las caracterizaciones anteriores contrasta con la percepción de un simple papel (el cliente) de los sistemas tradicionales de software.<sup>32</sup>

### 2.5.8 Sistema experto VS. Sistema clásico

La siguiente tabla compara las características de ambos tipos de sistemas<sup>33</sup>

**Tabla N° 2.5 Tabla Comparativa entre un Sistema Clásico y un Sistema Experto**

Sistema Clásico	Sistema Experto
Conocimiento y procesamiento combinados en un programa	Base de conocimiento separada del mecanismo de procesamiento
No contiene errores	Puede contener errores
No da explicaciones, los datos sólo se usan o escriben	Una parte del sistema experto consiste en el módulo de explicación
Los cambios son tediosos	Los cambios en las reglas son fáciles
El sistema sólo opera completo	El sistema puede funcionar con pocas reglas
Se ejecuta paso a paso	La ejecución usa heurísticas y lógica
Necesita información completa para operar	Puede operar con información incompleta
Representa y usa datos	Representa y usa conocimiento

### 2.6. Representación del conocimiento

El conocimiento es la comprensión adquirida, implica aprendizaje, concienciación y familiaridad con una o más materias; el conocimiento se compone de ideas, conceptos, hechos y figuras, teorías, procedimientos y relaciones entre ellos, y formas de aplicar los procedimientos a la resolución práctica de problemas.

---

<sup>32</sup> ROLSTON, David W. *Op. cit.*

---

El conocimiento que ha de funcionar en un SE es el conocimiento heurístico; el conocimiento heurístico es aquel conocimiento que ayuda a las personas u ordenadores a aprender, es el uso de los conocimientos empíricos. Las reglas de pensamiento, los trucos, los procedimientos o cualquier tipo de información que nos ayuda en la resolución de problemas.<sup>34</sup>

La representación del conocimiento es un esquema o dispositivo utilizado para capturar los elementos esenciales del dominio de un problema. Una representación manipulable es aquella que facilita la computación. En representaciones manipulables, la información es accesible a otras entidades que usan la representación como parte de un cómputo.

Debido a la variedad de formas que el conocimiento puede asumir, los problemas involucrados en el desarrollo de una representación del conocimiento son complejos, interrelacionados y dependientes del objetivo. En términos generales, el conocimiento debe estar representado de tal forma que:

- Capture generalizaciones.
- Pueda ser comprendido por todas las personas que vayan a proporcionarlo y procesarlo.
- Pueda ser fácilmente modificado.
- Pueda ser utilizado en diversas situaciones aún cuando no sea totalmente exacto o completo.
- Pueda ser utilizado para reducir el rango de posibilidades que usualmente debería considerarse para buscar soluciones.

El conocimiento declarativo puede ser representado con modelos relacionales y esquemas basados en lógica. Los modelos relacionales pueden representar el conocimiento

---

<sup>33</sup> DE ÁVILA Ramos, Jorge. *Op. cit.*

<sup>34</sup> DE MIGUEL González, Luis Javier. *Op. cit.*

en forma de árboles, grafos o redes semánticas. Los esquemas de representación lógica incluyen el uso de lógica proposicional y lógica de predicados.

Los modelos procedimentales y sus esquemas de representación almacenan conocimiento en la forma de cómo hacer las cosas. Pueden estar caracterizados por gramáticas formales, usualmente implantadas por sistemas o lenguajes procedimentales y sistemas basados en reglas (sistemas de producción). 35

En forma natural, el ser humano representa el conocimiento simbólicamente: imágenes, lenguaje hablado y lenguaje escrito. Adicionalmente, ha desarrollado otros sistemas de representación del conocimiento: literal, numérico, estadístico, estocástico, lógico.

La ingeniería cognoscitiva ha adaptado diversos sistemas de representación del conocimiento que, implantados en un computador, se aproximan mucho a los modelos elaborados por la psicología cognoscitiva para el cerebro humano. Tradicionalmente la representación del conocimiento conlleva el uso de marcos (frames), redes semánticas, cálculo de predicados o sistemas de producción<sup>36</sup>. Sin embargo, existen otros sistemas para la representación del conocimiento. Entre los principales sistemas se tienen:

Lógica Simbólica Formal:

- Lógica proposicional
- Lógica de predicados.
- Reglas de producción.

Formas Estructuradas:

- Redes asociativas.
- Estructuras frame.

---

<sup>35</sup> ARTEAGA, René y Juan Carlos Armijos. **Tutorial de programación Heurística.**  
URL([www.uc3m.es/cgi-bin/nph-count](http://www.uc3m.es/cgi-bin/nph-count))

- Representación orientada a objetos.

### **2.6.1. Lógica Proposicional**

La lógica proposicional es la más antigua y simple de las formas de lógica. Utilizando una representación primitiva del lenguaje, permite representar y manipular aserciones sobre el mundo que nos rodea. La lógica proposicional permite el razonamiento a través de un mecanismo que primero evalúa sentencias simples y luego sentencias complejas, formadas mediante el uso de conectivos proposicionales, por ejemplo Y (AND), O (OR). Este mecanismo determina la veracidad de una sentencia compleja, analizando los valores de veracidad asignados a las sentencias simples que la conforman.

La lógica proposicional permite la asignación de un valor verdadero o falso para la sentencia completa, pero no tiene la facilidad de analizar las palabras individuales que componen la sentencia.

La principal debilidad de la lógica proposicional es su limitada habilidad para expresar conocimiento.<sup>37</sup>

### **2.6.2 Lógica de Predicados**

Existen varias sentencias complejas que pierden mucho de su significado cuando se les representa en lógica proposicional. Por esto se desarrolló una forma lógica más general, capaz de representar todos los detalles expresados en las sentencias, esta es la lógica de predicados.

La lógica de predicados está basada en la idea de que las sentencias realmente expresan relaciones entre objetos, así como también cualidades y atributos de tales objetos. Los objetos pueden ser personas, objetos físicos, o conceptos. Tales cualidades, relaciones

---

<sup>36</sup> DE ALBORNOZ Bueno, Álvaro. **Laboratorio de procesamiento de imágenes.**  
URL([http://www.cic.ipn.mx/organizacion/lab de int art.htm](http://www.cic.ipn.mx/organizacion/lab_de_int_art.htm))

o atributos, se denominan predicados. Los objetos se conocen como argumentos o términos del predicado.

Al igual que las proposiciones, los predicados tienen un valor de veracidad, pero a diferencia de las proposiciones, su valor de veracidad, depende de sus términos. Es decir, un predicado puede ser verdadero para un conjunto de términos, pero falso para otro. 38

### **2.6.2.1. Ventajas y desventajas de la lógica de predicados**

A continuación se presentan algunos aspectos característicos de la lógica de predicados y su implementación computacional, el lenguaje de programación PROLOG:

Manejo de incertidumbre. Una de las mayores desventajas de la lógica de predicados es que sólo dispone de dos niveles de veracidad: verdadero y falso. Esto se debe a que la deducción siempre garantiza que la inferencia es absolutamente verdadera. Sin embargo, en la vida real no todo es blanco y negro. En cierta forma el PROLOG ha logrado mitigar esta desventaja, permitiendo la inclusión de factores de certeza.

Razonamiento monotónico. La lógica de predicados al ser un formalismo de razonamiento monótono, no resulta muy adecuada para ciertos dominios del mundo real, en los cuales las verdades pueden cambiar con el paso del tiempo. PROLOG compensa esta deficiencia proporcionando un mecanismo para remover los hechos de la base de datos. Por ejemplo, en TURBO PROLOG se tiene la cláusula RETRACTALL.

Programación declarativa.- La lógica de predicados, tal como está diseñada en PROLOG, es un lenguaje de programación declarativo, en donde el programador sólo necesita preocuparse del conocimiento expresado en términos del operador de implicación y los axiomas. El mecanismo deductivo de la lógica de predicados llega a una respuesta (si esto es factible), utilizando un proceso exhaustivo de unificación y búsqueda. A pesar de

---

<sup>37</sup> *Ibid.*

<sup>38</sup> *Ibid.*

que la búsqueda exhaustiva puede ser apropiada en muchos problemas, también puede introducir ineficiencias durante la ejecución. Para lograr un cierto control en el proceso de búsqueda, PROLOG ofrece la operación de corte, CUT. Cuando no se utiliza el CUT, PROLOG se convierte en un lenguaje puramente declarativo.<sup>39</sup>

### **2.6.3 Reglas de Producción**

La Representación del conocimiento en forma de reglas de producción fue propuesta por Post en 1943. La regla es la forma más común de representar el conocimiento, debido a su gran sencillez y a que es la formulación más inmediata del principio de causalidad. Una regla consta de un conjunto de acciones o efectos (una o más) que son ciertas cuando se cumplen un conjunto de condiciones o causas. La potencia de una regla está en función de la lógica que admita en las expresiones de las condiciones y de las conclusiones.

La conclusión se suele referir a la creación de un nuevo hecho válido, o la incorporación de una nueva característica a un hecho, mientras que la acción suele referirse a la transformación de un hecho.<sup>40</sup>

#### **2.6.3.1. Ventajas y desventajas de las reglas de producción**

Las ventajas que representan las reglas de producción son su carácter declarativo, su sencillez, su uniformidad - que permite la representación de conocimiento como de meta-conocimiento -, su independencia - que permite la supresión o inclusión sin que se vea afectado el resto de la base de conocimientos - y su modularidad al ser fácilmente agrupables.

La principal desventaja que presentan las reglas de producción, es la dificultad de establecer relaciones, para lo cual hay que recurrir al uso de meta reglas, lo que produce el

---

<sup>39</sup> *Ibid.*

<sup>40</sup> RICH, Elaine y Kevin Knight. **Artificial Intelligence.**

crecimiento muy rápido del número de reglas, lo que hace lento el proceso de inferencia y conduce a la introducción de repeticiones y lo que es peor, contradicciones.<sup>41</sup>

#### **2.6.4 Redes Asociativas**

Las redes semánticas o redes asociativas, fueron originalmente desarrolladas para representar el significado o semántica de oraciones en inglés, en términos de objetos y relaciones. Actualmente, el término redes asociativas ya no sólo se usa para representar relaciones semánticas, sino también para representar asociaciones físicas o causales entre varios conceptos u objetos.

Las redes asociativas se caracterizan por representar el conocimiento en forma Gráfico. Agrupan una porción de conocimiento en dos partes: objetos y relaciones entre objetos. Los objetos se denominan también nodos (elementos del conocimiento) y las relaciones entre nodos se denominan enlaces o arcos. Cada nodo y cada enlace en una red semántica, deben estar asociados con objetos descriptivos.

Estas redes son muy apropiadas para representar conocimiento de naturaleza jerárquica. Su concepción se basa en la asociación de conocimientos que realiza la memoria humana. Las principales aplicaciones son: comprensión de lenguaje natural, bases de datos deductivas, visión por computadora y sistemas de aprendizaje.<sup>42</sup>

##### **2.6.4.1. Ventajas y desventajas de las Redes Asociativas**

Las redes asociativas tienen dos ventajas sobre los sistemas basados en reglas y sobre los sistemas basados en lógica:

- Permiten la declaración de importantes asociaciones, en forma explícita.

---

<sup>41</sup> BRATKO Ivan. **Prolog programming for Artificial Intelligence.**

<sup>42</sup> ANGULO Usategui, José María y Anselmo del Moral Bueno. **Guía fácil de la inteligencia artificial.**

- Debido a que los nodos relacionados están directamente conectados, y no se expresan las relaciones en una gran base de datos, el tiempo que toma el proceso de búsqueda por hechos particulares puede ser significativamente reducido.
- Entre las desventajas de las redes asociativas, se pueden mencionar:
- No existe una interpretación normalizada para el conocimiento expresado por la red. La interpretación de la red depende exclusivamente de los programas que manipulan la misma.
- La dificultad de interpretación a menudo puede derivar en inferencias inválidas del conocimiento contenido en la red.
- La exploración de una red asociativa puede derivar en una explosión combinatoria del número de relaciones que deben ser examinadas para comprobar una relación.<sup>43</sup>

### 2.6.5 Estructuras Frame

Una plantilla (frame) es una estructura de datos apropiada para representar una situación estereotípica. Las plantillas organizan el conocimiento en objetos y eventos que resultan apropiados para situaciones específicas. La evidencia psicológica sugiere que la gente utiliza grandes plantillas para codificar el conocimiento de experiencias pasadas, o conocimiento acerca de cosas que se encuentran comúnmente, para analizar y explicar una situación nueva en su cotidiana actividad cognoscitiva.

Una plantilla representa un objeto o situación describiendo la colección de atributos que posee. Cada plantilla está formada por un nombre y por una serie de campos de información o ranuras (slots). Cada ranura puede contener uno o más enlaces (facets). Cada enlace tiene un valor asociado. Varios enlaces pueden ser definidos para cada ranura, por ejemplo:

- Rango. El conjunto de posibles valores para la ranura.
- Valor. El valor de la ranura.

---

<sup>43</sup> ARTEAGA, René y Juan Carlos Armijos. *Op. cit.*

- Default. El valor a ser asumido si no se especifica alguno.

Además los enlaces pueden ser procedimientos que residen en la base de datos y están aguardando para ser utilizados cuando se les necesite. Entre los más comunes se pueden mencionar:

- Si-Necesitado. Procedimiento(s) para determinar el valor actual de una ranura.
- Si-Agregado. Procedimiento(s) a ejecutarse cuando un valor es especificado para una ranura.
- Si-Modificado. Procedimiento(s) a ejecutarse si el valor de una ranura es cambiado.

A estos procedimientos también se los denomina demons y representan un concepto poderoso en las plantillas, esto es, la habilidad de combinar conocimiento procedimental dentro de la estructura de conocimiento declarativo de la plantilla. Esto sugiere que una plantilla puede ser un medio poderoso de representación del conocimiento, especialmente si se la incorpora en una red de plantillas.<sup>44</sup>

#### **2.6.5.1. Ventajas y desventajas de los Frames**

Las ventajas que se pueden establecer para los sistemas basados en plantillas son las siguientes:

Facilidad de proceso guiado por las expectativas. Un sistema basado en plantillas, mediante los demons es capaz de especificar acciones que deben tener lugar cuando ciertas condiciones se han cumplido durante el procesamiento de la información.

El conocimiento que posee un sistema basado en plantillas es significativamente más estructurado y organizado que el conocimiento dentro de una red asociativa.

---

<sup>44</sup> BENCHIMOL, Guy, Pierre Levine y Jean Charles Promerol. **Los sistemas expertos en la empresa.**

Las plantillas pueden ser estructuradas de tal forma que sean capaces de determinar su propia aplicabilidad en determinadas situaciones. En el caso de que una plantilla en particular no sea aplicable, puede sugerir otras plantillas que pueden ser apropiadas para la situación.

Se puede fácilmente almacenar en las ranuras valores dinámicos de variables, durante la ejecución de un sistema basado en conocimiento. Esto puede ser particularmente útil para aplicaciones de simulación, planeamiento, diagnóstico de problemas o interfaces para bases de datos.

Las principales desventajas que se pueden establecer para la representación del conocimiento mediante plantillas, son:

- Dificultad de representar objetos que se alejen considerablemente de estereotipos.
- No tiene la posibilidad de acomodarse a situaciones u objetos nuevos.
- Dificultad para describir conocimiento heurístico que es mucho más fácilmente representado mediante reglas.<sup>45</sup>

#### **2.6.6. Representación orientada a objetos**

Los objetos, son similares a las plantillas. Ambos sirven para agrupar conocimiento asociado, soportan herencia, abstracción y el concepto de procedimientos agregados. La diferencia radica en lo siguiente:

En las plantillas, a los programas y a los datos se los trata como dos entidades relacionadas separadas. En cambio en los objetos se crea una fuerte unidad entre los procedimientos (métodos) y los datos.

Los demons de las plantillas sirven sólo para computar valores para las diversas ranuras o para mantener la integridad de la base de conocimientos cada vez que una acción

de alguna plantilla, afecta a otra. En cambio, los métodos utilizados por los objetos son más universales ya que proporcionan cualquier tipo general de computación requerida y además soportan encapsulamiento y polimorfismo.

Un objeto es definido como una colección de información que representa una entidad del mundo real y una descripción de cómo debe ser manipulada esta información, esto es, los métodos. Es decir, un objeto tiene un nombre, una caracterización de clase, varios atributos distintivos y un conjunto de operaciones. La relación entre los objetos viene definida por los mensajes. Cuando un objeto recibe un mensaje válido, responde con una acción apropiada, retornando un resultado.<sup>46</sup>

#### **2.6.6.1. Ventajas y desventajas de la representación orientada a objetos.**

Los objetos, como forma de representación del conocimiento ofrecen las siguientes ventajas:

- Poder de abstracción.
- Encapsulamiento o capacidad de esconder información.
- Herencia, es decir pueden recibir características de sus ancestros.
- Polimorfismo, que permite crear una interfaz común para todos los diversos objetos utilizados dentro del dominio.
- Posibilidad de reutilización del código.
- Mayor facilidad para poder trabajar eficientemente con sistemas grandes.

Las desventajas son similares a las que se indicaron para las plantillas:

- Dificultades para manejar objetos que se alejan demasiado de la norma.
- Dificultades para manejar situaciones que han sido encontradas previamente.<sup>47</sup>

---

<sup>45</sup> ARTEAGA, René y Juan Carlos Armijos. *Op cit.*

<sup>46</sup> *Ibid.*

<sup>47</sup> *Ibid*

## 2.7. Componentes de un sistema experto

Los dos componentes principales de cualquier Sistema Experto son una BASE DE CONOCIMIENTOS y un PROGRAMA DE INFERENCIA, o también llamado MOTOR DE INFERENCIAS<sup>48</sup>.

La base de conocimientos del Sistema Experto con respecto a un tema específico para el que se diseña el sistema. Este conocimiento se codifica según una notación específica que incluye reglas, predicados, redes semánticas y objetos.

El motor de inferencia, que es el que combina los hechos y las preguntas particulares, utilizando la base de conocimiento, seleccionando los datos y pasos apropiados para presentar los resultados.<sup>49</sup>

Esta definición de las partes de un Sistema Experto es muy general, ahora se presenta una serie de componentes más detallados de un SE:

- Subsistema de control de coherencia. Este componente previene la entrada de información incoherente en la base de conocimiento. Es un componente muy necesario, a pesar de ser un componente reciente.
- Subsistema de adquisición de conocimiento. Se encarga de controlar si el flujo de nuevo conocimiento a la base de datos es redundante. Sólo almacena la información que es nueva para la base de datos.
- Motor de inferencia. Este componente es básico para un SE; se encarga de obtener conclusiones comenzando desde el conocimiento abstracto hasta el conocimiento concreto. Si el conocimiento inicial es muy poco, y el sistema no puede obtener ninguna conclusión, se utilizará el subsistema de demanda de información.
- Subsistema de demanda de información. Completa el conocimiento necesario y reanuda el proceso de inferencia hasta obtener alguna conclusión válida. El usuario

---

<sup>48</sup> DE MIGUEL González, Luis Javier. *Op. cit*

<sup>49</sup> CRIADO Briz, José Mario. **Sistemas Expertos**. *Op. cit.*

puede indicar la información necesaria en este proceso ayudado de una interfase de usuario (la cual facilita la comunicación entre el Sistema Experto y el usuario).

- Subsistema de incertidumbre. Se encarga de almacenar la información de tipo incierto y propaga la incertidumbre asociada a esta información.
- Subsistema de ejecución de tareas. Permite realizar acciones al Sistema Experto basadas en el motor de inferencia.
- Subsistema de explicación. Este componente entra en ejecución cuando el usuario solicita una explicación de las conclusiones obtenidas por el SE. Esto se facilita mediante el uso de una interfase.<sup>50</sup>

Muchos investigadores hacen representaciones visuales de los componentes de un SE. Por ejemplo, De Miguel muestra su diagrama de un SE en Gráfico N° 2.9.

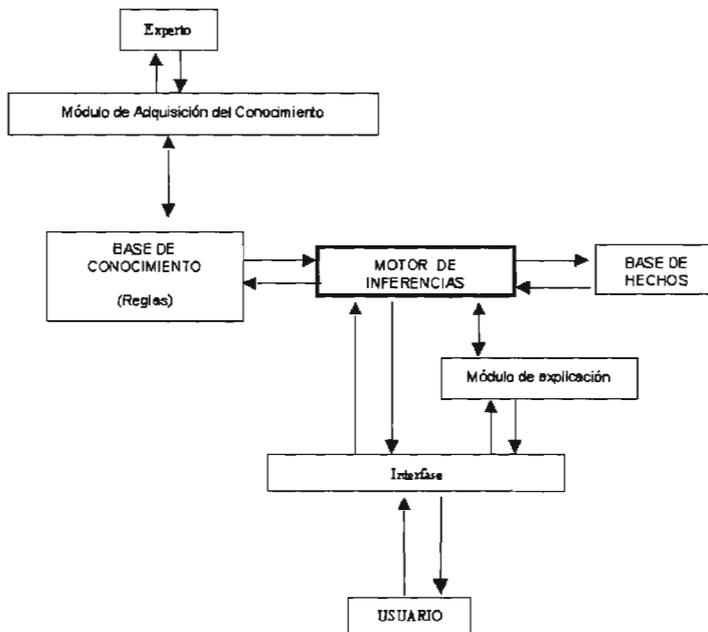


Gráfico N° 2.9 Estructura de un Sistema Experto según De Miguel

<sup>50</sup> SELL, Peter. **Sistemas Expertos para principiantes.**

Otra representación de componentes de un Sistema Experto es la de Samper quien nos dice que no existe una estructura común para un Sistema Experto. Sin embargo, la mayoría de los sistemas expertos tienen ciertos componentes básicos: base de conocimientos, motor de inferencia, base de datos e interfaz con el usuario. Muchos tienen, además, un módulo de explicación y un módulo de adquisición del conocimiento. El Gráfico N° 2.10 muestra la estructura de un Sistema Experto ideal.

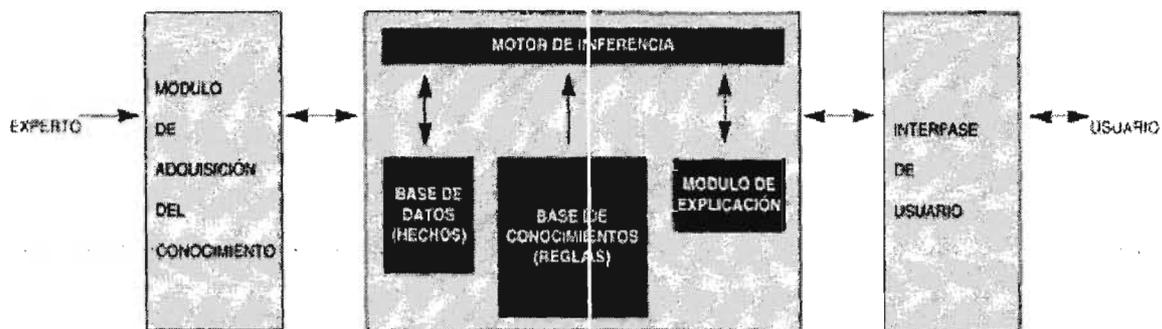


Gráfico N° 2.10 Estructura de un Sistema Experto ideal para Samper

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

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

Las reglas suelen almacenarse en alguna secuencia jerárquica lógica, pero esto no es estrictamente necesario. Se pueden tener en cualquier secuencia y el motor de inferencia las usará en el orden adecuado que necesite para resolver un problema.<sup>51</sup>

La base de datos o base de hechos es una parte de la memoria del ordenador que se utiliza para almacenar los datos recibidos inicialmente para la resolución de un problema. Contiene conocimiento sobre el caso concreto en que se trabaja. También se registrarán en ella las conclusiones intermedias y los datos generados en el proceso de inferencia. Al memorizar todos los resultados intermedios, conserva el vestigio de los razonamientos efectuados; por lo tanto, se puede utilizar explicar las deducciones y el comportamiento del sistema.

El motor de inferencias es un programa que controla el proceso de razonamiento que seguirá el Sistema Experto. Utilizando los datos que se le suministran, recorre la base de conocimientos para alcanzar una solución. La estrategia de control puede ser de encadenamiento progresivo o de encadenamiento regresivo. En el primer caso se comienza con los hechos disponibles en la base de datos, y se buscan reglas que satisfagan esos datos, es decir, reglas que verifiquen la parte SI. Normalmente, el sistema sigue los siguientes pasos:

- Evaluar las condiciones de todas las reglas respecto a la base de datos, identificando el conjunto de reglas que se pueden aplicar (aquellas que satisfacen su parte condición).
- Si no se puede aplicar ninguna regla, se termina sin éxito; en caso contrario se elige cualquiera de las reglas aplicables y se ejecuta su parte acción (esto último genera nuevos hechos que se añaden a la base de datos).
- Si se llega al objetivo, se ha resuelto el problema; en caso contrario, se vuelve al paso 1
- A este enfoque se le llama también guiado por datos, porque es el estado de la base de datos el que identifica las reglas que se pueden aplicar. Cuando se utiliza este

---

<sup>51</sup> SAMPER Márquez, Juan José. *Op. cit.*

.....

método, el usuario comenzará introduciendo datos del problema en la base de datos del sistema.

Al encadenamiento regresivo se le suele llamar guiado por objetivos, ya que, el sistema comenzará por el objetivo (parte acción de las reglas) y operará retrocediendo para ver cómo se deduce ese objetivo partiendo de los datos. Esto se produce directamente o a través de conclusiones intermedias o subobjetivos. Lo que se intenta es probar una hipótesis a partir de los hechos contenidos en la base de datos y de los obtenidos en el proceso de inferencia.

En la mayoría de los Sistemas Expertos se utiliza el encadenamiento regresivo. Este enfoque tiene la ventaja de que el sistema va a considerar únicamente las reglas que interesan al problema en cuestión. El usuario comenzará declarando una expresión E y el objetivo del sistema será establecer la verdad de esa expresión.

Para ello se pueden seguir los siguientes pasos:

- Obtener las reglas relevantes, buscando la expresión E en la parte acción (éstas serán las que puedan establecer la verdad de E)
- Si no se encuentran reglas para aplicar, entonces no se tienen datos suficientes para resolver el problema; se termina sin éxito o se piden al usuario más datos.
- Si hay reglas para aplicar, se elige una y se verifica su parte condición C con respecto a la base de datos.
- Si C es verdadera en la base de datos, se establece la veracidad de la expresión E y se resuelve el problema.
- Si C es falsa, se descarta la regla en curso y se selecciona otra regla.
- Si C es desconocida en la base de datos (es decir, no es verdadera ni falsa), se le considera como sub-objetivo y se vuelve al paso 1 (C será ahora la expresión E).

Existen también enfoques mixtos en los que se combinan los métodos guiados por datos con los guiados por objetivos.

La interfaz de usuario permite que el usuario pueda describir el problema al Sistema Experto. Interpreta sus preguntas, los comandos y la información ofrecida. A la inversa, formula la información generada por el sistema incluyendo respuestas a las preguntas, explicaciones y justificaciones. Es decir, posibilita que la respuesta proporcionada por el sistema sea inteligible para el interesado. También puede solicitar más información al SE si le es necesaria. En algunos sistemas se utilizan técnicas de tratamiento del lenguaje natural para mejorar la comunicación entre el usuario y el SE.

La mayoría de los sistemas expertos contienen un módulo de explicación, diseñado para aclarar al usuario la línea de razonamiento seguida en el proceso de inferencia. Si el usuario pregunta al sistema cómo ha alcanzado una conclusión, éste le presentará la secuencia completa de reglas usada. Esta posibilidad de explicación es especialmente valiosa cuando se tiene la necesidad de tomar decisiones importantes amparándose en el consejo del SE. Además, de esta forma, y con el tiempo suficiente, los usuarios pueden convertirse en especialistas en la materia, al asimilar el proceso de razonamiento seguido por el sistema. El subsistema de explicación también puede usarse para depurar el SE durante su desarrollo.

El módulo de adquisición del conocimiento permite que se puedan añadir, eliminar o modificar elementos de conocimiento (en la mayoría de los casos reglas) en el SE. Si el entorno es dinámico, entonces este componente es muy necesario, puesto que el sistema funcionará correctamente sólo si se mantiene actualizado su conocimiento. El módulo de adquisición permite efectuar ese mantenimiento, anotando en la base de conocimientos los cambios que se producen.<sup>52</sup>

## **2.8. Construcción de un sistema experto**

La construcción de un SE no es una tarea sencilla, debido a que involucra mucha participación de distintas personas, cada una de las cuales aportará algo para que el SE a

---

<sup>52</sup> SAMPER Márquez, Juan José. *Op. cit.*

desarrollar sea robusto y fácil de usar y mantener. Además se deben hacer varias elecciones en cuanto al desarrollo del Sistema Experto.

La primera decisión consiste en determinar si se comenzará el SE desde cero o se utilizará un shell – que es un SE sin la base de conocimientos –. Si se opta por usar el shell se debe elegir el que más se adecue al objetivo del SE que se desea construir, ya que existen diversos shells de Sistemas Expertos encaminados hacia distintos objetivos.

Si por el contrario se opta por comenzar desde cero, se deberá entonces determinar qué metodología utilizar, es decir, determinar la guía para el desarrollo del SE, cómo se implementará la base de conocimientos y el motor de inferencia, principalmente; y como complemento se debe elegir el lenguaje que se va a utilizar para el proyecto.

## **2.8.1 Aspectos Generales de la Construcción de un Sistema Experto**

### **2.8.1.1 El Equipo de Desarrollo**

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

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

- El experto. La función del experto es la de poner sus conocimientos especializados a disposición del Sistema Experto.
- El ingeniero del conocimiento. Es el ingeniero que plantea las preguntas al experto, estructura sus conocimientos y los implementa en la base de conocimientos.

- El usuario. El usuario aporta sus deseos y sus ideas, determinando especialmente el escenario en el que debe aplicarse el Sistema Experto.

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

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

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

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

La separación entre experto e ingeniero del conocimiento permanece, por regla general, inalterada.<sup>53</sup>

### **2.8.1.2 Construcción de las principales partes de un SE**

Como ya se mencionó, los principales componentes de un SE son la base de conocimiento y el motor de inferencia. La metodología tradicional se enfoca principalmente a la construcción de dichos componentes.

#### **La base de conocimiento**

La base de conocimiento es una base de datos que posee una información y unas reglas específicas sobre una materia o tema determinado. En la metodología tradicional, la base de conocimiento se conforma de objetos y atributos. Un objeto es una conclusión que es definida por sus atributos relacionados; y un atributo es una característica o cualidad específica que ayuda a definir un objeto.

La base de conocimiento se puede ver como una lista de objetos con sus reglas y atributos asociados. En el sentido más simple, la regla que se aplica para un atributo determina si un objeto “tiene” o “no tiene” determinado atributo. Por lo que un objeto se define usando una lista de atributos que el objeto posea o no.<sup>54</sup>

#### **El motor de inferencia**

Para construir un motor de inferencia, existen tres métodos básicos: encadenamiento hacia adelante, encadenamiento hacia atrás y reglas de producción.<sup>55</sup>

#### **Método de encadenamiento hacia adelante**

---

<sup>53</sup> CRIADO Briz, José Mario. **Sistemas Expertos**. *Op. cit.*

<sup>54</sup> SCHILDT Herbert. *Op. cit*

A este método se le llama conducido por datos, porque el motor de inferencia utiliza la información que el usuario le proporciona para moverse a través de una red de operadores AND y operadores OR hasta que encuentra un punto terminal que es el objeto. Si el motor de inferencia no puede encontrar un objeto que cumpla con los requisitos, el SE pide más información. Los atributos que definen al objeto crean un camino que conduce al mismo objeto: la única forma de alcanzar dicho objeto es satisfacer todas sus reglas. En resumen el sistema de encadenamiento hacia delante construye un árbol desde las hojas hasta la raíz, como se puede ver en la Gráfico N° 2.11

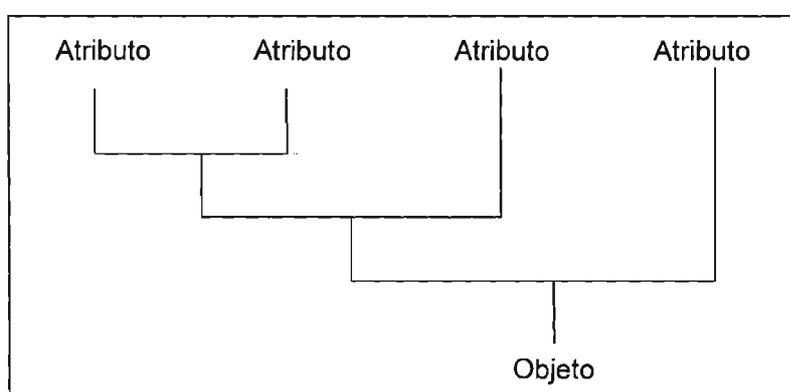


Gráfico N° 2.11 Método de encadenamiento hacia adelante

### **Método de encadenamiento hacia atrás**

Este método es el contrario al método anterior. Un motor de inferencia de encadenamiento hacia atrás comienza con una hipótesis (objeto) y pide información para confirmarlo o negarlo. A este método se le llama conducido por objetos porque el SE empieza con un objeto e intenta verificarlo. En resumen, el encadenamiento hacia atrás poda un árbol, lo cual es lo contrario al método anterior, en el cual se construye un árbol, esto se muestra en la Gráfico N° 2.12

---

<sup>55</sup> *Ibid.*

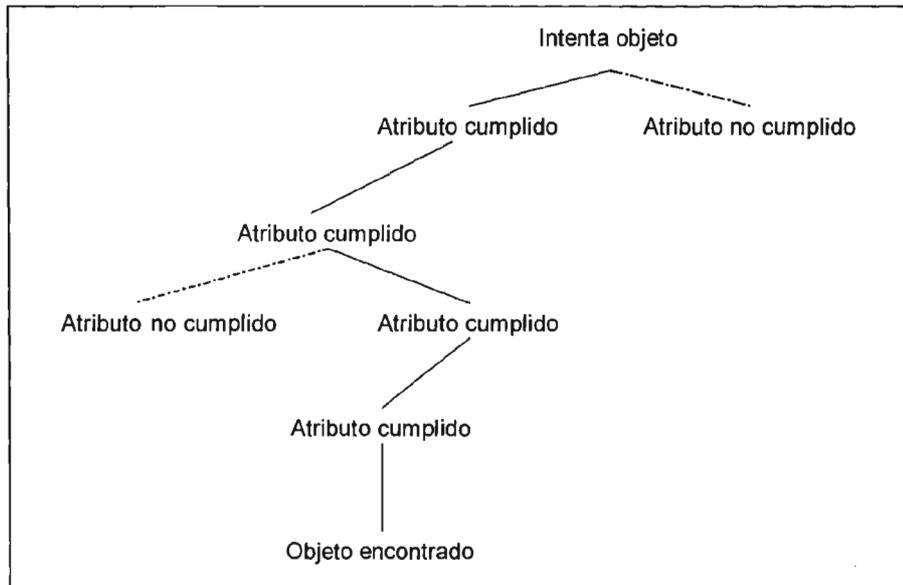


Gráfico N° 2.12. Encadenamiento hacia atrás

### Método de reglas de producción

Este método es una mejora al método de encadenamiento hacia atrás. La teoría operativa general es que el sistema pide como información aquella que elimine la mayor incertidumbre posible. El problema que existe con este método está en su dificultad para llevarse a cabo, primeramente debido a que la base de conocimiento a menudo es tan grande que el número de combinaciones que el sistema puede soportar se rebasa fácilmente, por lo tanto, el sistema no puede determinar exactamente qué información eliminará una mayor incertidumbre. En segundo lugar, los sistemas de reglas de producción requieren que la base de conocimientos contenga no sólo la información objeto – atributo, sino además un valor cuantificador, lo que hace aun más difícil la construcción de la base de conocimientos.

### 2.8.2 Metodologías para la construcción de SE

#### 2.8.2.1 Metodología de Prototipos

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

- No existen implementaciones similares que puedan servir de orientación al encargado del desarrollo en casi la totalidad de los casos.
- En muchos puntos, los requisitos necesarios están esbozados con muy poca precisión.
- El diseño y la especificación requieren una temprana determinación de la interfaz del software y de la funcionalidad de los componentes. Durante el desarrollo, resulta apropiado empezar con implementaciones tipo test para encontrar el camino hacia una solución definitiva y para hacerlas coincidir con las necesidades del usuario.

Un método efectivo es la implementación de un prototipo de Sistema Experto que permita llevar a cabo las funciones más importantes de éste, aunque con un esfuerzo de desarrollo considerablemente inferior al de una implementación convencional. Este proceder se define bajo el nombre de ‘Rapid Prototyping’. Para Sistemas Expertos, el ‘Rapid Prototyping’ es el procedimiento más adecuado, pues posibilita una rápida reacción a los deseos en constante cambio tanto por parte de los expertos como parte del usuario.<sup>56</sup>

### 2.8.2.2 Metodología Orientada a Objetos

La programación orientada a objetos (OOP en adelante) es un tipo de programación que provee una manera de modularizar programas estableciendo áreas de memoria particionadas para datos y procedimientos, que pueden ser usadas como plantillas para crear copias de tales módulos conforme se requieran.<sup>57</sup>

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

---

<sup>56</sup> CRIADO Briz, José Mario. **Sistemas Expertos**. *Op. cit.*

<sup>57</sup> TELLO, Ernest. **Object oriented programming for artificial intelligence**.

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

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

Por lo tanto si se utiliza la metodología orientada a objetos a la construcción de SE, se debe crear una clase que contenga el código referente a la base de conocimiento. Si dicha base de conocimiento es un conjunto de objetos con sus propiedades, entonces dicha clase será una clase contenedor de objetos de otra clase, los cuales contendrán el nombre del objeto y una lista de atributos. La Gráfico N° 2.13 muestra lo expuesto.

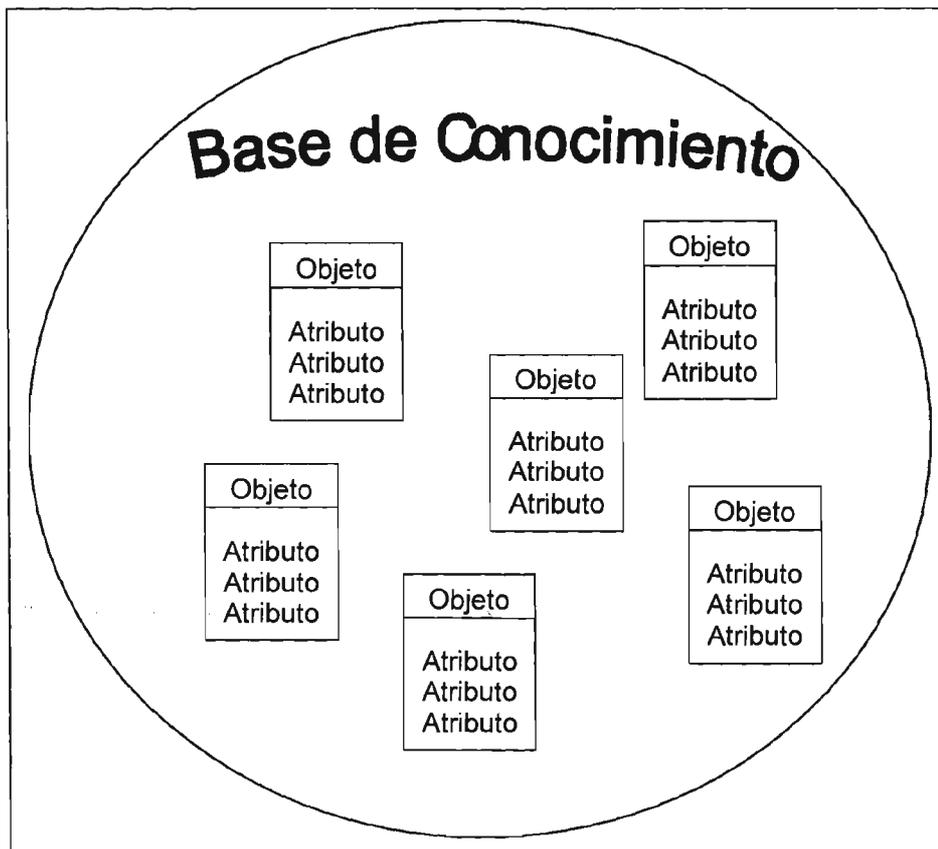


Gráfico N° 2.13 La base de conocimientos es vista como un objeto contenedor de otros objetos usando la Metodología Orientada a Objetos

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

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

Esta metodología tiene diversas ventajas, entre las cuales destaca el hecho que cualquier modificación o mantenimiento que se le quiera realizar a un determinado componente no afectará al otro y viceversa. Tello propone crear especializaciones o

jerarquías de clases (usando herencia) que incluyan métodos que permitan a los objetos modificarse a sí mismos, logrando con esto que un SE aprenda por sí mismo, modificando su base de conocimiento sin modificar los demás componentes. Otra ventaja de la OOP es que mediante la herencia, las clases hijas heredarán el comportamiento de las clases padres, evitando con esto, tener que incluir código redundante en las especializaciones.<sup>58</sup>

### 2.8.3 Lenguajes utilizados en la construcción de SE

#### 2.8.3.1 LISP

LISP, acrónimo de lenguaje de Procesamiento de Listas, fue inventado por John McCarthy y su equipo en la Universidad de Stanford a finales de 1950. Originalmente fue creado como un modelo computacional de procesos matemáticos, reflejando el rigor de las propias matemáticas.<sup>59</sup>

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

LISP utiliza un ciclo llamado leer – evaluar – imprimir. Cuando el programador teclea algo en LISP, el lenguaje toma lo que se ha tecleado, intenta responder de cualquier forma y después despliega dicha respuesta en la pantalla. El término en LISP usado para “ver lo que se ha tecleado” es leer; el término usado para “tratar de responder de cualquier

---

<sup>58</sup> *Ibid.*

<sup>59</sup> HASEMER, Tony y John Domingue. **Common LISP Programming for Artificial Intelligence.**

forma” es evaluar; y el término usado para “desplegar la respuesta en la pantalla” es imprimir. Si hubiera una persona en vez de una máquina en frente del programador, diríamos que dicha persona estaría escuchando, comprendiendo y respondiendo al programador.<sup>60</sup>

LISP utiliza los siguientes conceptos característicos:

- Listas y Átomos. La estructura más importante es la lista. Los átomos pueden subordinarse a cualidades.
- La Función. Cada función LISP y cada programa LISP tiene estructura de lista. Los programas no pueden distinguirse sintácticamente de los datos. LISP ofrece sus propias funciones básicas.
- Forma de Trabajo. LISP es un lenguaje funcional. Ofrece la posibilidad de realizar definiciones recursivas de funciones. La unión de procedimientos se realiza de forma dinámica, es decir en plena ejecución, no como en otros lenguajes de programación. El sistema realiza automáticamente una gestión dinámica de memoria.
- Los átomos son números, cadenas de caracteres o símbolos. Un símbolo puede tener varios valores, al igual que una variable en otros lenguajes de programación, como por ejemplo un número, o también puede ser el nombre de una función, o incluso ambos. Además a un símbolo se le pueden subordinar cualidades, que además del valor del símbolo, contienen información adicional. Estas cualidades también reciben el nombre de atributos.

Un componente importante de un sistema LISP es la gestión dinámica de la memoria. El sistema administrará el espacio en la memoria para las listas en constante modificación, sin que el usuario lo deba solicitar. Libera los espacios de memoria que ya no son necesarios y los pone a disposición de usos posteriores. La necesidad de este proceso se deriva de la estructura básica de LISP, las listas, que se modifican de forma dinámica e ilimitada.

---

<sup>60</sup> *Ibid.*

Además un sistema LISP abarca bastante más que el solo intérprete del lenguaje LISP. Consta de algunos cómodos módulos que ofrecen ayuda en el desarrollo y control del progreso en programas, como son el Editor, el File-System y el Trace. Por supuesto estos módulos sólo están en versiones de LISP que contengan la conocida interfaz Gráfico IDE (entorno de desarrollo integrado) típica de los modernos lenguajes visuales.

### 2.8.3.2 CLIPS

CLIPS es otra herramienta para el desarrollo de S.E. que ofrece un entorno completo para su construcción basado en reglas y objetos. CLIPS está siendo utilizado por numerosos usuarios como la NASA (que es su creadora), muchas universidades y empresas. CLIPS viene de (C Language Integrated Production System) y como su nombre indica uno de los objetivos que buscaban sus creadores era la fácil integración con programas en C. Así, al darle una portabilidad con programas en lenguaje C, las universidades que lo usan pueden trasladar fácilmente sus aplicaciones al entorno del agente.<sup>61</sup>

### 2.8.3.3 Prolog

Prolog es un lenguaje de programación que se centra alrededor de un conjunto pequeño de mecanismos, incluyendo reconocimiento de patrones, estructuras de datos basadas en árboles y backtracking (retroceso) automático. Este conjunto pequeño constituye una estructura de programación sorprendentemente poderosa y flexible. Prolog es ideal para resolver problemas que involucren objetos – en particular objetos estructurados – y relaciones entre ellos. Por ejemplo, un ejercicio muy sencillo en Prolog es expresar relaciones espaciales, de la forma: “la esfera azul detrás de la verde”.<sup>62</sup>

Prolog nació en Europa, y fue implementado primeramente para dar soporte al Procesamiento del Lenguaje Natural (PLN). Prolog trabajaba (y sigue haciéndolo hasta

---

<sup>61</sup> BAÑARES, José Ángel. **Herramientas para las asignaturas del IAIC.**  
URL (<http://diana.cps.unizar.es/IA/noticias.html>)

nuestros días) esencialmente con la lógica matemática.<sup>63</sup> Bratko completa el comentario afirmando que Prolog nació de una idea de programación lógica que emergió a principios de 1970, para usar la lógica como un lenguaje de programación.<sup>64</sup>

Prolog tiene cierto vigor híbrido en el sentido de que contiene características declarativas de la lógica computacional matemática y algunos aspectos procedurales de la programación convencional. En vez de escribir un procedimiento con una secuencia de pasos, un programador Prolog escribe un conjunto declarativo de reglas y hechos con sus respectivas relaciones. Debido a esto, los diagramas de flujos y las técnicas convencionales de programación no aplican en Prolog.

Walker afirma que niños de 10 años de edad encuentran a Prolog como un lenguaje sencillo de aprender. Pero los programadores profesionales con muchos años de experiencia a menudo lo encuentran revuelto y confuso. Lo revuelto de Prolog es su simplicidad y que no tiene las características más comunes de los lenguajes procedurales.<sup>65</sup>

Entre tales características con las que no cuenta Prolog se encuentran:

- Sentencias de asignación.
- Sentencias goto.
- Sentencias if – then – else.
- Ciclos do, ciclos for y ciclos while.

Sin embargo, Prolog compensa tales características con las siguientes, las cuales lo hacen un lenguaje poderoso para la representación del conocimiento:

- Predicados que expresan relaciones entre entidades.

---

<sup>62</sup> BRATKO Iván. *Op. cit.*

<sup>63</sup> WALKER Adrian, Michael Mc Cord, John Sowa y Walter Wilson. **Knowledge – Based Systems and Prolog.**

<sup>64</sup> BRATKO Ivan. *Op. cit.*

<sup>65</sup> WALKER Adrian, Michael Mc Cord, John Sowa y Walter Wilson. *Op. cit*

- Un método para definir predicados mediante reglas de aserción y hechos.
- Un método para hacer preguntas para comenzar cálculos.
- Un procedimiento de búsqueda hacia atrás para evaluar metas.
- Estructura de datos que pueden simular registros estilo Pascal o listas estilo Lisp.
- Un reconocedor de patrones que construye y analiza las estructuras de datos.
- Un conjunto de predicados preconstruidos para aritmética, entrada y salida y servicios de sistemas.

#### **2.8.3.4 Smalltalk**

Smalltalk fue el primer lenguaje de programación que fue diseñado para basarse exclusivamente en objetos. Fue originalmente inventado por Alan Kay en Xerox PARC en 1972, pero mucha gente le ha hecho importantes contribuciones al diseño del lenguaje. Este lenguaje se ha convertido en una opción muy popular en diversos campos como los videojuegos y la Inteligencia Artificial.

Virtualmente todo lo que existe en un sistema Smalltalk es una instancia de una clase particular de objeto y generalmente puede haber tantas instancias como se deseen. Esto significa que se pueden tener cualquier número de instancias de cualquier característica del sistema activas al mismo tiempo.<sup>66</sup>

#### **2.8.3.5. C y C++**

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

---

<sup>66</sup> TELLO, Ernest. *Op. cit.*

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

En cuanto a C++ se puede decir que es una extensión orientada a objetos de C, la cual permite utilizar la metodología orientada a objetos para la creación de diversos sistemas y programas, incluso los de IA. <sup>67</sup>

#### **2.8.4 Herramientas y shells utilizados en la construcción de SE**

Inicialmente cada SE que se creaba se construía a partir de un lenguaje de IA tal como LISP. Pero después de que muchos SE se construyeron así, quedó claro que estos sistemas estaban contruidos como un conjunto de representaciones declarativas (reglas) combinado con un intérprete de estas representaciones; también quedó claro que era posible separar el intérprete del conocimiento específico del dominio y por lo tanto se podían crear sistemas que podían ser usados para construir nuevos SE agregando simplemente conocimiento correspondiente al dominio del nuevo problema.

Así nacieron los shells los cuales sirven como base para muchos SE, ya que proporcionan mucha flexibilidad en representación del conocimiento y razonamiento, así como adquisición de conocimiento. <sup>68</sup>

A continuación se mencionan algunos de los principales shells de SE, que como ya se mencionó, son SE sin su base de conocimiento, es decir, se pueden utilizar de forma genérica para cualquier tema.

##### **2.8.4.1 Gold Works II**

---

<sup>67</sup> SCHILDT Herbert. *Op. cit.*

<sup>68</sup> RICH Elaine y Kevin Knight. *Op. cit.*

---

Esta herramienta de programación creada por Gold Hill Computers Inc. se ejecuta bajo LISP y utiliza la metodología orientada a objetos. Este programa corre en computadoras IBM compatibles, Macintosh y estaciones de trabajo Sun.

Las características que realzan a este programa incluyen una herramienta dinámica de gráficos que permite la creación de imágenes activas así como también permite desarrollar interactivamente interfaces Gráficos dinámicas sin tener que escribir ni una línea de código; también se incluyen visores orientados a gráficos para marcos, reglas y afirmaciones (aserciones). Se integra fácilmente con C.

#### **2.8.4.2 ART**

ART es una herramienta que tiene un número de características poderosas para el desarrollo de SE, por ejemplo, permite la construcción de varios escenarios o mundo hipotéticos, los cuáles pueden ser explorados automáticamente para determinar su deseabilidad. De esta forma varios escenarios alternos pueden ser explorados para probar un plan estratégico dado con un conjunto de sentencias dado.

La característica de esquema es otro aspecto de este programa el cual provee una representación orientada a objetos para sistemas resolutores de problemas basados en conocimientos.<sup>69</sup>

#### **2.8.4.3 LOOPS**

LOOPS fue desarrollado en el Xerox PARC en 1983. Una de las ideas centrales en el diseño del ambiente LOOPS es proveer un sistema de programación de IA que pueda soportar una estructura de múltiples paradigmas que permitan tantas opciones de entre paradigmas de programación como sea posible. La versión actual soporta cuatro paradigmas de programación principales: el paradigma orientado a objetos, el paradigma basado en reglas, el paradigma orientado a accesos y el paradigma procedural normal.

Como cualquier otro sistema de programación orientado a objetos, LOOPS permite la creación de jerarquías de clases e instancias de dichas clases.

#### **2.8.4.4 KEE**

KEE, acrónimo de Ambiente de Ingeniería del Conocimiento, de IntelliCorp es una de las herramientas de desarrollo orientados a objetos de SE más avanzada en la actualidad. Hoy en día tiene un considerable uso en esfuerzos mayores de desarrollo tanto en el sector comercial como en el gobierno.

Las funciones en KEE se implementan como extensiones de LISP, de manera que todo el vocabulario del lenguaje LISP puede usarse en conjunción con las funciones predefinidas. El sistema entero está construido en una estructura de objetos, que en el ambiente son llamados unidades.

#### **2.8.4.5 Humble**

Humble es un Shell orientado a objetos para SE escrito en Smalltalk que combina reglas con encadenamiento hacia delante y hacia atrás con representación de objetos, paso de mensajes y uso de objetos. Las reglas de sintaxis usadas en Humble son una versión modificada de la sintaxis de Smalltalk. A diferencia de la mayoría de los shells, Humble tiene la capacidad de construir bloques if – then – else.

La característica principal de Humble es que las reglas operan sobre entidades. Las entidades son un importante tipo de objeto que tienen una representación específica. En las aplicaciones las entidades se categorizan en un número de tipos diferentes los cuáles son definidos por el desarrollador o por el Ingeniero de conocimientos.

#### **2.8.4.6 EMYCIN**

---

<sup>69</sup> TELLO Ernest. *Op. cit.*

EMYCIN es acrónimo de Empty MYCIN (MYCIN vacío); este shell fue creado a partir de MYCIN, uno de los SE más exitosos en la época de los setentas. Se construyó con el fin de que fuera usado para otros fines, además del diagnóstico de enfermedades en la sangre (objetivo para el cual fue creado). Este shell utiliza un lenguaje, llamado ARL, el cual es una notación de Algol y provee facilidades para monitorear el comportamiento de reglas. Sin embargo es difícil distinguir diferentes tipos de conocimiento (no se puede hacer diferencia entre conocimiento de control y conocimiento a cerca de los valores esperados de los parámetros), por lo que el agregar una nueva regla también se vuelve un proceso complicado.<sup>70</sup>

## **2.9. Tareas que realizan los Sistemas expertos**

### **2.9.1 Interpretación, diagnóstico y monitoreo**

La interpretación consiste en encontrar el significado de los datos de entrada obtenidos por sensores o introducidos por el usuario.<sup>71</sup>

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

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

El diagnóstico consiste en identificar las causas internas que provocan un problema, avería o disfunción a partir de una serie de datos o síntomas que son consecuencia de la misma y que son observables.<sup>72</sup>

---

<sup>70</sup> VALDIVIA Rosas, David. **Sistemas Expertos**.  
URL(<http://www.fortunecity.com/skycraper/romrow/207/se/Portada.html>)

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

- Manifestaciones Nuevas. Es decir, síntomas que no se habían observado con anterioridad.
- Causas Nuevas.
- Manifestaciones debidas a varias causas.
- Datos Inaccesibles, caros o de obtención peligrosa.
- Relaciones no biyectivas entre los datos y las causas.
- Fallos o averías de aparición intermitente.
- Existencia de varios fallos simultáneos o en cadena.

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

En el campo del mantenimiento predictivo los Sistemas Expertos se utilizan fundamentalmente como herramientas de diagnóstico. Se trata de que el programa pueda determinar en cada momento el estado de funcionamiento de sistemas complejos, anticipándose a los posibles incidentes que pudieran acontecer. Así, usando un modelo computacional del razonamiento de un experto humano, proporciona los mismos resultados que alcanzaría dicho experto.<sup>73</sup> Gráfico N° 2.14 muestra el funcionamiento de un SE de este tipo.

---

<sup>71</sup> SÁNCHEZ y Beltrán, Juan Pablo. *Op. cit.*

<sup>72</sup> *Ibid.*

<sup>73</sup> DE MIGUEL González, Luis Javier. *Op. cit.*

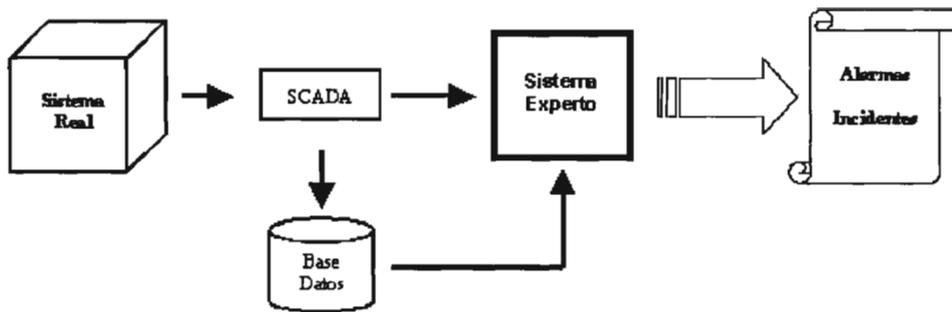


Gráfico N° 2.14 Funcionamiento de un SE usado en mantenimiento predictivo

### 2.9.2 Diseño y selección

El diseño se puede concebir de distintas formas:

- El diseño en ingeniería es el uso de principios científicos, información técnica e imaginación en la definición de una estructura mecánica, máquina o sistema que ejecute funciones específicas con el máximo de economía y eficiencia.
- El diseño industrial busca rectificar las omisiones de la ingeniería, es un intento consiente de traer forma y orden visual a la ingeniería de hardware donde la tecnología no provee estas características.
- Diseño es el proceso de especificar una descripción de un artefacto que satisface varias características desde un número de fuentes de conocimiento.<sup>74</sup>

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

Los SE prueban distintos diseños para verificar cuáles de ellos cumplen los requerimientos solicitados por el usuario, ésta técnica es llamada “generación y prueba”,

por lo tanto estos SE son llamados de selección. En áreas de aplicación, la prueba se termina cuando se encuentra la primer solución; sin embargo, existen problemas más complejos en los que el objetivo es encontrar la solución más óptima.

La utilidad de los programas basados en conocimientos aplicados a la ciencia y a la ingeniería no se limita al análisis sofisticado, muchos programas recientes han empezado a trabajar en la síntesis.<sup>75</sup>

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

### 2.9.3 Planificación

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

Los problemas que presentan la planificación mediante SE son los siguientes:

- Existen consecuencias no previsibles, de forma que hay que explorar y explicar varios planes.
- Existen muchas consideraciones que deben ser valoradas o incluirles un factor de peso.
- Suelen existir interacciones entre planes de subobjetivos diversos, por lo que deben elegirse soluciones de compromiso.
- Trabajo frecuente con incertidumbre, pues la mayoría de los datos con los que se trabaja son más o menos probables pero no seguros.
- Es necesario hacer uso de fuentes diversas tales como bases de datos.

---

<sup>74</sup> HOPGOOD, Adrian. **Knowledge – Based Systems for Engineers and Scientists.**

<sup>75</sup> WINSTON, Patrick Henry. **Artificial Intelligence.**

#### **2.9.4 Control**

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

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

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

#### **2.9.5 Reparación, corrección o terapia**

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

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

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

#### **2.9.6 Simulación, pronóstico o predicción**

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

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

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

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

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

Los sistemas de pronóstico deducen consecuencias posibles a partir de una situación. Su objetivo es determinar el curso del futuro en función de información sobre pasado y presente. Esto abarca diversos problemas, tales como predicciones meteorológicas, predicciones demográficas, o incluso previsiones de la evolución bursátil entre otros.<sup>76</sup>

### 2.9.7 Instrucción

---

<sup>76</sup> MARTÍNEZ De Ibarreta León, Francisco Javier. **Sistemas Expertos: Áreas de aplicación.**  
URL(<http://www.geocities.com/SiliconValley/Way/7788/SISEXP.HTM>)

---

Un sistema de instrucción (Sistema Experto para formación), realiza un seguimiento del proceso de aprendizaje de un estudiante. El sistema detecta errores de los estudiantes e identifica el remedio adecuado, es decir, desarrolla un plan de enseñanza para facilitar el proceso de aprendizaje y la corrección de errores.<sup>77</sup>

### **2.9.8 Recuperación de Información**

Los Sistemas Expertos, con su capacidad para combinar información y reglas de actuación, han sido vistos como una de las posibles soluciones al tratamiento y recuperación de información, no sólo documental. La década de 1980 fue prolija en investigación y publicaciones sobre experimentos de este orden, interés que continua en la actualidad.

Lo que diferencia a estos sistemas de un sistema tradicional de recuperación de información es que estos últimos sólo son capaces de recuperar lo que existe explícitamente, mientras que un Sistema Experto debe ser capaz de generar información no explícita razonando con los elementos que se le dan. Pero la capacidad de los SE en el ámbito de la recuperación de la información no se limita a la recuperación. Pueden utilizarse para ayudar al usuario, en selección de recursos de información, en filtrado de respuestas, etc Un SE puede actuar como un intermediario inteligente que guía y apoya el trabajo del usuario final. Para desempeñar de forma adecuada esta tarea, los enfoques centrados en la creación de modelos de las estructuras cognitivas del usuario son los más prometedores.<sup>78</sup>

### **2.10 Mantenimiento productivo total**

El mantenimiento productivo total (MPT) es un enfoque gerencial para el mantenimiento que se centra en la participación de todos los empleados de una

---

<sup>77</sup> *Ibid*

<sup>78</sup> TRAMULLAS y Kronos, Jesús. Recuperación de Información y Sistemas Expertos. URL(<http://www.tramullas.com/nautica/documatica/3-8.html>)

organización en la mejora del equipo. Este método se desarrolló en el sector manufacturero japonés, comenzando con la aplicación del mantenimiento preventivo al estilo norteamericano y europeo y avanzando hasta la aplicación de los conceptos de la administración de la calidad total y la manufactura justo a tiempo al campo del mantenimiento de los equipos.

El Instituto Japonés de Ingenieros de Planta definió el MPT en 1971 con cinco metas claves:

- Maximizar la eficacia global de equipo, que incluye disponibilidad, eficiencia en el proceso y calidad del activo.
- Aplicar un enfoque sistemático para la confiabilidad, la factibilidad del mantenimiento y los costos del ciclo de vida.
- Hacer participar a operaciones, administración de materiales, mantenimiento, ingeniería y administración en el control del equipo.
- Involucrar a todos los niveles gerenciales y a los trabajadores.
- Mejorar el rendimiento del equipo mediante actividades de grupos pequeños y el desempeño del equipo de trabajadores.

Los operadores del equipo son el punto central de las actividades del MPT. Aunque la mayoría de los operadores entienden lo que hace su equipo, pocos comprenden los mecanismos fundamentales sobre cómo funciona. El término “mantenimiento autónomo” se utiliza para describir las actividades de los operadores que se relacionan con el mantenimiento del equipo y con la naturaleza de estudio independiente de otras acciones de mejora del equipo. Los operadores realizarán tareas de limpieza, inspección, lubricación, ajustes, cambios de componentes menores y otras tareas de mantenimiento ligero que requieren cierta capacitación e instrucción, pero no destrezas completas de mantenimiento. El operador gradualmente aprende cómo diagnosticar los problemas del equipo antes de que se vuelvan serios.

En el MPT, siempre que un equipo funciona por debajo del nivel requerido, la pérdida de funcionamiento se registra y se monitorea. Estas pérdidas pueden agruparse en seis categorías: descomposturas, preparación y ajustes, trabajo en vacío y paros menores,

reducción en la velocidad, defectos y pérdidas de rendimiento. Las descomposturas y las preparaciones ocasionan tiempo muerto y producen un impacto en la disponibilidad; la reducción en la velocidad tiene un impacto en el tiempo del ciclo, y los defectos y pérdidas de rendimiento tienen un impacto en la calidad. La eficacia global del equipo, definida como el activo de disponibilidad, tiempo del ciclo y tasa de calidad, es la medida clave de eficacia del MPT. El operador y el trabajador de mantenimiento son capacitados para identificar problemas relacionados con la eficacia global del equipo y para realizar conjuntamente análisis de causas fundamentales para investigar las pérdidas.

Dentro del departamento de mantenimiento, la metodología del MPT fomenta el desarrollo de la planeación sistemática y el control del mantenimiento preventivo y correctivo, y apoya plenamente las actividades autónomas realizadas por el operador. En las plantas en donde el entorno de operación y mantenimiento ha sido mejorado hasta el punto de disminuir las devoluciones, se emprenden actividades para una prevención activa del mantenimiento, como se describió anteriormente en la sección sobre diseño para mantenibilidad. En todo momento deberá ponerse mucho énfasis en mejorar las habilidades del operador y del trabajador de mantenimiento. Los gastos en la capacitación son normalmente del orden del 5% al 8% del presupuesto para mano de obra.

### **2.10.1 Otorgamiento de facultades a los empleados**

No es necesario recalcar los beneficios de hacer participar a los operadores con el éxito del MPT. Una forma pragmática de lograr esto es empleando un método sistemático, basado en datos, para la transferencia de destrezas. La transferencia de destrezas el proceso de sacar las tareas que requieren destrezas bajas del dominio exclusivo de un grupo y llevarlas a una zona de tareas compartidas. Bajo esta política, un operador que haya sido capacitado y certificado adecuadamente puede realizar la tarea de un y viceversa. Esta sociedad entre la función de operaciones y la de mantenimiento tiene muchos beneficios que incluyen los siguientes:

- Los operadores y los mecánicos se convierten en personal con habilidades múltiples, lo cual conduce a un enriquecimiento del trabajo y a una mayor flexibilidad de los trabajadores.
- La participación de los operadores en el mantenimiento de rutina crea un sentido de responsabilidad, orgullo y propiedad.
- Los tiempos de demora se reducen y se incrementa la productividad.
- Se promueve el trabajo en equipo entre las funciones de operaciones y mantenimiento.

### **2.10.2 Administración del equipo**

El equipo es el punto focal del MPT. Este esfuerzo comienza identificando las pérdidas importantes del equipo. Como se indicó anteriormente, las siguientes seis pérdidas importantes del equipo:

- Fallas del equipo (descomposturas)
- Tiempo muerto por preparación y ajustes
- Trabajo en vacío y paros menores.
- Reducción de la velocidad.
- Defectos del proceso.
- Reducción del rendimiento.

La meta fundamental del MPT con respecto al equipo es aumentar su eficacia hasta su máximo potencial y mantenerlo en dicho nivel. Esto puede lograrse entendiendo las pérdidas anteriores y diseñando medios para eliminarlas.

### **2.10.3 Fallas del equipo.**

Las descomposturas representan un gran porcentaje de las perdidas totales. Deben hacerse todos los intentos por evitarlas. Para maximizar la eficacia del equipo, las

descomposturas deben reducirse a 0 cambiando la actitud de que las descomposturas son inevitables.

#### **2.10.4 Tiempo muerto por preparación y ajustes.**

Cuando termina la producción de un tipo dado de activo y el equipo se ajusta para estar listo a producir otro tipo de activo, se presentan pérdidas debido al tiempo muerto de preparación y a activos defectuosos. Estas pérdidas pueden disminuirse induciendo el tiempo de preparación. Muchas compañías están trabajando para lograr preparaciones de un solo minuto.

#### **2.10.5 Trabajo en vacío y paros menores.**

La producción puede verse interrumpida debido a un mal funcionamiento o a que una máquina esté ociosa entre la elaboración de dos activos. Las fuentes de estas pérdidas deben identificarse y eliminarse. La eliminación de los paros menores es una condición previa y esencial para la producción automatizada.

#### **2.10.6 Reducción de la Velocidad**

Estas pérdidas corresponden a la diferencia entre la velocidad de diseño del equipo y la velocidad de operación real la reducción de la velocidad puede deberse a problemas mecánicos ya una calidad defectuosa, o puede ser impuesta por el operador por temor de abusar del equipo. En otros casos, tal vez ni siquiera se conozca la velocidad óptima. El aumento de la velocidad da por resultado una mejora clara de la productividad si se identifican y se eliminan las razones de trabajar a una velocidad reducida.

#### **2.10.7 Defectos del proceso.**

Estos constituyen pérdidas en calidad ocasionadas por el proceso. Las condiciones que ocasionan el defecto deben identificarse y eliminarse.

### **2.10.8 Rendimiento reducido.**

Consiste en las pérdidas de arranques que ocurren durante las primeras etapas de la producción, desde el principio hasta su estabilización.

La mejora de la eficacia del equipo requiere la medición de estas pérdidas. Las seis pérdidas anteriores afectan la disponibilidad del equipo, su eficiencia y la calidad del activo de la siguiente forma:

La disponibilidad del equipo se ve afectada por la preparación y los ajustes y por las fallas del equipo.

La eficiencia del equipo se ve afectada por trabajar en vacío y paros menores y la reducción de la velocidad.

El rendimiento reducido y los defectos del proceso afectan la calidad del activo.

La eficacia global del equipo (OEE) está dada como sigue:

$$OEE = \text{Disponibilidad} \times \text{Eficiencia del desempeño} \times \text{Tasa de calidad} \dots \dots \dots (\text{Ecuación N}^\circ 14)$$

Donde

$$\text{Disponibilidad} = \text{Tiempo de carga} - \text{Tiempo muerto} \dots \dots \dots (\text{Ecuación N}^\circ 15)$$

#### **Tiempo de carga**

El tiempo de carga es el tiempo disponible menos el tiempo muerto planeado. El tiempo muerto planeado se refiere al tiempo muerto programado oficialmente en el plan de producción, como actividades de mantenimiento programado y administración.

Eficiencia de desempeño = Tiempo teórico del ciclo x Cantidad procesada. (Ecuación N° 16)

### **Tiempo de Operación**

Tasa de calidad = Cantidad procesada - Cantidad defectuosa..... (Ecuación N° 17)

### **Cantidad procesada**

La eficacia global del equipo puede reducirse a lo siguiente:

OEE = Tiempo teórico del ciclo x Cantidad procesada x Tasa de calidad.... (Ecuación N° 18)

### **Tiempo de carga**

El tiempo teórico del ciclo y el tiempo de carga son constantes por día. Por lo tanto la OEE está relacionada directamente con el número de artículos de buena calidad. La OEE puede mejorarse incrementando la disponibilidad, la eficiencia en el desempeño y, más importante, la tasa de calidad. Puede verse que la disponibilidad puede mejorarse reduciendo el tiempo muerto, y que la eficiencia en el desempeño puede mejorarse reduciendo el tiempo del ciclo.

Para mejorar la tasa de calidad, deberá identificarse el estado en el cual el equipo produce artículos de alta calidad. A continuación deberá establecerse una política de mantenimiento para conservar el equipo en dicho estado. Por lo tanto, la meta del mantenimiento deberá ser conservar el equipo en un estado en el que se produzcan cero defectos.

### **2.10.9 Implantación del MPT**

Nakajima [4] sugiere los siguientes doce pasos para implantar el MPT:

- Anunciar la decisión de la alta dirección de introducir el MPT.
- Lavar una campaña educativa para introducir el MPT.
- Crear organizaciones para promover al MPT.
- Establecer políticas básicas de MPT.
- Formular un plan maestro para el desarrollo del MPT.
- Mantener el impulso del MPT.
- Mejorar la eficacia de cada equipo.
- Desarrollar un programa autónomo de mantenimiento.
- Desarrollar un programa de mantenimiento programado para el departamento de mantenimiento.
- Llevar a cabo una capacitación para mejorar la destreza en operaciones y mantenimiento.
- Desarrollar un programa eficaz de administración.
- Perfeccionar la implantación del MPT y elevar sus niveles.

### **2.11 Definición de términos básicos**

- **Confiabilidad:** Probabilidad de que un equipo cumpla su misión específica (no falle) bajo condiciones de operación determinadas en un período de tiempo determinado.
- **Costos de mantenimiento:** Refleja los costos estimados promediados por año, cuando se presenta la falla.
- **Criticidad:** Característica de un equipo que representa el impacto de la falla de este sobre: el ambiente, la seguridad y la producción del sistema al cual pertenece.
- **Diagrama de instrumentación y tuberías (DTI):** Proveen información mecánica referente a un proceso o equipo determinado, mas detallada que en lo representado en un diagrama de flujo. Su equivalente en ingles es P&ID.
- **Equipos críticos:** Son aquellos que al fallar, producen un impacto en: producción, seguridad de personas y bienes, daños al ambiente.
- **Equipos estáticos:** Son dispositivos empleados para el filtrado, almacenaje y tratado químico de los fluidos que ellos contengan.

- Equipos no críticos: Son aquellos que al fallar, no producen ningún impacto.
- Equipos rotativos: Son máquinas o dispositivos empleados para elevar, transferir o mover fluidos.
- Falla: Es la ocurrencia no previsible inherente al equipo y que impide que este cumpla su función.
- Flexibilidad operacional: Se basa en las operaciones que existen, para seguir realizando la función del equipo al producirse la falla.
- Frecuencia de fallas: Es un índice de la tasa de la falla que presenta el equipo o sistema en estudio.
- Intercambiadores aniónicos: Son equipos que se encargan de intercambiar los iones negativos que posee el agua con los iones negativos que posee la resina aniónica (OH<sup>-</sup>). Entre sus partes más importantes se encuentran: distribuidor, strainers, válvulas y la resina aniónica.
- Intercambiadores catiónicos: Son equipos que se encargan de intercambiar los iones positivos contenidos en el agua, con los iones positivos que posee la resina catiónica (H<sup>+</sup>). Entre sus partes más importantes se encuentran: distribuidor, strainers, válvulas y la resina catiónica.
- Impacto a la seguridad, higiene y ambiente (SHA): Constituyen los efectos o daños causados a la seguridad o al ambiente.
- Impacto operacional: Indica la repercusión de la falla en el sistema o equipo, sobre la operación o rendimiento de la planta.
- TFS: Tiempo fuera de servicio tomado en días.
- TO: Tiempo en operación tomado en días.
- TPEF: Tiempo promedio entre falla.
- Trenes Desmineralizadores: Los trenes desmineralizadores están compuesto por un intercambiador catiónico y un intercambiador aniónico.