ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD

"DISEÑO E IMPLEMENTACION DE UN SISTEMA EXPERTO PARA SELECCIONAR PERSONAL"

TESIS DE GRADO

Previa a la Obtención del Título de

INGENIERO EM COMPUTACION

PRESENTADA POR:

RICARDO BAQUERIZO CORNEJO

Guayaquil - Ecuador

1.993

AGRADECIMIENTO: Mi más sincero agradecimiento a mi director de tesis Ing. Sixto García A., quien colaboró generosamente con excelente disposición en la realización de este proyecto.

También agradezco a la Ing. Lucila Pérez C. con su ayuda en los momentos iniciales, y luego con su generosa colaboración por medio del material bibliográfico ayudó definitivamente a la realización de esta Tesis de Grado.

Por último quiero agradecer a la Ing. Dusya Vera B. quien colaboró en las pinceladas finales de este trabajo.

DEDICATORIA:

A DIOS por la vida la posibilidad de estudiar, trabajar y ser feliz.

A mi esposa quien con su sacrificio constante me empujó y dio ánimo para dedicar cada hora y minuto a este proyecto. Quien con su sonrisa me alegra cada día.

A mis padres a quienes quiero tanto, y quienes me dieron tanto, nunca podré devolver todo el bien que han hecho por mi.

DECLARACION EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en esta Tesis, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral"

(Reglamento de Exámenes y Títulos Profesionales)

R Baquerizo C.

Ricardo Baquerizo Cornejo

Subdecano de la Facultad de
Ingeniería en Electricidad
Ing. Armando Altamirano

Director de Tesis
Ing. Sixto Garcia A.

Miembro del Tribunal
Ing. Jaime Puente P.

Miembro del Tribunal
Ing. Guido Caicedo

INDICE

INTRODUCCION

La documentación que encontraremos a continuación quiere explicar aunque sea someramente una de las técnicas más interesantes que está desarrollando la tecnología. Es quizás la que más atrae a la gente de la calle. Alrededar de la cual se han desarrollado muchas películas y novelas de ciencia ficción. Vamos a hablar de la **Inteligencia Artificial** y de los **Sistemas Expertos.**

Durante siglos la gente se ha preguntado si habrá inteligencia más allá de nuestro planeta, donde el hombre no es capaz de dominar, de conocer, de mirar. A partir de la invención de las computadoras, los científicos se han empeñado en fabricar **esta** inteligencia de la que hablábamos antes, y no sólo buscarla en el exterior.

Actualmente una parte de la **ciencia** de la computación se dedica exclusivamente a intentar fabricar inteligencia donde no hay inteligencia, se intenta reemplazar la capacidad humana por un alqo compuesto básicamente de silicio y conectado a un tomacorriente o una batería para darle "vida" **a** lo que no tiene vida, y para darle "inteligencia" a lo que nu tiene inteligencia.

Pero dejemos de soñar un momento y concentrémonos en la realidad actual, y conocida. En este momento existen sistemas muy avanzados denominados "Sistemas Expertos", se los utiliza con éxito en algunas ramas de la ciencia, como son la medicina, la geología, la química, la agricultura, la economía, etc.

De sistemas computarizados como estos dependen muchas veces nuestra salud, el alimento de los pueblos y la capacidad monetaria de cada uno de nosotros.

Hemos hecho incapié en la importancia del tema que trataremos a continuación, para intentar meter al lector en este mundo maravilloso y aún poco conocido que es la inteligencia artificial.

La documentación cuya introducción estamos leyendo intentará lograr dos propósitos:

A. Documentar los avances en las técnicas de la Inteligencia Artificial en la construcción de Sistemas Expertos. Además explicar las aplicaciones de los sistemas expertos en el mundo moderno.

b. Desarrollar un prototipo de Sistema Experto capaz de servir de herramienta para la selección de personal en las empresas.

El segundo propósito ha sido desarrollar un "prototipo" de Sistema Experto y no Sistema Experto, no porque queramos ser conservadores a pesimistas, sino porque para desarrollar un Sistema Experto se necesitan varios años de trabajo, un equipo de gente entrenada, y mucho dinero. Es lo que se interpreta de la siguiente cita textual.

> Un sistema en producción sobre micros realizado por un buen equipo a partir de una herramienta de desarrollo, con una base de conocimientos de tipo medio (del orden de 500 reglas) y sobre un problema estándar, deberá comprarse (o venderse) por un precio entre US$50.000 y US$100.000 todo incluido (sensibilización, realización y formación).[1]

> De todas formas, la constitución de un equipo <<sistema experto>> permanente sólo se justifica, cuando se desea introducir masivamente esta tecnología en la empresa, ya que por

[1]Benchimol-Levine-Pomerol, "Los Sistemas Expertos en la Empresa", Macrobit. 1990. Página 184.

término medio se tardaría de tres a cuatro años.[2]

En los sucesivos capítulos iremos avanzando paso a paso para completar los dos objetivos fijados.

El capítulo primero explica la historia de los sistemas expertos, el concepto y sus fundamentos.

El capítulo segundo trata de las aplicaciones actuales y las que están en desarrollo para el uso de sistemas expertos en las diferentes áreas de la ciencia:

-Medicina

-Botánica

-Química

-Finanzas

-etc.

En el tercer capítulo se trata en forma más específica las herramientas actuales para el desarrollo de sistemas expertos, tales como lenguajes tradicionales, lenguajes de

---

[2]Benchimol-Levine-Pomerol, "Los Sistemas Expertos en la Empresa", Macrobit. 1990. Página 185.

inteligencia artificial, y conchas (shell) para sistemas expertos.

En el capítulo cuarto estudiaremos el análisis y diseño del Sistema Experto Para Seleccionar Personal "SESEP".

En el capítulo quinto veremos como trabaja el Sistema Experto Para Seleccianar Personal "SESEP", el modo de correr el programa, las pruebas realizadas, las ventajas que da, y su aplicación en las empresas.

La conclusión recoge los puntos más destacables e importantes de este trabajo de investigación.

Y por último en el apéndice encontraremos el respaldo de los casos utilizados para probar el Prototipo de Sistema Experto para Seleccionar Personal.

CAPITULO I

"SISTEMAS EXPERTOS"

1.1 BREVE HISTORIA.-

A partir de la segunda guerra mundial, y con el desarrollo de las computadoras se empezó a investigar la posibilidad de desarrollar computadores que utilicen la inteligencia al iqual que lus seres humanos.

Es así como en 1943 S. Warren, MC Culloch y Walter Pitts trabajaron can **Redes** Neuronales. Las Redes Neuronales eran un modelo electrónico que intentaba imitar a1 cerebra humano. Se quiso crear una máquina que funcionase al igual que el **cerebra** humano y así lograr tener una máquina can inteligencia.

En 1957 se desarr-allo en la Universidad de Cornell, el PERCEPTRON, **esta** máquina utilizaba Redes Neuronales como filosofía de su funcionamiento.

El PERCEPTRON y las Redes Neuronales perdieron gran parte de su impartancia cuando Marvin Minsky y Seymaur Papert escribieron un 1ibro demostrando que máquinas

basadas en Redes Neuronales no eran útiles para resolver problemas complejos. Ya que tenían una capacidad muy reducida.

Al hablar de Redes Neuronales hay que tener presente que tratan de imitar la estructura neuronal del cerebra, y el cerebra tiene cerca de 10 billones de neuronas y entre 1.000 y 10.000 enlaces entre cada neurona y las que la rodean. Coma sabemos la tecnología na estaba la suficientemente desarro1lada como para alcanzar una densidad tan alta de circuitos electrónicos. Gracias al desarrol la acelerado de la tecnologia, es posible que en los actuales años 90 se vuelva a intentar trabajar con Redes Neuronales.

Entre los años 1960 y 1970 se sentaron los principios básicos para las organizaciones en árbol para resolver problemas. Las investiqaciones se centraran en problemas fáciles de escribir pero muy complejos de resolver-, como por ejemplo el juego de ajedrez a la demostración de teoremas matemáticos.

Quedaron de lado las investigaciones dande las definiciones no fuesen precisas. Es decir dande hubiese

que hacer aproximaciones, sacar probabilidades, a simplemente donde no se tengan cubiertas todas las reglas.

En los años 60 como hemos dicho las investigaciones se centraron en sistemas can un entarna de fácil definición. Con la excepción de DENDRAL el cual apareció a mediados de los años 60. Este sistema sirve para determinar estructuras moleculares, tomando la información de un espectógrafo de masas. DENDRAL fue desarrollada en LISP y FORTRAN.

El más conocido Sistema Experto fue el MYCIN, desarrollado por la universidad de Standford entre 1972 y 1976. El MYCIN es un sistema que utiliza el conocimiento inexacto para el diagnóstico de enfermedades de la sangre. La primera versión de MYCIN fue desarrollada en LISP.

Poco tiempo después del MYCIN nació el PROSPECTOR. El PROSPECTOR es un sistema experto de consulta para ayudar a los geólogos en la exploración de minerales.

En 1959 nació el lenguaje de inteligencia artificial LISP. El LISP maneja átomos y listas, y es sumamente modular.

A partir de LISP nacieron algunos lenguajes, en 1971 MICRO-PLANNER, en 1972 PLANNER, en el mismo año CONNIVER. En 1976 nace SMALLTALK-72 el cual es el primer lenguaje orientado a objetos.

Entre los años 1970 y 1975 fue desarrollado el PROLOG, lenguaje basado en la lógica de predicados. Fue desarrollado por Colmereaur y Roussel.

Al especializarse más y más los lenguajes de inteligencia artificial se llegó a desarrallar las conchas (shell), las cuales son un entorno de sistema experto. A las conchas basta ingresarles la información, reglas y base de conocimientos, para obtener un sistema experto funcionando.

Sin embargo la inteligencia artificial está aún en proceso de investigación y queda mucho por hacer.

De acuerdo a Peter Jackson en su libro "Introduction to Expert Systems" debemos ser cautelosos al hablar de la tecnología de los Sistemas Expertos. Ya que las técnicas aún están en vías de desarrollo, y a pesar de *eso* hay una tendencia muy grande a comercializarlas. Hay muchas técnicas ya comercializadas para las cuales aún nu se ha

prabado su eficacia.

1.2 DEFINICIONES, CARACTERISTICAS Y TECNICAS.-

1.2.1 ADQUISICION DE CONOCIMIENTOS.-

Uno de los tópicos más importantes al pensar en un Sistema Experto es el que busca lograr que el Sistema Experto sea capaz de adquirir canucimientns.

En el mundo actual las situaciones son muy diferentes después de un periado de tiempo pequeño. Esto se debe al desarrollo técnico tan veloz que se ha venida dando en los últimos 50 años. Digo esto porque enfrentar un problema un dia puede tener matices muy variadas al dia siguiente, y no se diga un mes después. Es pues imprescindihle que si queremos crear una máquina o un sistema capaz de resolver situaciones, será necesario que dicho sistema también sea capaz de aprender de las experiencias pasadas y/o coma mínimo que sea capaz de ser alimentado con nueva información en el momento que el usuario lo considere necesario.

Esta adquisición de cnnucimientas se puede dar pur procesos tan variados como los siguientes:

- Que el usuario tenga que de alguna forma modificar el código del sistema
- Que el usuario modifique la base de datos del sistema
- Que el sistema pregunte al usuario, cuando encuentre que no tiene la información suficiente, y que la nueva información sirva de experiencia para que el sistema la pueda utilizar en futuros requerimientos

Por supuesto la primera opción nombrada es la más básica y presenta pocas facilidades por cuanto no todo usuario tiene, y no es correcto pedírselo, el entrenamiento en la parte técnica del código del Sistema Experto

Las opciones segunda y tercera prestan claras facilidades al usuario, sobre todo la tercera, la cual da realmente la idea de estar trabajando con un sistema inteligente.

Para empezar a trabajar con un Sistema Experto experto, lo normal es empezar par ingresar la información básica en la base de conocimientos. Esta infnrmación normalmente se la consigue de un "experto". Para obtenerla, se debe especificar en reglas o ejemplos el

pensamiento de un experto. Lo usual es plantear al experto casos cancretas que sirvan para abtener un patrón estándar de resolución de problemas.

La ventaja más impartante que tiene un sistema capaz de adquirir conucimientos, es que toma poco tiempo enseñarle, en relación al tiempo que toma preparar a una persona hasta el punto de ser un "experto".

Pensemos en los años de escuela, colegio, universidad, postgrado, años de prácticas y experimentas, etc, 1.0 cual no toma menus de 15 años. Es más los seres humanos somos tan frágiles que podemos morir en cualquier momento durante o después de esta preparación. Un Sistema Experto puede tardar uno a das años en adquirir el conocimiento, y luego no lo pierde, no desaparece y no se enferma. Además se lo puede duplicar e instalar para trabajar en cualquier lugar del mundo, al mismo tiempo. Está de más decir que esto es imposible para un ser humano.

1.2.2 FIABILIDAD.-

Conocer la fiabilidad de un Sistema Experto es realmente más sencillo que conocer la categoría de un "experto".

Como decíamos antes, un "experto" tarda muchos años en llegar a serlo. Lo cual no implica que después de estos años sea reconocido como tal. Pues hace falta otro argumento que es la fama o la imagen que muestra, y con la cual se lo llega a reconocer. Wn "experto" humano tiene que demostrar que lo es en casos reales de la vida, dedica parte de su tiempo a conferencias, charlas, experimentos o artículos, debe demostrar que es brillante.

Para probar la fiabilidad de un Sistema Experto, lo que hay que hacer es probar las respuestas contra las respuestas de un "experto", y probarlo en casos reales que ya hayan ocurrido. Un Sistema Experto tiene además que demostrar como llegó a la respuesta. Esta demostración la puede hacer mostrando el algoritmo utilizado, en el caso de problemas matemáticos, o mostrando pasa a paso el proceso, para el caso de problemas con objetos de símbolos.

1.2.3 DOMINIO DEL CONOCIMIENTO.-

¿ Qué diferencia encontramos entre un experto humano y un Sistema Experto en cuanto a su capacidad de contener o dominar el conocimiento ?

La diferencia se encuentra básicamente en que ningún

ser humano es capaz de aumentar su velocidad de razunamiento, ni su capacidad de memoria, ni es capaz de permanecer por siempre. Ningún ser humano puede transmitir ni. heredar a otro su conocimiento.

Un Sistema Experto depende únicamente del desarrollo tecnológico para dar respuestas en menor tiempo, para aumentar su capacidad de memoria. Un Sistema Experto puede morir tecnológicamente, porque uno mejor lo reemplaza, pero el conocimiento puede ser transmitido, manipulado y almacenado sin ningún problema, de tal forma que el nuevo Sistema Experto 1o utilice.

A medida que pasen los años y a medida que sea necesario se podrá lograr que un sólo Sistema Experto resuelva problemas de más de un Area del saber. Se podrá integrar dos Sistemas Expertos en tópicos diferentes, para resolver problemas más complejos. O un nuevo Sistema Experto podrá utilizar el conocimiento adquirido por otros, para complementar e1 conocimiento y so1ucionar problemas.

1.2.4 RESOLUCION DE PROBLEMAS.-

Para resolver un problema un "experto" necesita conocer las reglas bajo las cuales se rige el problema,

además necesita tener una inteligencia y una forma de razonar específica dentro del campo al cual se circunscribe el mismo.

Un Sistema Experto funciona parecido a un "experto". El Sistema Experto necesita conocer las reglas o leyes de la situación que va a resolver y el conocimiento sobre ese tópico. También necesita tener una forma de razonar, lo que se denomina motor de inferencia. Este motor de inferencia sirve para encontrar la solución por el camino de la razón.

Si no existiera el mntor de inferencia probablemente se le pediría al computador que revise todas las situaciones posibles del problema y que luego le aplique las reglas y el conocimiento para eliminar las situaciones incorrectas.

El motor de inferencia hace que a la var que se buscan las soluciones posibles, se apliquen las reglas y el conocimiento necesario para eliminar durante el razonamiento las soluciones no válidas.

1.3 COMO TRFIEAJA UN SISTEMFI EXPERTO.-

Un Sistema Experto es el resultado de una combinación elementos tales como: interfaz del usuario, base de conocimientos y motor de inferencia.

La interfaz del usuario es la herramienta que permite al usuario comunicarse con el Sistema Experto, es la interfaz la que permite al usuario ingresar los conocimientos al Sistema Experto, permite modificar el conocimiento y modificar reglas. La interfaz del usuario permite además que el sistema muestre los resultados. El sistema puede pedir más información si le es necesaria.

La interfaz puede ser compleja con iconos, ratón , etc, o muy sencilla hasta el nivel de sólo interactuar con el usuario a nivel. de línea de usuario o "prompt".

La base de conocimientos es una base de datos donde se encuentra la información organizada para que el sistema puede utilizarla. La base de conocimientos puede tener una estructura muy compleja con varias niveles de profundidad que relacionen los datos, pero también puede estar formada por datos en un sólo nivel.

El motor de inferencia es el cerebro del Sistema Experto, es quien decide como trabajar. El motor de inferencia busca la solución a un problema tomando la información que necesite de la base de conocimientos. El motor de inferencia crea el camino a seguir de acuerdo a la información que le proporcione la base de conocimientos, y de acuerdo a los datos que se tiene sobre el problema.
El motor de inferencia puede tener uno de los siguientes tipos de control:

- exploración en profundidad
- exploración en anchura
- exploración heurística

Partamos de que por tener unos datos iniciales, y unas reglas, existen varios caminos a seguir. Si ordenamos en forma de árbol los caminos tendremos como posición inicial, la situación actual, como segmentos las reglas que apliquemos, los nodos serán las situaciones intermedias.

Los extremos del árbol son el punto hasta donde nos permite avanzar el conocimiento actual.

A continuación plantearemos un ejemplo tomado del libro "Los Sistemas Expertos en la Empresa" página 32.

Este ejemplo es miry ilustrativo y servirá para explicar en forma sencilla La exploración en profundidad y la exploración en anchura.

Ejemp1a:

Tenemos cinco pasibilidades de inversión A=200(7%), B=300(6%), C=400(6%, D=500(4.5%) y E=800(5%. E1 primer número indica la cantidad total que se puede invertir (en miles de ECUS) y el segundo el porcentaje de rendimiento anual de la inversión.

Dispongo de un millón de ECUS, ¿ qué tipo de inversiones debu hacer ? Los elementos del problema son la inversiones posibles y lo que reportan anualmente, por ejemplo la inversión A produce (en miles de ECUS) 200*7%=14, B produce 300*6%=18, mientras que si hago la inversión A+B obtendría 32 (miles de Ecus). vay a introducir en la base de datos pares de la forma (A;14), (A+B;32), (A+E;54), etc. Estos pares caracterizan cada inversión y su rendimiento anual.

Las reglas de producción son:

Si 1a inversión total no es superior a 1'000.000 (Ecus). Entonces añadir una inversión tal que la suma

total de las inversiones **sea** inferior o igual a 1'000.000 (Ecus).

Inicialmente mi base **de** datos contiene (V;0) siendo V la decisión de no invertir. Los hechos (A;14), (B;18), (C;24), (D;22,5) y (E;40) aparecerán enseguida.

Notemos que la inversión A+B es la misma que B+A, y convenimos que una inversión sólo puede realizarse una vez en un sentido. Coma consecuencia las sumas A+B, A+B+C, A+D,... sólo se consideran siguiendo un orden alfabético.

En este problema cuyo objetivo no es un hecho conocido no nos queda otra solución que desarrollar por completo el árbol de búsqueda, tal coma se ve en **la** figura **1.1.**

La solución al problema se hace evidente, se eligen las inversiunes A+B+C.

Lo que nos interesa ahora es describir ciertas estrategias de control para el desarrollo de este árbol. Partiendo de (V;0) puede añadir (A;14) y después elegir una regla para llegar a (A+B;32) después (A+B+C;52). Es decir que para un nodo n_1 elijo **para** aplicar una regla que tenga

como sucesor a n_1. La elección sabre las reglas aplicables es arbitraria a priori. Si elijo, por ejemplo, el orden alfabético de las reglas aplicables, obtengo el desarrollo (V;0), (A;14), (A+B;32), (A+B+C;56) he llegado a una hoja terminal, remonto al nodo anterior y aplico alguna de las reglas que todavía no he aplicado y sigo subiendo todo cuanto sea necesario.

Tal búsqueda se llama búsqueda en profundidad a exploracibn vertical (depth-first). Hemos vista en la figura 1.2 el orden en que van encontrándose los nodos en la búsqueda en profundidad, con la elección alfabética de las reglas.

Si el número de nodos es muy grande, una buena idea es: fijar a priori una profundidad y no traspasarla bajo ninguna condición.

La búsqueda en anchura o exploracibn horizontal (breadth-first) significa que en cada nodo activamos todas las reglas aplicándolas a este nado, después elegimos atra nado de la misma prafundidad y repetimos la misma operación nuevamente. La figura 1.3 muestra el orden de creación de

los nodos en la búsqueda en profundidad, donde se eligen tanto las reglas como los nodos -dentro de la misma profundidad- par orden alfabético.

A menudo en la solución de un problema se determina un objetivo que es la solución del problema con que estamos enfrentados.

Podríamos establecer en objetivo aquí: por ejemplo <<obtener un rendimiento anual mayor o igual a 50 (miles de Ecus)>>. Detendremos la búsqueda a través del árbol cuando alcancemos un hecho (X;Y) con Y igual o mayor de 50. En este caso el método de explorar el árbol es significativo. Si lo exploramas en profundidad (figura 1.2) encontraremos la inversión (A+B+C;56) en tres objetivos. Si la exploramos en anchura (figura 1.3) encontraremos la inversión (A+E;54) en nueve objetivos.

Además de las búsquedas en profundidad y en anchura existe la búsqueda heurística. Como hemos visto, al hacer la búsqueda en profundidad o anchura se utiliza un algoritmo ciego, porque no nos fijamos en los datos para en base a ellos escoger el mejor camino. Una búsqueda heurística se caracteriza por evaluar la información que se

obtiene en cada nuevo nodo. En base a esta información, se decide el sucesor de cual nodo tomar.

Una función heurística tipica es en la que se activan primero todas las reglas en anchura, luego se toma el sucesor del nodo cuyo valor sea el mayor. A partir de este punto se repite el proceso.

Una búsqueda heurística no asegura que se alcance la mejor respuesta, pero intenta obtener un compromiso entre la calidad de la respuesta y el tiempo tomado en obtenerla.

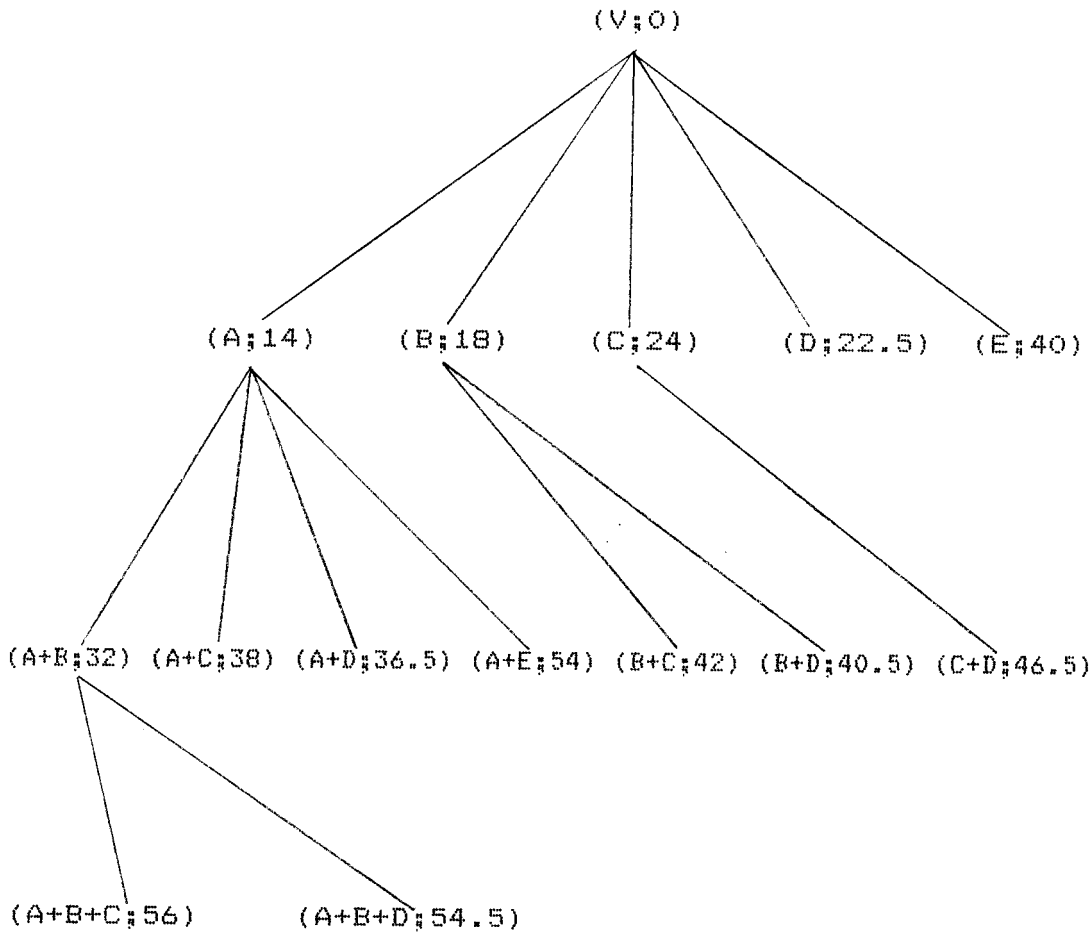


Figura 1.1 Arbol completo del problema de la inversión

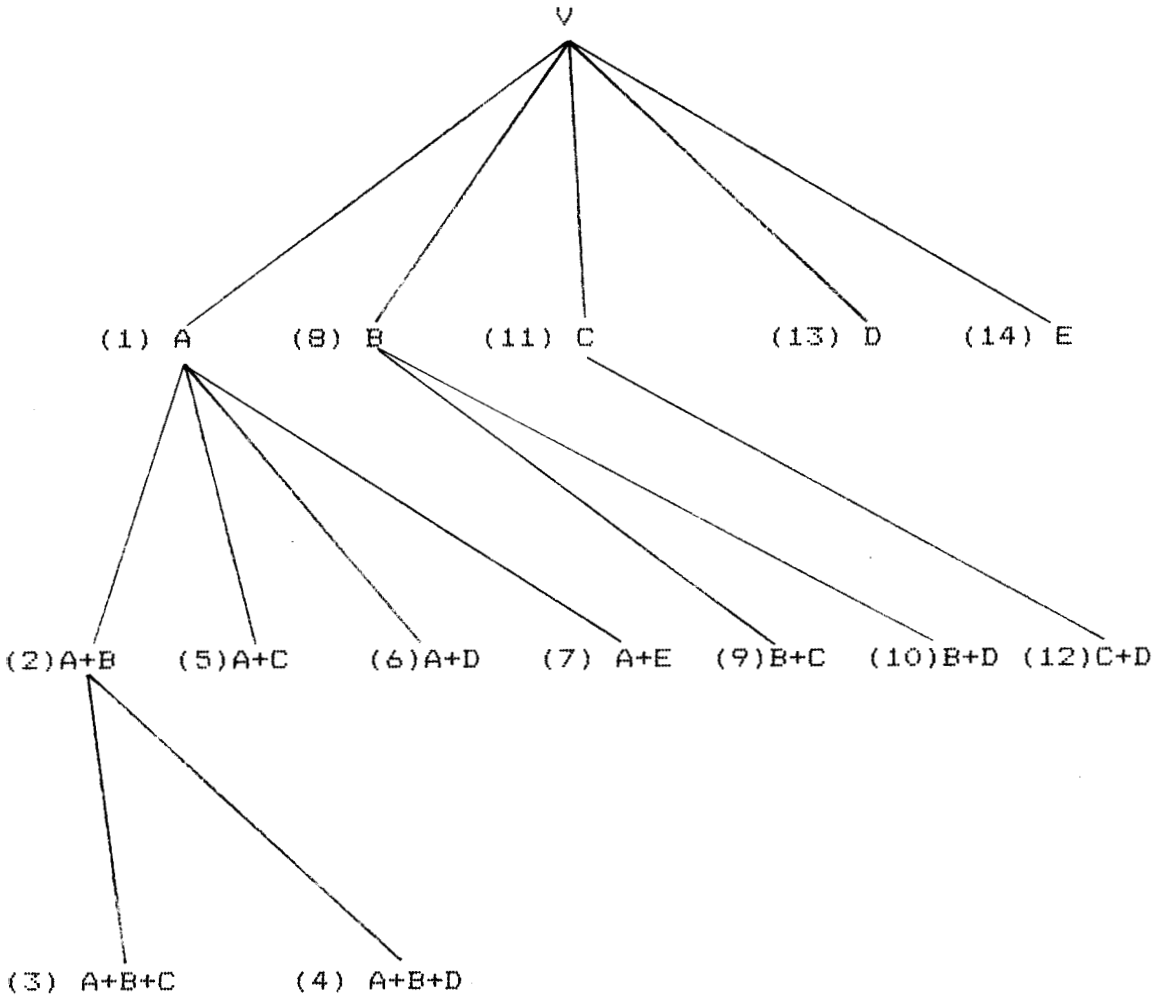


Figura **1.2** Búsqueda en profundidad

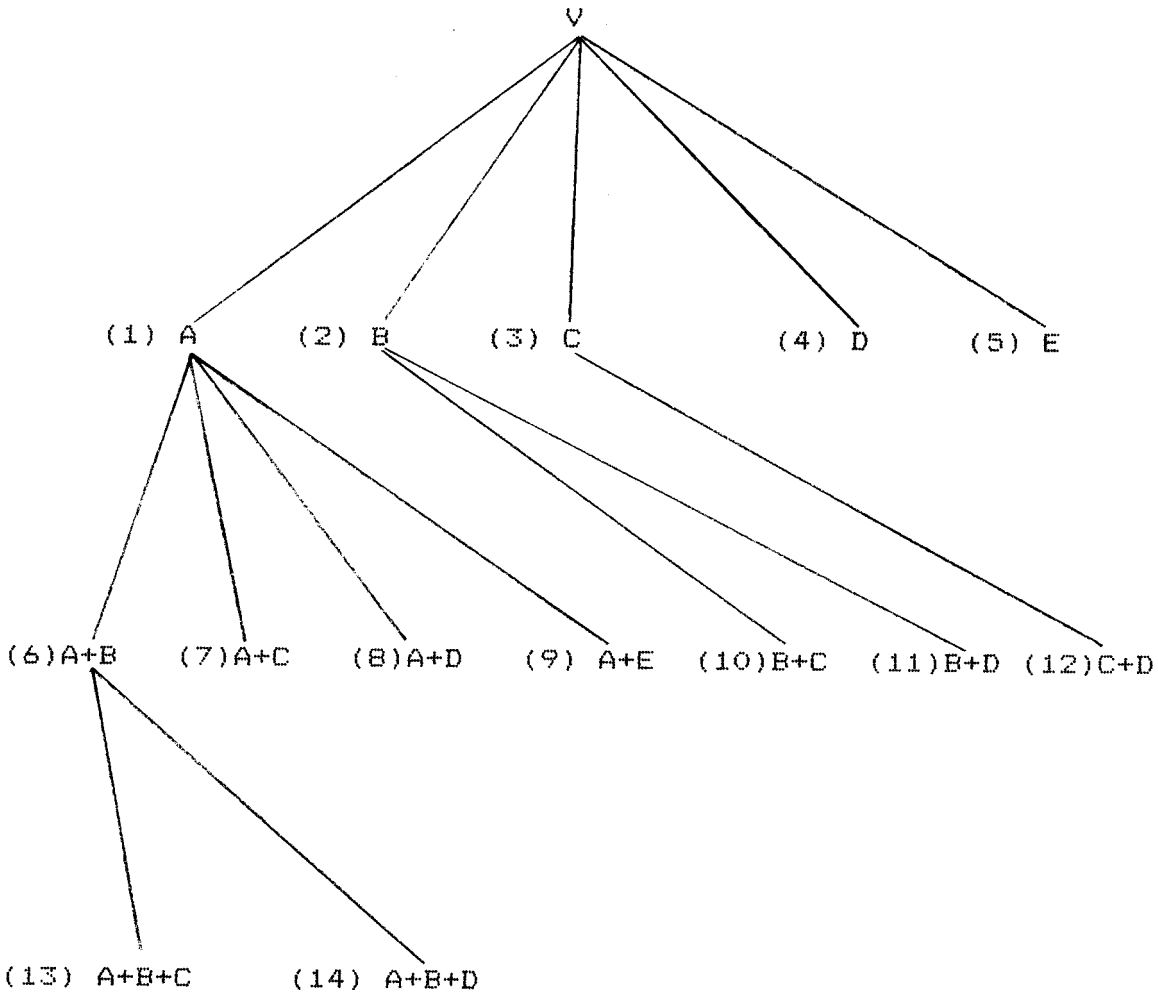


Figura **1.3** Búsqueda en anchura

CAPITULO II

"APLICACION DE SISTEMAS EXPERTOS"

2.1 INTERPRETACION.-

Los Sistemas Expertos tienen entre sus aplicaciones la de interpretar datos o signos o salidas de sensores. A partir de los datos obtenidos, el Sistema Experto debe sacar conclusiones de lo que estd ocurriendo. La interpretación puede ser sencilla si todos los datos convergen a un mismo punto.

Ejemplo:

Premisas:

Son las 23h00

El cielo estd oscuro

Conclusión:

Es de noche

Sin embargo, los datos pueden parerer contradictorios en ciertas ocasianes.

Ejemplo:

Premisas:

Son las 19h00

El cielo está claro

Conclusión:

Los datos provienen de los polos

Estos ejemplos básicos nos demuestran la necesidad de que el Sistema Experto esté dotado de un conocimiento completo que abarque todas las posibilidades de resultados. Además el Sistema Experto deberá ser capaz de resolver situaciones en las que existan datos contradictorios. Para hacer esto deberá utilizar cálculos de medias, ponderaciones, deberá en ocasiones despreciar valores, y por último deberá respander que no existe interpretación alguna.

PROSPECTOR es ejemplo

2.2 DIAGNOSTICO.-

En muchas ocasiones conocemos una serie de situaciones con respecto a algo, pero no conocemos que causa estas situaciones.

Por ejemplo:

Nuestro auto simplemente no enciende.

o sino

Nuestro auto se apaga constantemente sin razón aparente.

El diagnóstico identifica las causas internas que provocan el problema.

Típicamente se utiliza un **Sistema** Experto que interpreta los datos del problema antes de utilizar el Sistema Experto que hace el diagnóstico. Este trabajo en conjunto lo pueden realizar Sistemas Expertos independientes a un Sistema Experto que contenga incorporada un segundo Sistema Experto que sea el interpretador.

Los Sistemas Expertos de diagnóstico son muy complicados por cuanto se encuentran a menudo frente a varios problemas.
De acuerdo a J.P. Sánchez y Beltrán los problemas que ocurren son los siguientes:

- síntomas que no estaban incorporados al **Sistema** Experto por que no existían con anterioridad

- causas que no se habían dado antes

- síntomas que se deben a varias causas

- datos inasequibles o caros

- distintas causas que presenten las mismos síntomas o una causa que presente varios síntomas

- fallas intermitentes

- síntomas contradictorios par la existencia de varias fallas

El Sistema Experto más destacado como Sistema Experto de diagnóstico es el **MYCIN.** Nos detendremos a explicar aunque en forma superficial al MYCIN, ya que este es un ejemplo representativa de Sistema Experto.

Una de las ramas de **la** ciencia **que** más muestra **la** necesidad de la utilización de Sistema Experto es **la** medicina. Esto se debe a que la información médica a aumentado en gran proporción en las últimas décadas. Prueba de esto es que cada vez los médicos tienden a dedicarse más y más a materias muy específicas. Además, **la** distribución de los médicos expertos alrededar del munda no es proporcionada a las necesidades, y por último can **la** tendencia del mundo a dar servicios (cuidados médicos) de

mejor calidad, la demanda de tiempo de los expertos a aumentado.

Un ejemplo de Sistema Experto aplicado a la medicina es MYCIN. MYCIN es un Sistema Experto que se dedica a diagnosticar enfermedades de sangre.

La medicina para el tratamiento de infecciones son los antibióticos, los cuales deben ser administrados con inmenso cuidado. Los antibióticos mal aplicadas pueden matar no sólo las bacterias que producen la infección, sino también los organismas de los cuales dependemos para vivir, como anticuerpos, glóbulos rojos, etc.

La utilización del antibiótico incarrecto puede no curar en absoluto la infección. Ya que no existe ningún antibiótico que mate a todas las bacterias. Es más las bacterias pueden crear resistencia a algunos antibióticos para los que antes fueron sensibles.

La selección de la terapia a seguir para el caso de infecciones debe cumplir los siguientes pasos:

A. Decidir si el paciente tiene una infección significante.

B. Determinar el organismo causante de la infección.

C. Seleccionar un grupo de drogas apropiadas para combatir la infección.

D. Escoger fa droga más apropiada o una combinación de ellas.

MYCIN fue desarrollado para ayudar a doctores que no son expertos en el campo de los antibióticos, para que puedan diagnosticar y recetar el tratamiento de infecciones de sangre.

MYCIN es un ejemplo clásico de Sistema Experto porque es un sistema típico para ofrecer diagnósticos y facilitar información.

Además MYCIN satisface un amplio espectro de características que 1o garantizan coma ejemplo representativo; funciona en un ambiente absolutamente real. y satisface una necesidad real, es confiable y entrega respuestas calidad. De MYCIN han aparecido una serie de sistemas adicionales que sirven para entregarle información o para hacer algún proceso adicional, además a partir de MYCIN se ha desarrollado Sistemas Expertos en otras áreas

del conocimiento.

Consta de tres par-tes principales:

A. El sistema de consultas hace preguntas, saca conclusiones y entrega información.

B. El sistema de explicación, el cual responde preguntas y justifica la información proparcionada.

C. El sistema de adquisición de reglas, el cual permite ingresar nuevas reglas y modificar las existentes.

2.3 REPARACION, CORRECCION O TERAPIA.-

Luego que un Sistema Experto ha interpretado los datos y otro Sistema Experto ha hecha el diagnóstico es necesario entrar a la etapa de reparación. Un Sistema Experto entra en esta etapa, y se debe encargar básicamente de las siguientes puntos:

- dar un conjunto de reparaciones posibles para el problema

- escoger la reparación más rápida, económica y que no cause problemas secundarios

procedimientos participa a menudo la mano humana. Un operador o un proceso externo también automático puede hacer cambios en momentos que el Sistema Experto está por ejemplo ejecutando algún tipo de operación y no alcanza a terminar. Esto puede causar que el sistema quede bloqueado.

2.5 SIMULACIÓN, PRONÓSTICO O PREDICCIÓN.-

La simulación es una técnica consistente en crear modelos basados en hechos, observaciones e interpretaciones, sobre el ordenador con el fin de estudiar el comportamiento de los mismos mediante la observación de las salidas para un conjunto de entradas (sensibilidad del sistema frente a los datos).[3]

Llamemos al objeto a ser simulado objeto X. De acuerdo al párrafo anterior, para simular un objeto X, es necesario estudiar los resultados que presenta ese objeto X frente a un conjunto de entradas. Una vez que conocemos el comportamiento del objeto X, creamos un modelo, matemáticc o simbólico según sea el caso, que se comporte igual cuandc se le aplican las mismas variables de entrada.

---

[3]J.P. Sánchez y Beltrán, "Sistemas Expertos, Una Metodología de Programación", Macrobit. 1990. pag 88

Los Sistemas Expertos le dan a la simulación un toque especial, que no se puede lograr con la matemática pura, y es simular el comportamiento humano. Esta aplicación de Sistemas Expertos se utiliza típicamente combinada con métodos de simulación matemdticos. De esta forma hay un complemento entre los modelos matemáticos y el comportamiento humano.

Un ejemplo de Sistema Experto trabajanda en conjunto con un simulador puede ser un sistema de proyecciones de estados financieros. En este caso el simulador manejaria toda la parte matemática y el Sistema Experto analizaría los resultados de acuerdo al razonamiento que utilizaría un experto humano. Además el Sistema Experto podría sugerir acciones a tomar frente a los resultados obtenidos.

CAPITULO III

HERRAMIENTAS PARA CONSTRUIR SISTEMAS EXPERTOS

3.1 LENGUAJES.-

De acuerdo a J.P. Sánchez y Beltrán, "...no es la herramienta la que otorga las características de un Sistema Experto sino que es la estructura del programa,....."[4]

Según esta definición, nu importa el lenguaje de programación, o la máquina que se utiliza, sino lo importante es que se cumplan unas candicianes de estructura en la programación.

Partiendo del punto anterior podemos recorrer diferentes niveles de lenguajes posibles para programar un S.E, yendo de menor a mayor especialización, encontraremos lenguajes procedurales, lenguajes diseñados especialmente para inteligencia artificial y por último tenemos las conchas (shell). En los siguientes subtítulos revisaremos paso a paso cada una de estas herramientas.

---

[4]J.P. Sánchez y Beltrán, "Sistemas Expertos, Una Metodología de Programación", Macrobit. 1990. pag.101

3.1.1 LENGUAJES PROCEDURALES.-

Un lenguaje procedural es uno tal que su código cumple un recorrido típico "Top Down" de arriba hacia abajo. El programa empieza en un punto donde ejecuta 1a primera instrucci6n, 1a segunda instrucción está normalmente escrita a continuación de la primera y así sucesivamente se encuentran todas las instrucciones que forman un programa. "En pocas palabras, los lenguajes procedurales son aquellos en los cuales el programador describe un algoritmo -<<paso a paso>> a través de una serie de instrucciones- con el exacto e invariable camino que el programa debe seguir."[5]

El lenguaje más básico es el lenguaje ensamblador. El lenguaje ensamblador presenta en teoría una aptitud máxima en cuanto a la libertad que da al programador para controlar todo. Sin embargo por el mismo hecho de ser sumamente simple no tiene estructuras que permitan al programador solucionar problemas conocidos casi en forma automática. En el lenguaje ensamblador el programador tiene que programar hasta el más mínimo detalle.

Para ilustrar podríamos decir que el lenguaje

---

[5]Benchimol-Levine-Pomerol, Las Sistemas Expertos en 1a Empresa", Macrobit. 1.990. pag. 81

ensambladar permite controlar cada pieza de hardware que tenga el computador, pero para presentar solamente una pantalla bonita con menús de selección necesita que se haga un programa completo.

Entre los lenguajes procedurales más destacados se encuentran Fortran, Basic, Pascal, C, C++, entre otros muchos.

En este tipo de lenguaje programar un sistema como par ejemplo "Las diez reinas (Ten queens)" implica escribir un código sumamente complejo y difícil de comprender, recursivo y con muchas condiciones "si entonces", el cual toma entre 100 y 200 líneas de código.

Desarrollar un sistema igual en lenguajes más especializados como por ejemplo Prolog es mucho más fácil y el código resultante no toma más de 30 líneas.

3.1.2 LENGUAJES DE INTELIGENCIA ARTIFICIAL.-

Como comentamos anteriormente, na es el lenguaje lo que hace a un Sistema Experto, sina su estructura. Sin embargo, hay lenguajes que son más aptos para desarrollar Sistemas Expertos por que tienen incorporadas al lenguaje estructuras que simplifican el trabajo de programación.

Entre los lenguajes más conocidos para desarrollar Sistemas Expertos se encuentran el Lisp y el Prolog.

3.1.2.1 LISP.-

El lenguaje **Lisp** aparece a finales de los años 50, se basa en la combinación de átomos y listas. Las listas son una estructura de árbol binario. El tratamiento de listas se hace sumamente sencillo con Lisp, al igual que el manejo de estructuras madu1ares.

3.1.2.2. PROLOG (PROGRAMMING IN LOGIC).-

El PROLOG fue desarrollado en La universidad de Marsella par Alain Colmereaur y F'h. Roussel entre 1970 y 1980.

El PROLOG tiene par la menos dos cualidades dignas de ser destacadas. La primera es que se basa en la lógica de predicados. Y la segunda es que es un lenguaje declarativo. En un lenguaje declarativo el programador escribe el ambiente baja el cual el programa debe trabajar, pera no define un algoritmo que marque punto por punto el recorrido del proqrama. El programa debe empezar a correr baja ciertos parámetros, pero al enfrentarse con las diferentes alternativas debe ir creando el camino a seguir.

3.2 CONCHAS.-

Las Conchas (shell) son las herramientas de mas alto nivel para desarrollar Sistemas Expertos Una concha es una herramienta completa pero vacía de conocimientos, es decir le falta que el usuario le ingrese toda la información necesaria para hacerla un experto en un área determinada del conocimiento.

Los componentes principales de una concha son los siguientes:

- motor de inferencia
- base de conocimientos
- editor
- utilidades

Las utilidades pueden estar compuestas de los siguientes módulos:

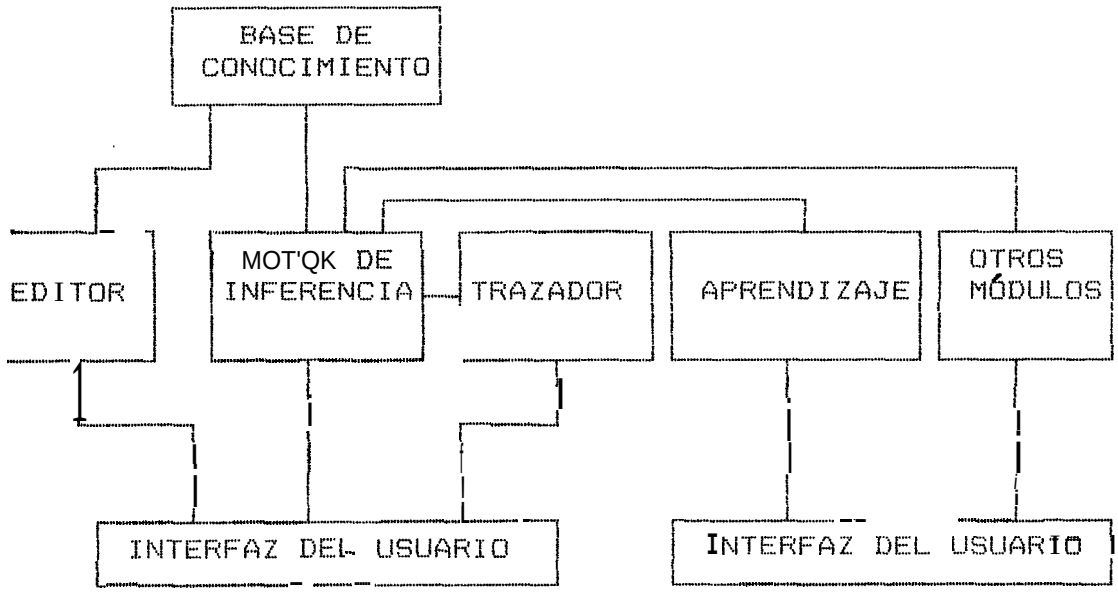
- traza
- motor de desarrollo
- modulo de aprendizaje

A continuación observaremos el diagrama de la arquitectura básica de una concha[6].

---

[6]Benchimol-Levine-Pomerol, "Los Sistemas Expertos en la Empresa", Macrobit. 1990. Página 96.

ARQUITECTURA BÁSICA DE UNA CONCHA O SISTEMA EXPERTO



Ya que los conceptos referentes al motor de inferencia y a la base de conocimientos explicados en el capítulo primero se aplican completamente a las conchas no los repetiremos en este capítulo. Pasaremos directamente a explicar que es el editor, la traza, el motor de desarrollo y el módulo de aprendizaje.

EDITOR

El editor tiene como fin ingresar el conocimiento a la base de conocimientos.

Un editor rudimentario solo hará esto. Pero un editor

que realmente se precie de serlo debe tener otras utilidades, como:

- facilitar la introducción de datos, mostrando los datos posibles siempre que haya un conjunto definido de los mismos
- verificar que la regla se ha escrito correctamente
- visualizar reglas que tengan las mismas premisas y/o las mismas conclusiones
- verificar que los datas ingresados a las variables estén dentro de los rangos de validez, o que sean del tipo correcto
- permitir ingreso de comentarios asociados al conocimiento
- listar reglas y los datos agrupados par subconjuntos

TRAZA

La traza es un elemento esencial del Sistema Experto Su principal. tarea es seguir, como su nambre lo indica, la traza de los razonamientos del motor de inferencia.

La **traza** es esencial porque explica al usuario el PORQUE y el COMO se llego a un resultado.

Por ejemplo si el sistema da una respuesta X, le

podemos preguntar ? COMO X ? (como llego a X ?)El sistema responderá que A y B => X.

Si el sistema nos pregunta ¿ es cierto D ? Le podemos preguntar ¿ POR QUE D? (por que pregunta por D ?) El sistema responderá PORQUE F es cierto, si D es cierto, F y D => Z

MOTOR DE DESARROLLO

El módulo utilizado durante el desarrollo se llama motor de desarrollo. Sirve para probar las reglas y debe permitir comprobar la coherencia con numerosos ejemplos.

Debe naturalmente proponer nuevas reglas. Además debe detectar las nuevas reglas que saliéndose de la red conducen a caminos sin salida.

MODULO DE APRENDIZAJE

Permite el ingreso de ejemplos y en base a estos formula nuevas reglas que le sirven para llegar a las mismas conclusianes de los ejemplos.

Muchas veces el motor de desarrollo va relacionado con el modulo de aprendizaje.

CAPITULO IV
ANÁLISIS Y DISEÑO
DEL SISTEMA EXPERTO
PARA SELECCIONAR PERSONIAL

4.1 DEFINICION,OBJETIVOS Y LIMITACIONES.-

Se quiere un Sistema Experto capaz de ayudar a las personas responsables de la selección de personal en las empresas. Además se necesita que el sistema sea capaz de manipular la información, de tal forma, que cada per-soria que se encuentre ingresada en la base de datos, sea un candidatn a cualquier puesto que se solicite. Y solamente la capacidad o la experiencia de esta persona será la que le brinde las oportunidades para obtener un puesto determinado.

El "SESEP" Sistema Experto para Se1eccionar Personal cumple todos los requisitos esperados. Y e s una herramienta muy útil para los profesionales en la rama.

Coma es de esperar el Sistema Experto para Seleccionar Personal tiene sus limitaciones, dadas por las siguientes circunstancias:

El conocimiento y la experiencia en nuestro medio, referente a Sistemas Expertos es nulo a casi nulo. Pues a más de que no existen en nuestro medio grandes empresas dedicadas al desarrollo de software, tenemos la dificultad de que el desarrollo de Sistemas Expertos es mucho más complejo que el desarrolla de sistemas tradicionales. Y es un área que se encuentra en el mundo aún envias de desarrollo.

El lenguaje de programación que se ha utilizado es el Turbo Prolog, el cual es uno de los más difundidos, y es claramente un lenguaje de inteligencia artificial. Sin embargo la última versión que desarrolló Borland, el propietario de la marca, es la 2.0 de 1.988. Esto hace que el lenguaje tenga muchas deficiencias, como la dificultad para hacer una interfaz amigable al usuario, el manejo de la memoria tampoco es eficiente; esto se nota en el momento de querer compilar el programa ejecutable, lo cual no se puede hacer con más de 200 líneas de código. Por stro lado la recuperación de memoria es mala, ya que en muchas ocasiones el programa se queda sin memoria después de que algunas funciones que la ocuparon la liberan, como es el caso de funciones recursivas que terminan.

Otras dificultades que se plantearon fueron la falta de

ejemplos, en los que se maneje la interfaz con el usuario, esto hubiese ayudado mucho para mejorar la utilización de recursos del lenguaje.

Sin embargo no todo es malo con el Turbo Prolog, si queremos resolver algún problema de lógica o en los que haya que buscar la solución a partir de premisas dadas, el Turbo Prolog se convierte en una herramienta poderosísima. Resumiendo, el Turbo Prolog, podemos decir que es un gran lenguaje, que es un ejemplo de lenguaje de inteligencia artificial, y que tiene mucho futuro si se lo sigue mejorando, pero si se lo deja como está, no tendrá mucho tiempo de vida.

4.2 ANALISIS.-

4.2.1 LA SELECCION DE PERSONAL.-

El planteamiento inicial para el desarrollo del sistema fue desarrollar un prototipo de Sistema Experto el cual sirva de herramienta para la selección de personal en las empresas.

Se debe tener en cuenta que la selección de personal tiene un aspecto muy objetivo hasta cierto punto y luego se vuelve sumamente subjetivo. Y normalmente la decisión

final es netamente subjetiva.

Si queremos simplificar el proceso de selección de personal, podemos decir que tiene dos partes principales, cuyo orden depende de la persona que lo maneja. Estas dcss partes son las pruebas y las entrevistas. Algunos gerentes de recursos humanos aplican primero la; pruebas, y otros aplican primero las entrevistas. La que si está bastante establecido es aplicar las dos cosas en algún mamento.

Después de la entrevista con el gerente de recursos humanos viene normalmente la entrevistas con el jefe directu de la persona que se busca. Estas dos entrevistas implican necesariamente que el candidato deberá pasar por dos pruebas sumamente subjetivas.

Lo que hemos dicho hasta el mamento puede parecer un poco fuera de lugar, pero no es así, par-que estas cavilaciones fueron las que nos 1levaron a buscar un sistema que sea flexible.

La flexibilidad debía basarse en dos puntos primordiales, el primero es dar la posibilidad al usuario de crear su propio entorno de trabajo. Es decir permitir

que el usuario libremente cree el formato con el que va a definir un cargo y con el que va a medir a los candidatos.

El segundo punto es copar todos los aspectos objetivamente posibles, pero dejar la puerta abierta para que el usuario haga sus propias interpretaciones y tame la decisión final. Permitir además al usuario disminuir o aumentar los requerimientos a medida que aprende las mejoras que puede hacer en las definiciones de los cargos.

4.2.2 EASES DE DATOS.-

Por otro lado, desde el punto de vista técnico inicialmente se observa la existencia dos tipos de datos. Datos que definen cargos y datos de personas. La interrogante planteada enseguida fue, como relacionar los datos de las personas con los de los cargos. Pues si un gerente departamental pide un jefe de área can unos requisitos; y por otro lado los candidatos presentan cada uno sus curriculums, con datos que no contestan las preguntas de los requisitos del puesto, ¿ cómo se relaciona la información 3

Por otro lado si se estandariza la información para un cargo y los curriculums que presentan los candidatos, ¿ qué ocurre si se requiere candidatos para otro puesto, con

requisitos que no se pidieron antes ?, los primeros candidatos no sirven porque no se conoce nada de ellos en las áreas de la nueva vacante, porque nadie preguntó antes.

Surge entonces la idea de manejar tres bases de datos y no dos.

La nueva **base** de datos almacenaria parámetros generales. Luego en todos los cargos se definirían estos parámetros, y a todos los candidatos se le preguntarían todos sus datos referentes a estos parámetros.
Utilizando tres bases de datos, se lagra que cualquier persona que presentó su curriculum **sea** un candidato a cualquier puesto, inclusive si ya esta persona trabaja en la empresa.

Las tres bases de datos serían las siguientes:

A. Ease de datos de Parámetros
B. **Base** de datos **de** Perfiles **de** Per-sonas
C. Base **de** datos de Perfiles de Cargos

La base de datos de parámetros tiene como fin unificar criterios para manipular la información tanto de las

personas como de los cargos.

Ejemplos de parámetros son lus siguientes:

Nombre

Apellido

Idioma

Colegio

Universidad

Carrera

Especialización

Las bases de datos de personas y de perfiles funcionan muy parecidas, pues cada una de ellas almacena el código de la persona (número de cédula) o del cargo (nombre del cargo), el nombre del parámetro cuyo dato se va a almacenar, y por último el dato mismo. A continuación tenemos un ejemplo ilustrativo.

Ejemplo:

para decir que la persona cuya cédula de identidad es 0000000011, habla el idioma inglés:

(0000000011,idioma,inglés)

para decir que el cargo de gerente financiero requiere que la persona hable francés:

(gerente financiero,idioma,francés)

4.2.3 MODULARIDAD.-

Basados en el planteamiento para manejar las bases de datos, podemos pensar que lo correcto para modularizar el programa es seguir el mismo esquema. Es decir, crear un módulo que manipule la información de parámetros, uno que maneje lo que corresponde a los datos de las personas y por último uno que se encargue de los perfiles de los cargos.

Además será necesario crear un módulo que haga el proceso mismo de selección.

Fuera de estos cuatro módulos se necesita crear un 61timo que sirva para utilitarios, como copiar archivos, firmador discos, etc.

4.3 DISEÑO.-

4.3.1 BCISE DE CONOCIMIENTOS.-

La base de conocimientos se fundamenta en HECHOS que relacionan la información. Estos HECHOS son del tipo:

HECHO(dato1,dato2,...,datoN)

el ejemplo siguiente explica mejor el concepto:

person("0920807544","NOMBRE","ESTEBAN")

lo cual significa que la persona cuyo numero de cédula es 0920807544 tiene el NOMBRE de ESTEBAN.

Vale acotar que con el fin de simplificar la codificación **de** las personas el Sistema Experto para Seleccionar Personal utiliza el "número de cédula" como clave de identificación de las personas;; para los cargos el sistema utiliza como clave el "nombre del cargo".

Para almacenar el conocimiento el **sistema** utiliza los siguientes HECHOS:

- parámetros("PARÁMETRO","PROFESIONAL O PERSONAL", "SIGNO")
- person("CEDULA","PARÁMETRO","DATO")
- perfil("CARGO","PARÁMETRO","DATO","PUNTAJE")
- necesario("CARGO","PARÁMETRO","SI o NO")

Estos HECHOS sirven para definir tres bases de datos, que marcan los lineamientos generales del sistema.

La primera base de datos maneja lo que llamamos PARÁMETROS, estos parámetros son los que definen las condiciones que se van a comparar entre un perfil de un cargo, y el perfil de una persona.

Par ejemplo el IDIOMA que se requiere en el cargo, será comparada con el IDIOMA que habla la persona. O la EDAD que se requiere en el. cargo sería comparada con la EDAD de la persona.

En los casos anteriores IDIOMA y EDAD son PARÁMETROS.

La segunda posición indica si el PARÁMETRO es del tipo PROFESIONAL O PERSONAL, esto es importante para diferenciar si un PARÁMETRO se debe comparar con las características de un cargo, o simplemente sirve como data referencial de la persona. Por ejemplo el nombre y la dirección de una persona son datos PERSONALES, peru el idioma y la experiencia son datos PROFESIONALES. El sistema utiliza únicamente los datos PROFESIONALES para efectos de escoger al mejor candidato.

La posición denominada SIGNO, sirve para comparar los datos. Recordemos que no necesariamente todos los **datos** deben ser iguales entre un perfil de cargo y el perfil de la persona. Pues en muchas ocasiones se fijan simplemente límites para algunas condiciones de los cargos. Cuando buscamos personas que sean menores de veinte años o cuando queremos personas que **pol-** lo menos hayan estudiado cinco años, estamos preguntando por edades menores al perfil, o por experiencias mayores o iguales a cinco años.

El SIGNO el sistema lo interpreta de acuerdo **a** la siguiente tabla:

=	igual
>	mayor igual que
<	menor igual que

Aunque matemáticamente los signos > y < significan solamente "mayor que" y "menor que", por fines prácticos se ha desarrollado el sistema **para** interpretarlos como "mayor igual que" y "menor igual que".

Pasemos **a** la segunda base cuya estructura se asienta sobre el HECHO PERSON. Esta **base** de **datos** almacena toda la

información referente a las personas. Ésta más, cada HECHO de esta base de datos es una característica de la persona.

Primero se debe poner el número de cédula de identidad, luego el PARÁMETRO que se define, y por último el DATO correspondiente a dicho PARÁMETRO.

Si queremos decir que la persona con número de cédula de identidad 0909090909 habla el idioma francés, debemos expresarlo con el siguiente HECHO:

```
person("0909090909","IDIOMA","FRANCÉS")
```

La última base de datos funciona alrededor de dos HECHOS. Estos hechos son:

```
perfil("CARGO","PARÁMETRO","DATO","PUNTAJE")

necesario("CARGO","PARÁMETRO","SI o NO")
```

El HECHO PERFIL funciona igual que el HECHO PERSON, sirve para definir cada una de las condiciones requeridas para un cargo.

Habremos notado que el HECHO PERFIL tiene un cuarto campo que no existe en PERSON. Este campo sirve para darle un valor a cada condición.

Si tenemos claro que para el contralor de la empresa es mucho más importante tener experiencia en contabilidad que saber hablar el idioma inglés, entonces seguramente le daremos más puntaje a la característica de experiencia que a la de idioma. Utilizando este sistema de puntajes podemos lograr que al final las personas que cumplen las condiciones más importantes para el cargo, tengan puntajes más altos.

Asociado al HECHO PERFIL, se encuentra el HECHO NECESARIO, NECESARIO sirve para dar una caracteristica de precisión al programa.

Explicaremos primero la importancia de asignar a ciertas candiciones un carácter de absoluto.

Cuando el. perfil del cargo tiene una o varias condiciones que si nu se cumplen eliminan automáticamente a la persona, hay que definir esta condición coma absoluta. Esto quiere decir que si no se cumple la persona no sirve para el cargo. A continuación veremos varios ejemplos que nos aclararán la idea:

- cuando queremos pilotos de avión no aceptamos personas mayores a 50 años
- cuando buscamos un agregado cultural para Francia requerimos que **hable** francés
- cuando buscamos **a** una persona que de clases prenatales **a** señoras embarazadas buscamos a una mujer

Aunque cualesquiera de las personas de los ejemplos anteriores tenga una mente muy privilegiada, o **hable** inglés perfecto, no sirve para el cargo que buscamos, porque no cumple lo que llamamos una condición necesaria.

El HECHO NECESARIO consta de tres relaciones, que son el CARGO, el PARÁMETRO, y el SI o NO, lo cual sindica si ese parámetro en ese cargo, es una condición necesaria o no lo es.

Cuando el programa selecciona **a** las personas escribe al lado un campo que lo denominamos **"status"!**, el "status" puede marcar "ok" o "no" si marca "ok" indica que la persona cumplió todas las condiciones necesarias. Si marca "no", indica **que la** persona incumplió por lo menos una de las condiciones necesarias.

En conclusión el sistema cuenta con tres bases de datos que almacenan los parámetros, las características de las personas y las características de los cargos. Cuando el programa selecciona compara las características de las personas con las características de los cargos. Si una de las características del cargo considerada necesaria no se cumple, el sistema marca el campo "status" con la palabra "no", caso contrario el "status" indica "ok".

4.3.2 REGLAS.-

Las REGLAS se pueden definir como el camino que utiliza un Sistema Experto para encontrar una solución.

Las REGLAS usualmente utilizan a los HECHOS como información para llegar a las respuestas o para obtener conclusiones, lo cual para el caso es lo mismo.

A continuación veremos un ejemplo tomado del manual del usuario de Turbo Prolog versión 2.0.

Tenemos los siguientes HECHOS:

```
le_gusta(bill,cindy)
le_gusta(cindy,bill)
le_gusta(bill,dogs)
```

significan:

a **bill** le gusta cindy

a cindy le gusta bill

a bill le gustan los perros

Tenemos la siguiente REGLA:

a cindy le gusta **lo** que le gusta a **bill**

lo que en el lenguaje de Turbo Prolog se traduce:

le_gusta(cindy,Algo) si le_gusta(bill,Algo)

dicho de otra forma, para que **a cindy** le guste alga, tiene que gustarle **a** bi11

En el subtema 4.3.3 veremos el. proceso de razonamiento para **encontrar** la solución al problema, por el momento **nos** interesa concentrarnos **en** las REGLAS.

El Sistema Experto para Seleccionar Personal, tiene por supuesto, sus REGLAS más importantes en **el** módulo de selección.

Estas REGLAS son las siguientes:

comparar("CARGO","PARÁMETRO","DATO_CARGO","CEDULA","SIGNO",
" NECESARIO","VALOR","PUNTOS","STATUS")

consultar_repetidos("PARÁMETRO","NUMERO EN LISTA","N VECES
REPETIDO")

consultar_datos_perfiles2("CARGO","PARÁMETRO","DATO","PUNTA
JE"," N VECES REPETIDO")

status("STATUS",STATUS")

Hemos escogido estas REGLAS para explicarlas porque a más de estar involucradas en el momento de la selección, cada una sirve de ejemplo de otras muchas que están involucradas en el proceso.

comparar("CARGO","PARÁMETRO","DATO_CARGO","CEDULA","SIGNO",
"VALOR","PUNTOS","STATUS")

Esta regla es el corazón de todo el sistema, porque es la que hace el proceso de comparación.
Tiene como datos de entrada:

CARGO

PARÁMETRO

DATO_CARGO

CEDULA

NECESRRIO

SIGNO

VALOR

Y coma datos de salida:

PUNTOS

STATUS

La función compara los datos del cargo ingresados, con los de la persona portador-a del número de cédula CEDULA. Para esta comparación utiliza los datos de entrada CARGO, PARÁMETRO, DATO_CARGO y CEDULA. Utiliza NECESARIO para identificar si la condición es considerada necesaria, y SIGNO para hacer la comparación "=" (igual), ">=" (mayor igual) y "<=" (menor igual). Utiliza el VALOR para, en caso de que la condición se cumpla, asignar a esa persona la cantidad de puntos contenida en VALOR.

Los datos PUNTOS y STATUS, son la salida de la función. PUNTOS, es la variable que guarda el VAL-OH de la condición cuando la persona ha cumplido el requisito. El

campo STATUS, indica "no" cuando una de las condiciones necesarias no se ha cumplido; en el caso contraria se mantiene como una variable sin asignar.

Al terminar el proceso de selección se chequea STATUS, si tiene valor "no", se lo deja así. Si la persona cumplió todas las condiciones necesarias, STATUS debe mantenerse como una variable no asignada, y es en este momento que se le asigna el valor- "ok".

Par-a resumir la REGLAS "comparar" tiene dos salidas. PUNTOS indica los puntos que la persona ganó por cumplir con un requisito determinado. STATUS indica "no" si la persona no cumplió alguno de los requisitos necesarins.

consultar_repetidos("PARÁMETRO","NUMERO EN LISTA","NUMERO REPETIDO")

Internamente se genera una base de datos temporal transparente a1 usuario, la cual sirve para manipular los parámetros repetidos. Los parámetros repetidos son uno de los problemas que hay que resolver al plantearse desarrollar un sistema de selección de personal. Estos se dan cuando para un cargo se requiere más de un data para un

mismo parámetro. Par ejempla cuando requerimos que una persona hable más de un IDIOMA, o tenga más de una EXPERIENCIA DE TRABAJO, o cuando la carrera que se necesita que haya estudiado puede ser por ejemplo "Ciencias Sociales" a "Psicología" o "Psiquiatría".

Este es uno de los problemas más complejos, porque el sistema debe buscar en la base de datos de cargos todas las posibilidades que tiene un misma parámetro y camparar cada posibilidad cnn la base de datos de personas.

Si un cargo requiere que la persona hable por lo menos uno de los siguientes idiomas:

INGLéS
ALEMÁN
FRANCéS

Y la persona habla los siguientes idiomas:

IRANCéS
ESPAÑOL

El sistema deberá contrastar cada uno de los idiomas

requeridos contra cada uno de los idiomas que habla la persona, solamente deberá asignar puntos al idioma FRANCéS, pero el. STATUS na deberá cambiarlo a "no" al verificar los idiomas INGLéS y ALEMÁN.

Recordemos que la condición requería que hable por lo menos uno de los idiomas mencionados. Como la persona si habla por lo menos uno de los idiomas, aunque no hable los otros dos, deberá concedérsele un "ok".

Nos hemos detenida a explicar el prnblema de los parámetros repetidos, pero nn hemos explicado la REGLA "consultar_repetidos", que causo la explicación anterior. Volviendo pues a esta HEGLA, podemos decir que sirve para asignar una relación a los parámetros. Esta relación se compone de dos números muy útiles. El primero es la posición que ocupa dicho parámetro dentro de la lista de pardmetros. El segundo indica el número de veces que dicho parámetro se encuentra repetido. Estos dos datos se los consulta de la base de datos temporal que nombramos anteriormente. Y sirven coma datos de entrada a la regla que observaremos a continuación.

consultar_datos_perfiles2("CARGO","PARÁMETRO","DATO","PUNTA

JE"," N VECES REPETIDO")

Como la REGLA anterior deciamos que sirve para obtener los datos de entrada para la REGLA "consultar_repetidos", también la REGLA "consultar_datos_perfiles2" tiene como fin obtener datus para la REGLA "comparar", la primera de las reglas revisada, y la más importante de todo el sistema.

La REGLA "consultar_datos_perfiles2" busca el dato correspondiente al CARGO, al parámetro PARÁMETRO y el dato se encuentra N VECES REPETIDO.

status("STATUS",STATUS")

La REGLA "status" es una de las más sencillas, pero es muy efectiva. Esta REGLA revisa que si variable STATUS está asignada. Si está asignada, entances la deja igual. Si la variable nu está asignada, entonces se le asigna el valor- "ok".

Recordemos que la variable STATUS puede tener los valores "no" u "ok", al explicar la REGLA "comparar" decíamos que a la variable STATUS se le asigna "no" cuando una de las condiciones necesarias nu se cumple. Pues si la

regla "comparar" termina de revisar todas las características de la persona, y esta variable nu ha sido asignada significa que se le debe asignar "ok". Es en este momento cuando entra a funcionar la REGLA "status", la cual hace la asignación definitiva.

4.3.3 PROCESO DE RAZONAMIENTO.-

A continuación entraremos en más detalle al proceso que utiliza el motor de inferencia en cada una de las reglas descritas en el subtema 4.3.2.

```
comparar("CARGO","PARÁMETRO","DATO_CARGO","CEDULA","SIGNO",
"VALOR","PUNTOS","STATUS")
```

El código de la REGLA "comparar" es el siguiente:

```
comparar(_,Parámetro,DatoCargo,Cedula,"=",_,Puntos,Puntos,_
):-
        person(Cedula,Parámetro,DatoPersona),
        DatoPersona=DatoCargo,!.

comparar(_,Parámetro,DatoCargo,Cedula,"<",_,Puntos,Puntos,_
):-
```

```
            person(Cedula,Parámetro,DatoPersona),
            fronttoken(DatoPersona,DatoPersonaStr,_),
            fronttoken(DatoCargo,DatoCargoStr,_),
            str_int(DatoCargoStr,DatoCArgoInt),
            str_int(DatoPersonaStr,DatoPersonaInt),
            DatoCargoInt<=DatoPersonaInt,!.

comparar(_,Parámetro,DatoCargo,Cedula,">",_,Puntos,Puntos,_
):-
            person(Cedula,Parámetro,DatoPersona),
            fronttoken(DatoPersona,DatoPersonaStr,_),
            fronttoken(DatoCargo,DatoCargoStr,_),
            str_int(DatoCargoStr,DatoCargoInt),
            str_int(DatoPersonaStr,DatoPersonaInt),
            DatoCargoInt>=DatoPersonaInt,!.

comparar(Cargo,Parámetro,_,Cedula,"=",_,_,0,_):-
            perfil(Cargo,Parámetro,Dato,_),
            person(Cedula,Parámetro,DatoPersona),
            DatoPersona=Dato,!.

comparar(_,_,_,_,_,SN,_,0,Status):-
            SN="S", Status="no",!. /*0 rechazado*/
comparar(_,Parámetro,_,_,_,"N",_,0,_):-
```

```
        parámetros(Parámetro,"E",_),!.
comparar(_,_,_,_,_,_,_,0,_):-!.
```

Está construido en siete bloque cada uno de los cuales representa una especie de "case" de los lenguajes procedurales.

Los bloques primero, segundo y tercero se diferencian básicamente en que cada uno tiene un SIGNO diferente de comparación, el primera tiene el signo "=", el segundo el signo "<", y el tercero el signo ">". La única diferencia extra entre el bloque uno y los bloques das y tres es que el dos y tres comparan datos numéricos y parte del código se utiliza para transformar el "string" en número.

De las tres bloques analizaremos el primero que es el más representativo.

La definición de la REGLA es la siguiente:

```
comparar("CARGO","PARÁMETRO","DATO_CARGO","CEDULA","SIGNO",
" NECESARIO","VALOR","PUNTOS","STATUS")
```

La REGLA real tiene menos información porque en este casa el nombre del cargo nu será necesario, ya que el dato

que hace es retroceder a la línea anterior y buscar otra posibilidad a DatoPersona. Si esta persona tiene otro dato para el mismn parámetro modifica DatoPersona y pasa a la siguiente línea. Si esta vez se cumple que DatoPersona es igual a DatoCargo la regla asigna el puntaje y termina. Si nu se cumple retrocederá las veces que sean necesarias hasta obtener una respuesta positiva o hasta que la persona no tenga más características para el mismo parámetro.

Utilicemos el ejemplo de los idiomas:

Si un cargo requiere que la per-snna hable:

FRANCÉS

Y la persona habla los siguientes idiomas:

ESPAÑOL
FRANCÉS

El programa comparará FRANCÉS can ESPAÑOL. L.n cual dará un error, pero el programa retrocede a la línea anterinr

person(Cedula,Parámetro,DatoPersona)

Busca otra característica para DatoPesona que tenga la persona con numero de cédula Cédula y que pertenezca al parámetro Parámetro. Esta característica es FRANCéS. Ahora DatoPersona tiene el dato FRANCéS, lo compara con DatoCargo, y la comparación es correcta.

Cada bloque se pr-ueba únicamente si el anterior falla. Es decir, los bloques cuatro, cinco, seis y siete ocurren únicamente si fallan los bloques uno, dos y tres.

A su vez si se cumple el bloque cuatro, los bloques cinco, seis y siete no se activan.

Los bloques cuatro, cinco, seis y siete **son** diferentes cada uno del otro, así **que** los revisaremos uno por uno.

Cuando se compara un dato de un perfil frente a todos los datos de una persona, y ese dato no se cumple, y además ese dato es necesario, entonces parecería que la persona merece un STATUS "no".

Esto no es verdad, porque primero nos debemos asegurar

que ese parámetro no tiene mds condiciones posibles que se puedan cumplir. Recordemos que hemos partido de que cuando hay varias posibilidades en un parámetro considerado necesario, por la menos una se debe cumplir. No hemos dicho que todas se deben cumplir, sino por lo menos una. Los correcto en este caso es buscar atro Dato que se pueda cumplir. Si otro Dato es igual al DatoPersona entonces no le asignamos "no" a STATUS.

```
comparar(Cargo,Parámetro,_,Cedula,"=",_,_,0,_):-
        perfil(Cargo,Parámetro,Dato,_),
        person(Cedula,Parámetro,DatoPersona),
        DatoPersona=Dato,!.
```

La regla Dato con DatoPersona, si no se cumple retrocede y cambia DatoPersona, vuelve a comparar. Mientras hayan más DatoPersona seguirá cambiándolo. Si la comparación sigue siendo negativa y no hay más DatoPersona, el motor de inferencia retrocede una línea más, modifica Dato y empieza nuevamente el proceso.

Este bloque fallará si no existe para el parámetro que se analiza ningún perfil de cargo que sea igual a un dato de persona.

Volvamos a1 ejemplo de los idiomas.

Si. un cargo requiere que la persona hable pnr lo menos uno de los siguientes idiomas:

INGLéS

ALEMÁN

FRANCéS

Y la persona habla los siguientes idiomas:

FRANCéS

ESPAÑOL

En el momento que el idioma que se analiza es el primero, y no se cumple la comparación del bloque uno, implica que la persona no habla INGLéS, el bloque dos y el bloque tres también fallarán, porque comparan números y no "strings".

El programa intentará con el bloque cuatro. Este b1oque i.ntentará 1o siguiente:

INGLéS = FRANCéS => ERROR

INGLéS = ESPAñOL => ERROR

Retrocederá e intentará:

ALEMÁN = FRANCéS => ERROR

ALEMÁN = ESPAñOL => ERROR

Volverá a retraceder e intentará el último idioma que le queda:

FRANCéS = FRANCéS => CIERTO

Como acertó, el motor de inferencia no irá a los siguientes bloques. Y la persona no tendrá un STATUS "no", porque no se la merece.

```
comparar(_,_,_,_,_,SN,_,0,Status):-
        SN="S", Status="no",!.
```

Si las pruebas anteriores han fallado, el motor de inferencia pasa al bloque cinco. SN es la variable que indica si la condición que se analiza es necesaria o no. Si SN es igual a "S", entonces asigna el Status "no", y le asigna un puntaje de cero.

```
comparar(_,Parámetro,_,_,_,"N",_,0,_):-
        parámetros(Parámetro,"E",_),!.

comparar(_,_,_,_,_,_,_,0,_):-!.
```

Si el dato que se busca pertenece a los parámetros personales, se le asigna cero al puntaje, al igual que si el dato buscado no pertenece a ninguna de las condiciones mencionadas anteriormente. Esto ocurre en los bloques seis y siete.

Pasaremos a la regla "consultar_repetidos":

```
consultar_repetidos("PARÁMETRO","NUMERO EN LISTA","N VECES
REPETIDO")
```

El código de la REGLA es el siguiente:

```
consultar_repetidos(Parámetro,NLista,N):-
     repetidos(Parámetro,NLista,N),!.

consultar_repetidos(_,_,0):-!.
```

En este caso tenenios dos bloques. El primer bloque es el principal, el segundo sirve únicamente para evitar que por alguna falla el programa se caiga.

Los datos de entrada sun:

PARÁMETRO

NÚMERO EN LISTA

El dato de salida es:

N VECES REPETIDO

La REGLA recibe el nombre del parámetro y el número de ese par-Ametra en la lista, y llama al HECHO

repetidos(Parámetro,NLista,N)

"repetidos" es una base de datos temporal en la cual se encuentran todos los parámetros acompañados de un número que indica la posición dentro de la lista y de un segundo número qcie indica el número de veces que se encuentra repetido con respecto a sus predecesores.

Al llamar al HECHO "repetidos" este responde el número de veces que se encuentra repetido N.

El siguiente ejemplo nos aclarará como funciona la base de datos.

PARÁMETRO	NUMERO EN LISTA	N VECES REPETIDO
EDAD	1	1
ESTUDIOS	2	1
ESTUDIOS	3	2
ESTUDIOS	4	3
UNIVERSIDAD	5	1
IDIOMA	6	1
IDIOMA	7	2

El dato que se obtiene de "consultar_repetidos" sirve para ser ingresado a la REGLA "consultar_datos_perfiles2".

La siguiente REGLA que veremos es:

```
consultar_datos_perfiles2("CARGO","PARÁMETRO","DATO","PUNTA
JE",    "N VECES REPETIDO")
```

El código es el siguiente:

```
consultar_datos_perfiles2(Cargo,Parámetro,Dato,Puntaje,
```

```
      NRepetido):-
            retractall(total(_)), asserta(total(1)),
            perfil(Cargo,Parámetro,Dato1,Puntaje),
            consultar_total(Total),          Total1=Total+1,
            asserta(total(Total1)),
            Total=NRepetido,
            Dato=Dato1,!.

consultar_datos_perfiles2(_,_,"",0,_):-!.
```

Los datos de entrada de esta REGLA son:

CARGO

PARÁMETRO

NREPETIDO

Los datos de salida son:

DATO

PUNTAJE

La idea en esta REGLA es encontrar el dato que le corresponde a un parámetro. Sabemos que este parámetro puede estar repetido. Si el parámetro está en la posición

uno, tomaremos el primer dato, pero si está en la segunda posición tomaremos el segundo dato, en fin si está en la posición N tomaremos el dato de la posición N.

Esta REGLA trabaja de la siguiente forma:

A. asigna cero a la variable TOTAL

B. graba el valor 1 en la variable TOTAL

C. llama a la base de datos "perfil", la cual se explicó en el subtema 4.3.1

D. consulta el valor de la variable total y graba el valor- más uno

E. compara la variable total con el número de veces que el parámetro está repetido

F si son iguales asigna DATO1 de la base de datos a la salida de la variable DATO

si no son iguales el motor de inferencia retrocede hasta la posición en que se llama a la base de datos "perfil", para buscar una nueva opción, luego repetimos este proceso desde el punto 4, hasta encontrar la solución al problema.

Si algo falla se hace el segundo bloque que se encarga de asignar "" a la variable DATO, y 0 al PUNTAJE. Esta

opción no debería de ocurrir, pero se la utiliza por seguridad.

La última REGLA que revisaremos se llama "status".

La variable de entrada es la primera y la de salida la segunda.

```
status("STATUS",STATUS")

status(StatusTemp,StatusTemp):-
     bound(StatusTemp),!.
status(_,"ok").
```

La REGLA recibe un dato en la primera variable que debe ser "no". En la segunda línea se pregunta si esa variable está asignada. Si la variable es "no" entonces sabemos que está asignada, y la variable se copia automáticamente en la de la salida.

Si la variable no está asignada, cae en el segundo bloque de la REGLA "status", el cual solamente asigna "ok" a la variable.

CAPITULO V

DESCRIPCIÓN REL SISTEMA EXPERTO PARA SELECCIONAR PERSONAL

5.1 IMPLEMENTACIÓN Y PRUEBAS.-

La implementación del Sistema Experto para la Selección de Personal es sencilla, consiste en copiar el archivo "SESEP.EXE" al directorio de trabajo.
Fuera de esto no es necesario hacer ninguna instalación adicional.

El sistema se ha probado en la selección de dos cargos, en dos empresas diferentes y con diferentes usuarios (Gerentes de Recursos Humanos). Estos dos usuarios utilizaron criterios bien diferentes para la selección. En el primer caso no se tomó en absoluto pruebas de aptitudes y en el segundo estas pruebas tuvieron el mayor peso en puntaje.

Para proteger a los candidatos a la selección se han utilizado solamente las iniciales de sus nombres y apellidos; los nombres de las empresas han sido cambiados.

Detalles impresos de cada caso se encontrarán en el

apéndice.

Caso 1.-

Se quiere contratar en la empresa "Distribuidores Nacionales" un "Ejecutivo de Cuentas", el cual debe cumplir con los siguientes requisitos:

-Edad minima	24
-Edad máxima	31
-Estudios	Adm.de Empresas
	Psicología
	Mercadotecnia
-Manejar Lotus	
-Manejar Wordperfect	

Se prefiere que la persona sea graduada en su carrera, además es preferible que la carrera sea de ingenieria a que sew de tecnología.

Para el perfil antes descrito se consideran indispensables que se cumplan los requisistos de la edad y la carrera profesional.

Se construyó el perfil y se ingresó al Sistema Experto

para Seleccionar Personal. Como resultado quedó el perfil siguiente.

PARAMETRO	DATO DE PERFIL	NECESARIO	PUNTAJE
01.-EDAD MINIMA	:24	[S]	[00] *
02.-EDAD MAXIMA	:31	[S]	[00] *
03.-SEXO	:- - -	[N]	[00]
04.-ESTUDIOS	:ADM. DE EMPRESAS	[S]	[20] *
05.-ESTUDIOS	:PSICOLOGIA	[S]	[20] *
06.-ESTUDIOS	:MERCADOTECNIA	[S]	[20] *
07.-ANIOS DE ESTUDIO	:- - -	[N]	[00]
08.-TECNOLOGIA (S/N)	:S	[S]	[01] *
09.-INGENIERIA (S/N)	:S	[N]	[05] *
10.-GRADUADO (S/N)	:S	[N]	[03] *
11.-ESPECIALIZACIO	:- - -	[N]	[00]
12.-IDIOMA INGLES	:N	[N]	[00]
13.-COMP.LOTUS (S/N)	:S	[N]	[02] *
14.-COMP. WP (S/N)	:S	[N]	[02] *
15.-EXPERIENCIA	:- - -	[N]	[00]
16.-AÑOS DE EXPER.	:- - -	[N]	[00]

Nótese que el sistema tiene más parámetros de los necesarios, estos parámetros extra, son muy importantes en

otros cargos, es par esto que fueron ingresados. Hay que recardar que el sistema fue desarrollada para trabajar con varios cargos a la vez.

Con el fin de facilitar la observación rápida de los parámetros que nos interesan, se han dibujado asteríscos al lado derecho del puntaje de cada uno.

A continuación vamos a revisar algunos puntos interesantes sobre la definición de las parámetros.

Los par-&metros 01 y 02, EDAD MINIMA Y EDAD MAXIMA respectivamente, tienen una [SI indicando que son absolutamente necesarios. Sin embargo en el puntaje tienen [OO], porque en este caso no es que preferimos a las personas que se encuentran entre los 24 y 31 años, sino que solamente nos interesan éstas, por tanto no tiene sentido asignar puntos a la edad.

Los parámetros enumerados 04, 05 y 06, tienen una particularidad especial, los tres parámetros son iguales. En este punto cabe mencionar que en la primera prueba que se hizo al sistema, éste no daba la opción a utilizar parámetros repetidas y los resultados fueron desastrozos.

Pues si nos interesa cualquier candidato que cumpla una de var-ias opciones, y no se pueden repetir- lus parámetros, entonces para dar igual posibilidades a todos no debemos poner ninguna de las condiciones; sino estaríamos dandn ventaja a la condición que si se escriba sobre la que no, a pesar de que para nosotros las dos posibilidades son cor-rectas. En la ocasión que narraba ocurrió que eliminamos esa característica, y los resultados salieron distorsionados, porque el programa puso al mismo nivel a los que competían por otros cargos, independientemente de si habían estudiado lo que se requería para el puesto o no.

La solución es permitir que se ingresen varios parámetros iguales cuando se requieran. El sistema considerará que los candidatus deben cumplir por lo menos uno de los requisistos con parámetros iguales.

Los siguientes parámetros que nos interesan son los 08, 09 y 10, éstos sirven para dar ventajas a los candidatos que hayan estudiado una carrera de tecnología, o una de ingeniería y que se han graduado. La importancia asignada la da el puntaje. De acuerdo al siguiente orden.

PARAMETRO PUNTAJE

INGENIERIA	05
GRADUADO	03
TECNOLOGIA	01

Por ejemplo, una persona graduada en ingenieria, obtendrá ocho puntos, cinco por ingeniería y tres por haberse graduado; mientras que una que estudia una tecnología y aún no se ha graduado, sólo obtendrá un punto.

Los últimos dos parámetros que nos interesan son el 13 y el 14, indican si la persona ha maneja LOTUS y/o WORDPERFECT. Se le asigna dos puntos por cada una de estas condiciones que se cumplan. Si no se cumplen no pasa nada porque no son consideradas condiciones necesarias.

Después del praceso de selección, los cuatro candidatus más destacados completaron entre 20 y 28 puntos de un total de 32 puntos posibles. Sin embargo el primero y el segundo, teniendo 28 y 23 puntos, fueron considerados como no aptos por el sistema, el sistema escribió un "no" junto al puntaje de cada uno.

Los candidatos tercero y cuarto completaron 22 y 20 puntos respectivamente.

En este momento fue aconsejable que el usuario chequee el detalle de los resultados. Al ingresar a la opción de detalle se encontró que el primero y el segundo candidato fueron declarados como no aptos porque no cumplen con el mínimo de edad, que es 24. El candidato mejor opcionado tiene 22 años de edad. El segundo mejor opcionadn tiene 23 años de edad.

La decisión final fue seleccionar al candidato que tuvo mejor puntaje en el sistema, ya que el asunto de la edad se lo obvió tomando en cuenta que este candidato a los 22 años ya se habia gradciado en la carrera de Ingeniería en Mercadotecnia, y pur tanto se consideró que tenía la madurez suficiente como para desempeñar el cargo.

Caso 2.-

Se requiere contratar en la empresa "Importadores del Ecuador" un "Asistente de Costos".

Los requisistos que debe cumplir el nuevo "Asistente de Costos", son los siguientes:

PARAMETRO	DATO DE PERFIL	NECESARIO	PUNTAJE
01.-EDAD MINIMA	:20	[N]	[00] *
02.-EDAD MAXIMA	:25	[N]	[00] *
03.-IDIOMA	:INGLES	[S]	[20] *
04.-IDIOMA	:- - -	[S]	[00]
05.-EXPERIENCIA	:- - -	[N]	[00]
06.-RAZONAM. VERBAL	:79	[N]	[10] *
07.-TRABAJO A PRESIO	:79	[N]	[10] *
08.-POTENCIAL	:79	[N]	[20] *
09.-HABILIDAD NUMER.	:79	[N]	[10] *
10.-HA TRABAJ.(S/N)	:S	[S]	[00] *

Se notará que los parámetros son diferentes a las del. Caso 1. Esto se debe a que cada uno de los usuarios utilizó un archiva diferente. Si se hubiese utilizado el mismo archivo, los parámetros serían los del Caso 1 más los del Caso 2, además los candidatos de ambos casos estarían presentes en los dos procesos de selección.

Volviendo al caso, las edades no se cansideraran indispensables como si. fue en el Caso 1.

A los datos de las pruebas psicológicas se les asignó

10 puntos excepto a la de potencial que es la considerada más impartante para este puesto, pues la persona debe ser capaz de asumir mayores responsabilidades o de cambiar de puesto sin que esto ocasione caos en la compañía.

En este caso sólo hubo la oportunidad de ingresar al sistema a un grupo exclusivo de cinco candidatas, ya que algunos fueron rechazados instantáneamente después de las pruebas psicológicas, y no llegaron a nuestras manos.

Como resultado se obtuvo dos mejares puestos, can 70 puntos de 70 puntos posibles, seguidos por un tercer candidato con 60 puntos.

Solamente uno de los cinco candidatas fue considerado apto por el sistema, estn se debe a que la mayoría de ellos recién han empezado la Universidad, y no cumplen con el requisito de haber trabajado antes.

Como decíamos antes el primer puesto lo empataron dos candidatos con 70 puntos sobre 70. El cargo se le dio a uno de los dos, básicamente porque su nivel de inglés fue realmente mejor.

Sin embargo la selección no termino del todo allí, ya que una semana después, se presentó un nueva vacante en el mismo departamento. El "Asistente de Costos" previamente contratado pasó a ocupar la nueva vacante, el puesto quedó nuevamente libre. Se volvió a utilizar la misma lista de candidatos para la selección. Esta vez quedaban 4 de los 5, ya que uno fue escogido previamente.

El primer- puesto lo ocupaba un candidato con 70 puntos y el segundo crno con 60 puntos. La diferencia entre los dos fue el resultado de la prueba de trabajo bajo presión.

Se consideró que el que tenía menor puntaje era lógico que estuviese más bajo debido a que no había trabajado nunca y además sólo tenía 19 años de edad.

Se revisó el nivel de inglés de los das primeros candidatos, se los entrevistó por segunda ocasión, la decisión final fue a favor del candidato que se ubicaba en la segunda posición de acuerdo al sistema de selección de personal.

La causa de la decisión no es clara, simplemente el segundo candidato le pareció mejor al jefe de área que hizo

la entrevista.

5.2 MANUAL DEL USUARIO.-

Este manual está dividido en dos partes principales, la primera es el movimiento dentro de los menús, la segunda explica la información que hay en cada pantalla de los menús. El punto 5.2.1 habla de los menús, y los siguientes puntos explican la información que hay en cada menú.

5.2.1 MENUS.-

Los menús del sistema implican la digitación de un número para escoger la opción.

El menú principal. consta de las siguientes opciones:

1 Datos de personas

2 Datos de perfiles

3 Seleccianar

4 Parámetros

5 Utilitarios

0 Salir

Al digitar la opción 1, ingresamos a los datos de personas, y encontramos el siguiente menú:

1 **Ingresar,** Madificar

2 **Eliminar**

3 **Listar**

0 Salir

Este menú se repite para la opción 2 Datos de perfiles del menú principal, ya que el manejo de datos de personas y perfiles de cargos, es muy parecido. Lo que a continuación veamos sobre dato de personas se aplica completamente a datos de perfiles.

La opción ingresar, modificar, permite el ingreso de personas nuevas a la base de datos, y también permite modificar datos de personas que ya se encuentran en la base de datos.

La opción eliminar, permite borrar a una persona de la base de datos.

La opción listar permite ver a todas las personas existentes en la base de datos. Dentro de esta opción tenemos a su vez la opción detalle de personas la cual muestra en detalle a cada una de las personas y permite imprimirlas.

Si **volvemos al menú principal,** nos **quedan** por revisar Seleccionar, Parámetros y Utilitarios.

La opción 3 es Seleccionar. Esta opción es realmente la más importante porque aquí se realiza el proceso de selección. Automáticamente nos aparece un listado de todos los cargos que se han ingresado a la base de dato de cargos. Tenemos que escoger el cargo para el cual nos interesa seleccionar al mejor candidato. Una vez escogido el cargo el programa empieza a trabajar, comparando a cada una de las personas con el perfil del cargo. Hasta que por fin se muestra en pantalla un listado con todas las personas, ordenado por puntajes obtenidos, e indicando si las personas cumplen o no todas las condiciones esperadas. Después de esto, nos queda usar la opción Detalle de Personas la cual nus sirve para visualizar en pantal.la a una de las personas con todas sus características frente al perfil del cargo.

Avancemos a la siguiente opción del menú principal. Parámetros sirve para definir los parámetros de comparación. Al entrar a esta opción se nos abren las siguientes sub-opciones:

1 Crea, Ingresa

2 Listar

3 Eliminar

0 Salir

Crea, Ingresa sirve para crear lus pardmetro desde cero o para insertar nuevos parámetros cuando ya existen algunos.

Listar muestr-a la lista de todos los parámetros y da la oportunidad de imprimir.

Eliminar permite eliminar de la base de datos cualquier parámetro que no queramos que permanezca a la misma.

La última opción del menú principal es Utilitarios. Dentro de Utilitarios tenemos la posibilidad de:

1 Cupiar archivos a disco

2 Copiar archivos de disco

3 Firmador disco

4 DOS

0 Salir

Las opciones de este menú son sumamente claras, 1 permite copiar archivos a un disco A o B, con la idea de

sacar respaldo de la información; 2 permite copiar archivos hacia el disco que estamos utilizando en el momento, dentra del directorio que está el programa. 3 permite firmador discos A y B, y DOS permite al usuario salir al Sistema operativo DOS.

Cuando nos movamos dentro de los menús tendremos en algunas ocasiones el anuncio de Ayuda, el cual presionando F1 nos facilita ver cuales son las posibilidades que tenemos en ese momento para escoger. Sir-ve para no tener que memorizar los números de cédula de cada persona, los nombre exactos de los cargos, o los datos que se han ingresado a las personas, a los cargos o a los parámetros.

5.2.2 PARAMETROS.-

Un parámetro está definido por tres partes. Al ingresar un parámetro hay que inqr-esar las tres partes.

A. Parámetro

B. Carácter que diferencia si el pardmetro es profesional o personal. Si el pardmetro es Profesional se escribe una "P", si es personal se escribe una "E".

C. El signo de comparación del pardmetro, que puede ser cualquiera de los signos entre paréntesis (=, >, <).

Cuando se quiere ingresar un parámetro nuevo el sistema pregunta los datos antes mencionados. Primero pregunta si el dato es personal o profesional, si el dato es per-sonal, ya no hace más pregurtas, **pero** si es profesional el sistema pregunta si el dato es tipo texto o numérico. Al ser tipo texto el sistema asigna signo de comparación igual (=), pero si el dato es numérico, el **sisterna** pregunta cual es el signo de comparación (=, >, <).

La definición de los parámetros está explicada con más detalle en el subtema 4.3.1.

5.2.3 DCSTOS DE PERSONAS.-

Al entrar a ingresar datos de personas, el sistema nos pide el dato y lo graba, no hay que hacer **nada** más. Si queremos modificar el dato el sistema pregunta el dato anterior y luego pide el nuevo dato.

5.2.4 PERFILES DE CARGOS.-

Cada dato de perfil está conformado por:

A. Dato de perfiL

B. Carácter que define si el dato es necesario o no

C. Puntaje

La definición de los perfiles de cargos, está explicada con más detalle en el subtema 4.3.1.

5.2.5 SELECCIONAR.-

En esta sección no se ingresa información, únicamente se selecciona la opción que se requiere. Para obtener los resultados del sistema. Primero se debe escoger el cargo para el cual se desea seleccionar la persona.

Se marca 3 Seleccionar Cargo y se digita el número del cargo que se desea. El programa muestra el listado de las personas ordenadas de acuerdo a su mayor afinidad para el puesto.

Para el detalle de alguna persona se toma 3 Detalle de Personas y se digita de la lista, el número de la persona que se desea revisar en detalle.

El programa muestra el listado de las características de la persona frente a las características del cargo.

5.2.6 UTILITARIOS

En los utilitarios tenemos tres cuatro opciones para seleccionar.

1 Copiar archivos a disco

2 Copiar archivos de disco

3 Firmador disco

4 DOS

La opción 1 sirve para copiar los archivos que contienen las bases de datos a un disco flexible contenido en un "disk drive" A o B.

La opción 2 sirve para trae los archivos de bases de datos desde disco flexibles al directorio corriente.

La opción 3 sirve para firmador discos flexibles.

La opción 4 sirve para salir al sistema operativo DOS.

5.3 FUNCIONAMIENTO.-

El Sistema Experto para Seleccionar Personal clasifica la información que se le ingresa y la analiza de tal forma

que entrega al usuario un resultado objetivo del análisis. Además sirve de base de datos al usuario.

Al programa se le deben definir los parámetros bajo los cuales se va a trabajar. Luego de esta definición se le ingresa la información de cargos y de personas. Cuando el usuario ingresa las opciones de entrada de informacibn, el sistema uti1iza los parámetros previamente definidos para preguntar al usuario la información y el orden en que los debe ingresar.

Una vez que el usuario ha ingresado la información, el sistema está listo para procesar-, y ayudar en la selección.

Pensemos que el usuario ingreso un cargo con todos los datos que el quiere que el candidata cumpla, a que el cargo requiere que se cumplan. Además el usuario ingresó la información de algunos candidatos al puesto. Toca ahora entrar a la opción de selección, donde el programa revisa una a una las características de cada persona y la compara con las caracteristicas del cargo, el programa asigna puntos a las persanas por cada una de las características que se cumplen. Además el sistema analiza si las condiciones tienen que cumplirse con carácter de

obligatorio, y si es el caso y la o las condiciones no se cumplen el sistema marca a la persona como no idónea para el cargo.

Finalmente después del análisis el sistema muestra un listado con todos los candidatos, ordenados de mejor a peor con el puntaje obtenido y con el indicativo de si cumplieron todas las características necesarias o no.

Con la información en pantalla el usuario puede escoger los mejores candidatos, luego escogiendo la opción de detalle, puede revisar porque un candidato es mejor que otro, que es lo que no cumplió un candidato descalificado, que ventajas tiene un candidato con buen puntaje pero descalificado; además puede verificar que los resultados del análisis han sido correctos.

De esta forma el usuario tiene en sus manos toda la información correcta para tomar la mejor decisión.

5.4 APLICACIONES.-

La aplicación que tiene el sistema es como herramienta de ayuda para las per-sonas tienen que escoger entre un grupo de candidatos a un cargo.

El sistema tiene múltiples ventajas, como son, la flexibilidad para corregir definiciones equivocadas que se pueden percibir al ver los resultados. Aunque se corrija cualquier definición, o se aumentando la información, el sistema vuelve a estar listo para emitir un nuevo resultado.

Es flexible permitiendo que cada usuario organice su formato de trabajo, que ordene los parámetros como le parezca mejor, y que asigne los puntajes que según su criteria.

Se pueden visualizar a todos los candidatos ordenadas de mejor a peor, y revisar el detalle de cada uno de ellos. En cuestión de segundos se puede visualizar resultados de los candidatos con las ventajas y desventajas de cada uno.

Una vez que la información de una persona quedó ingresada al sistema, no es necesario volver a digitarla cuando se presenten nuevas vacantes.

Las personas que ya fueron seleccionadas para un cargo pueden aparecer como mejores candidatos para los nuevos cargos, esto permite la rotación en la empresa.

CONCLUSIONES Y RECOMENDACIONES

¿ Qué podemos decir del trabajo realizado ? ¿Qué quedó después de tantas horas de esfuerzo, tantos días sin distracción, y tardes aburridas ?

Lo que quedó es el comienzo de un camino muy largo que apenas hemos empezado a recorrer. El camino de la ciencia y de la investigacidn en el área de la inteliqencia artificial y de los sistemas expertos.

El trabajo es pequeño, es cierto pero deja la pauta par-a que muchos otros pisen este camina can algunos pasos ya dados. La investigación no se hace de la noche a la mañana ni la hace un sólo hombre, la hace la sociedad científica en el paso de 10s siglos.

Después de este trabajo de investigación nos queda una investigación y una explicacidn sobre lo que existe en inteligencia artificial y en sistemas expertas.

Esta es un área de la ciencia que debemos aprovechar, difundirla entre nuestra gente para estar al día en los cambios tecnológicos y en los pasos que da el mundo, para

dar nosotros los pasos junto al mundo, para ser nosotros, una generación que haga cambios en nuestro país.

Conocer y desarrollar sistemas expertos es algo interesantísimo, y nos ha dado criterio para pensar en como aplicar la inteligencia artificial aunque sea en pequeños sistemas a nuestro trabajo en las empresas.

En cuanto a los lenguajes de inteligencia artificial, recomendamos hacer una investigación más detallada sobre cuales son los mejores para trabajar can ellos. El lenguaje utilizado en este proyecto fue Turbo Prolog, el cual objetivamente no lo recomendamos como lenguaje de trabajo, aunque quizAs sea útil como lenguaje educativo. El Turbo Prolog es muy bueno para el momento preciso de la resolución de problemas, es más, es excelente, pero un sistema no es sólo la operación de resolución. En este sistema hemos tenido muchos problemas en la utilización de memoria, en las interfaces con el usuario, en la manipulación de información. Es probable que versiones de Prolog de otras marcas hayan seguido desarrollando el lenguaje, y que todos los problemas que hemos tenido se los pueda superar con facilidad.

Un lenguaje que recomendamos analizar es el Lisp, el cual

también tiene una tradición de ser muy bueno. Lamentablemente no podemos abarcarlo todo rn un solo proyecto, y ria hemos alcanzado a cubrir todas estas Areas que hubiesen sidn muy interesantes.

Recomendamos desarrollar una versión del Sistema Experto para Seleccionar Personal con una combinación de lenguajes para manejar la interfaz con el usuario y con Prolog para resolver la parte inteligente del sistema debe ser el siguiente paso en la investigación de esta materia. Afortunadamente esta iniciativa ya fue tomada por nuestro director de tesis, y el trabajo ya se está realizando en la Espol.

APÉNDICE A

RESPALDOS DEL CASO NÚMERO 1

NOMBRE DEL CARGO: EJECUTIVO DE CUENTAS

PARAMETROS	DATO DE PERFIL	NECESARIO	PUNTAJE
01.-EDAD MINIMA	: 24	[S]	[00]
02.-EDAD MAXIMA	: 31	[S]	[00]
03.-SEXO	: -	[N]	[00]
04.-ESTUDIOS	: ADM. DE EMPRESAS	[S]	[20]
05.-ESTUDIOS	: PSICOLOGIA	[S]	[20]
06.-ESTUDIOS	: MERCADOTECNIA	[S]	[20]
07.-ANIOS DE ESTUDIOS	:	[N]	[01]
08.-TECNOLOGIA (S/N)	: S	[N]	[01]
09.-INGENIERIA (S/N)	: S	[N]	[05]
10.-GRADUADO (S/N)	: S	[N]	[03]
11.-ESPECIALIZACION	: -	[N]	[00]
12.-IDIOMA INGLES	: N	[N]	[00]
13.-COMP. LOTUS	: S	[N]	[02]
14.-COMP. WP	: S	[N]	[02]
15.-EXPERIENCIA	: -	[N]	[00]
16.-ANIOS DE EXPERIENCIA	: -	[N]	[00]

SIS'TEMA EXPERTO PARA SELECCION **DE** PERSONAL

CEDULA	NOMBRE	APELLIDO	PUNT.	STATUS
01.-[0912147196]	[E	] [S	] [28]	no
02.-[0909562654]	[J	] [C	] [23]	no
03.-[0910708148]	[R	] [E	] [22]	ok
04.-[0911842086]	[H	] [R	] [20]	ok
05.-[1202348411]	[G	] [T	] [9]	no
06.-[0300718624]	[A	] [N	] [8]	no
07.-[0910770163]	[N	] [P	] [8]	no
08.-[0910224104]	[E	] [D	] [7]	no
09.-[0910744077]	[J	] [R	] [7]	no
10.-[0909434250]	[A	] [D	] [5]	no

1 Pantalla Siguiente
2 Pantalla Anterior
3 Detalle de Personas
0 Menu

[ ]

NOMBRE DEL CARGO: EJECUTIVO DE CUENTAS
NUMERO DE CEDULA 0912147196

PARAMETROS	DATO DEL PERFIL.	DATO DE LA PERSONA
01.-NOMBRE	[ ]	[E] [0]
02.-APELLIDO	[ ]	[S] [01
03.-TELEFONO	[ ]	[351049] [0]
04.-EDAD MINIMA	[24]	[22] [01
05.-EDAD MAXIMA	[31]	[22] [0I
06.-SEXO	[-]	[M 3 [0]
07.-ESTUDIOS	[ADM. DE EMPRESAS]	[MERCADOTECNIA I [0]
08.-ESTUDIOS	[PSICOLOGIA]	[ ] [0]
09.-ESTUDIOS	[MERCADOTECNIA]	[ ] [20]
10.-ANIOS DE ESTUDIOS	[ ]	[4 3 [0I
11.-TECNOLOGIA (S/N)	[S]	[N] C 0]
12.-INGENIERIA (S/N)	[S]	[S I [5]
13.-GRADUADO (S/N)	[S]	[S I [3]
14.-ESPECIALIZACION	[-]	[I [0]
15.-IDIOMA INGLES	[N]	[S 3 [0]
16.-COMP. LOTUS	[S]	[N I [0]
17.-COMP. WP	[S]	[N I C 0]
18.-EXPERIENCIA	[-]	[COMERCIO EXTERIOR I [0]
19.-ANIOS DE EXPERIENCIA	[-]	[1 I [**03**

NOMBRE DEL CARGO: EJECUTIVO DE CUENTAS
NUMERO DE CEDULA 0909562654

PARAMETROS	DATO DEL PERFIL	DATO DE LA PERSONA	
01.-NOMBRE	[	] [J	1 1 0]
02.-APELLIDO	[	] [C	1 [0]
03.-TELEFONO	[	] [399434	] [01
04.-EDAD MINIMA	[24	] [23	] [0]
05.-EDAD MAXIMA	[31	] [23	1 [0]
06.-SEXO	[-	] [M	1 [01
07.-ESTUDIOS	[ADM. DE EMPRESAS	] [MERCADOTECNIA	] C 01
08.-ESTUDIOS	[PSICOLOGIA	] [	] C 01
09.-ESTUDIOS	[MERCADOTECNIA	] [	] C 203
10.-ANIOS DE ESTUDIOS	[	] [1	3 C 03
11.-TECNOLOGIA (S/N)	[S	] [S	1 C 1]
12.-INGENIERIA (S/N)	[S	] [N	] C 0]
13.-GRADUADO (S/N)	[S	] [N	] [**03**
14.-ESPECIALIZACION	[-	] [	] C 0]
15.-IDIOMA INGLES	[N	] [S	1 [0]
16.-COMP. LOTUS	[S	] [S	1 [2]
17.-COMP. WP	[S	] [N	I [0]
18.-EXPERIENCIA	[-	] [	1 [0I
19.-ANIOS DE EXPERIENCIA	[-	] [4	1 [0]

APÉNDICE B

RESPALDOS DEL CASO NÚMERO 2

NOMBRE DEL CARGO: ASISTENTE DE COSTOS

PARAMETROS	DATO DE PERFIL	NECESARIO	PUNTAJE
01.-EDAD (MIN)	: 20	[N]	[00]
02.-EDAD (MAX)	: 25	[N]	[00]
03.-IDIOMA	: INGLES	[S]	[20]
04.-IDIOMA	: -	[S]	[00]
05.-EXPERIENCIA	:	[N]	[00]
06.-RAZONAMIENTO VERBAL	: 79	[N]	[10]
07.-TRABAJO A PRESION	: 79	[N]	[10]
08.-POTENCIAL	: 79	[N]	[20]
09.-HABILIDAD NUMERICA	: 79	[N]	[10]
10.-HA TRABAJADO (S/N)	: S	[S]	[00]

SISTEMA EXPERTO PARA SELECCION DE PERSONAL

CEDULA	NOMBRE	APELLIDO	PUNT.	STATUS
01.-[0000000000]	[I]	[A]	[70]	no
02.-[1304049818]	[J]	[N]	[70]	no
03.-[0908982747]	[F]	[F]	[60]	no
04.-[0913966255]	[R]	[Q]	[50]	ok
05.-[0914432919]	[CH]	[C]	[40]	no
06.-[0912037884]	[F]	[C]	[20]	no

3 Pantalla Siguiente
2 Pantalla Anterior
3 Detalle de **Personas**
0 Menu

[ ]

NOMBRE DEL CARGO: ASISTENTE DE COSTOS
NUMERO DE CEDULA 0000000000

PARAMETROS	DATO DEL PERFIL	DATO DE LA PERSONA
01.-NOMBRE	C	1 [I I [0]
02.-APELLIDO	C	I [A 1 [0]
03.-EDAD (MIN)	[20	I [20] [01
04.-EDAD (MAX)	[25	I [20] [0I
05.-IDIOMA	[INGLES	1 [INGLES I C 20]
06.-IDIOMA	[-	] [I [0I
07.-EXPERIENCIA	C	] [I [01
08.-RAZONAMIENTO VERBAL	[79	I [90 I C 10]
09.-TRABAJO A PRESION	C79	1 [90] C 10]
10.-POTENCIAL	[79	I [90] [20]
11.-HABILIDAD NUMERICA	[79	1 [90] C 10]
12.-HA TRABAJADO (S/N)	[S	I [N I [01
13.-SUELDO WUE ASPIRA	C	1 [250.000 I C 0]

NOMBRE DEL CARGO: ASISTENTE DE COSTOS
NUMERO DE CEDULA 1304049818

PARAMETROS	DATO DEL PERFIL	DATO DE LA PERSONA
01.-NOMBRE	[	1 [J] [01
02.-APELLIDO	C	] [N 1 [0]
03.-EDAD (MIN)	[20	] [22 I [0]
04.-EDAD (MAX)	[25	1 [22 1 [01
05.-IDIOMA	[INGLES	] [1 C 201
06.-IDIOMA	[-	] [INGLES 1 [0]
07.-EXPERIENCIA	C	] [I C 0]
08.-RAZONAMIENTO VERBAL	[79	1 c99 1 [10]
09.-TRABAJO A PRESION	[79	1 [99 1 [10]
10.-POTENCIAL	[79	I [80 I C 201
11.-HABILIDAD NUMERICA	[79	1 [99 1 C 10]
12.-HA TRABAJADO (S/N)	CS	1 [N] C 0]
13.-SUELDO QUE ASPIRA	[	1 [ ] [01

HOMBRE DEL CARGO: ASISTENTE DE COSTOS
NUMERO DE CEDULA *0908982747*

PARAMETROS	DATO DEL PERFIL	DATO DE LA PERSONA
01.-NOMBRE	C	1 [F I [0]
02.-APELLIDO	[	1 [F] [0]
03.-EDAD (MIN)	c20	I [19 I [01
04.-EDAD (MAX)	c25	I [19 1 [0]
05.-IDIOMA	[INGLES	] [ ] [201
06.-IDIOMA	[-	I [INGLES I C 0]
07.-EXPERIENCIA	C	] [1 C 0]
08.-RAZONAMIENTO VERBAL	[79	] [99 1 [10]
09.-TRABAJO A PRESION	[79	] *c27* I [01
10.-POTENCIAL	[79	1 [90 1 C 20]
11.-HABILIDAD NUMERICA	[79	1 [99 1 [10]
12.-HA TRABAJADO (S/N)	[S	I [N 1 C 0]
13.-SUELDO QUE ASPIRA	C	1 [S/.400.000 1 [0]

APÉNDICE C

FUNDAMENTOS DE LENGUAJE PROLOG

I. DEFINICIÓN DE PROLOG.-

Prolog (PROgramación LOGica) es un lenguaje declarativo que permite la representación de datos por medio de la lógica simbólica. En este tipo de lenguaje, los programas usan razonamiento deductivo para resolver los problemas: dados una serie hechos y reglas que determinan interrelaciones, Prolog puede responder preguntas sin que el usuario o el programador deba preocuparse de cómo lo hace.

Basic, Fortran, Pascal, C y todos los otros lenguajes tradicionales de programación son "procedurales", el programador debe proveer un conjunto de instrucciones que indiquen paso a paso como se puede llegar a1 resultado final. Prolog es más simple: sólo se requiere describir el problema (la meta) y las reglas globales para resolver éste. Luego, Prolog automáticamente deter-mina el mejor método para implementar esta meta. Los programas en Turbo Prolog pueden tipicamente ser unas 10 veces más cortos que un programa en Basic o Pascal que haga lo mismo.

Prolog es un Lenguaje de Inteligencia Artificial.

II. PROLOG COMO LENGUAJE DE INTELIGENCIA ARTIFICIAL.-

Los computadores que utilizan la Inteligencia Artificial operan en un modo "relacional", similar a la forma como lo hacen nuestros cerebros y no en la forma congelada del formato de direcciones discretas de memoria. Pensar relacionalmente es decir: si digo la palabra "rojo", pienso en objetos y condiciones que se asocian al concepto de "ser rojo": tomates, corazones, manzanas, etc. Teniendo la clase general, se puede fácil y rápidamente proceder de ahí y entrar en el contexto de otros parámetros de búsqueda dados.

Las técnicas de Inteligencia Artificial están interesadas en dos Areas principales de la tecnología de la computación: la interfaz de lenguaje natural y el sentido visual del mundo. Esta segunda Area, la interpretación visual del computador de lo que ocurre a su alrededor, todavía necesita mucho estudio en el futuro.

El reconocimiento del lenguaje natural es la habilidad de un programa de aceptar y entender exitosamente las sentencias diarias de nuestro lenguaje. No se necesitaría sintaxis especial para escribir estructuras de comandos. Prolog puede responder a preguntas como: "Qué provincias

están en la Costa del Ecuador?" y responder "Esmeraldas, Manabí, Guayas, El Oro". Responderá lo mismo con variaciones de la pregunta como: "Listar las provincias en la Costa del Ecuador" a "Provincias en Ia Costa del Ecuador'?" o "provincias, costa_Ecu?"

La habilidad de entender lenguaje natural provee también tolerancia a los errores. En los lenguajes procedurales cada paso que lleva a cabo el computador necesita una secuencia de sintaxis exacta y datos bajo un formato preestablecido. Sin embargo, tanto los programadores como los usuarios pueden cometer errores. El software debe Ejer capaz de aceptar flexiblemente información, siendo esta cercana en escritura y significado.

Otra meta de la Inteligencia Artificial es la implementación de Sistemas Expertos.

III. PROLOG COMO LENGUAJE PARA SISTEMAS EXPERTOS.-

Los datos que usamos en nuestra vida diaria son claros y certeros. Tenemos la habilidad de preguntar y obtener, de profesionales que dominan diversos campos, respuestas expertas a preguntas específicas. Con un programa adecuado, el experto humano puede r-espander cina serie de

preguntas, el software las almacena, y luego, cuando se le hagan preguntas al sistema, él responderá tama si el experto estuviera ahí.

Los expertos humanos pesan sus opiniones, alguno puede estar 60 % seguro que si ciertas condiciones están presentes, un resultado específico tendrá una mejor oportunidad promedio de ocurrir. Preguntando y obteniendo respuestas a una serie de preguntas relacionadas, un sistema experto puede proveer un mejor porcentaje de seguridad.

Construir un sistema experto viable tama más que simple programación. Cualquier experto, humano o cibernético, es sólo tan bueno como lo sea la suma de su conocimiento. El sistema consiste de reglas. Un doctor examinando a su paciente aplica reglas, usualmente sin siquiera pensar en ellas: "Temperatura de 41? Este hombre está enfermo, debo continuar. Esta tosiendo. Usaré mis estetoscopio, encontraré que su pecho está congestionado. ¿ Neumonía ?, ¿ sólo un enfriamiento ?, ¿ síntomas de gripe?" El doctor entonces aplica otras reglas para posibilidades más exactas, hasta resolver razonablemente el problema. "Usted tiene gripe, tenga la siguiente receta". Un programa experto médico puede hacer In mismo.

Diagnósticos mucho más complicados están siendo manejados hoy por sistemas expertos médicos y de otros campos.

Las características del lenguaje Prolog, como veremos a continuación, lo ubican como una herramienta importante para la implementación de sistemas expertos.

IV. COMPARACIÓN PROLOG VS BASIC.-

Como ejemplo, veamos rápidamente cómo un programa en Prolog se ve y actúa, en comparación con su equivalente en Basic. El programa se basa en las premisas "Juan es amigo de María" y "Bill es amigo de las amigas de Juan". Hacer que el programa responda a la pregunta "De quién es amigo Bill?", puede ser muy complicado en los lenguajes tradicionales, imperativos o procedurales, pues éstos están atados al hardware. En ellos, cada paso que el computador da para determinar la respuesta a una pregunta, debe ser deletreado con detalle.

```
10  REM *****  Prograsa en Basic para encontrar Quién es
    amiga de Bill  *****  Escrito por Ralph Roberts  *****
30  VATA CHARLIE ES AMIGO DE ELLEN
40  DATA FARLEY ES AMIGO DE WANDA
50  DATA RHETT ES AMIGO DE HISS SCARLET
60  DATA MARVIN ES AMIGO DE PEARL
70  DATA JUAN ES AMIGO DE MARIA
PO  FOR X=1 TO 5:  REAV A$(X):  NEXT X
90  INPUT "Name"; B$
100 IF B$="BILL" THEN LET B$=@JUAN" ELSE GOTO 210
```

```
110 PRINT
120 FOR X=l  TO 5
130      C$=""
140      FUR Y=l TO LEN(A$(X))
150           IF MID$(A$(X),Y,1)=" " THEN 190
160           C$=C$ + MID$(A$(X),Y,1)
170      NEXT Y
180      I F C$=B$ THEN 220
190 NEXT X
200 PRINT "Datos insuficientes": PRINT: GOTO 230
210 PRINT "BILL"; MID$(A$(X), Y, LEN(A$(X))-Y+1)
220 END
```

Para responder a la pregunta "Quién es **amigo** de Bill, si Bill es amigo de quien Juan es amigo'?", **BASIC** establece los datos, los lee en un arreglo, crea un lazo para comparar y determinar quién es amigo de Juan, y luego responde la pregunta. Los comandos MID$ y LEN muestran que el programa está ganando complejidad. Si se añaden más reglas y más datos, el programa crecerá en cientos de líneas. Además, nuestro programa en BASIC no puede responder otra pregunta relacionada a los datos. Si preguntamos de quién es amigo Rhett o Marvin, obtendremos la repuesta "Datos insuficientes".

```
/*  Programa en Prolog */

domains
     person, lady = symbol

predicates
     amigo(person, lady)

clauses
     amigo(charlie, ellen).
```

```
amigo(farley, wanda).
amigo(rhett,miss_scarlett).
amigo(marvin, pearl).
amigo(juan, maria).
amigo(bill, x) if amigo(juan, x).
```

En Prolog el programa es más corto y poderoso. No hay comandos para representar pracedimientas, no hay instrucciones para leer variables o comparar datos. Prolog sabe cómo hacer todo, por sí solo. Nosotros sólo debemos declarar reglas y hechos. Más adelante explicaremos las secciones de un programa en Prolog.

Al correr este programa una ventana de diálogo nos permitirá realizar diferentes preguntas:

- Si escribimos amigo(bill, Quien), nos responderán Quien=maria.
- Si preguntamos amigo(bill,maria), nos dirán True.
- Si decimos amigo(Quien,maria), se encontrarán dos repuestas: Quien=juan y Quien=bill.
- Si preguntamos amigo(Quien,Chica), obtendremos una lista de relaciones como Quien=charlie, Chica=ellen.

Este ejemplo no significa que los lenguajes de quinta generación reemplazarán a los lenguajes procedurales.

Herramientas diferentes sun apropiadas para trabajos dife-. rentes.

V. LA PROGRAMACIÓN LOGICA.-

La programación lógica, propia de lenguajes como Prolog, depende fuertemente del concepto de "razonamiento hacia atrás" o "backtracking". "Backtr-acking" puede simplificarse a decir "A es verdadero si B y C existen". Uti1izando esta técnica, las reglas declaradas son ejecutadas baja exactas invocaciones procedura1es usadas en los viejos lenguajes. Así pues, a través del "backtracking", Prolog puede ser también usado como un lenguaje puramente procedura1.

El termino "procedural" se acostumbra aplicar a los viejos lenguajes, pero para Prolog tiene un significado diferente. En BASIC, Pascal, C y otros, nosotros debemos especificar no sólo cómo los problemas deben ser resueltos, sino también la forma en que el computador lo hace, paso por paso. Una implementación procedural en Prolog sólo se refiere a detallar el orden en el cual el problema es resuelto, peru no las instrucciones de máquina.

A diferencia de otros programas, que constituyen esencialmente una colección de funciones, los programas en Prolog son una secuencia de relaciones, reglas y hechos sobre un tema, constituyendo todo esto una "Base de Datos" que puede ser consultada o ingresada. Por medio del siguiente ejemplo ilustraremos el concepto de "backtracking".

```
domains
     prolog, persona = symbol

predicates
     gusta(persona, prolog)
     explica(persona, prolog)

clauses
     gusta(ralph, prolog)
     gusta(lector, prolog) if explica(ralph, prolog)
     explica(ralph, prolog)
```

En lenguaje coloquial, esto quiere decir que a Ralph le gusta el Prolog, que al lector le gustará el Prolog si lo explica Ralph, y que Ralph explica Prolog.

Cuando este programa es corrido, aparece en pantalla una ventana e ingresamos la pregunta **gusta(Quien, prolog).** La respuesta es **Quien=ralph** y **Quien=lector.** Las variables empiezan can mayúscula y existe tatal. 1ibertad par-a escoger sus nombres.

Cómo consigue Prolog estos resultados?. No hemos especificado ningún procedimiento, sólo hemos declarados hechos y reglas. Prolog examina el último hecho **explica (ralph, prolog)** y determina que éste no tiene condiciones, sino que es válido por si sólo. Como **este** único hecho no satisface la pregunta, se **analiza** la regla anterior que **dice gusta(lector, prolog) if explica(ralph, prolog).** En **este caso sí encontramos el** formato standard de conclusión-condición. Prolog va **nuevamente de atrás hacia** adelante, viendo qué está después del **if** y se da cuenta que ese hecho ya lo conoce. Así pues **la** conclusión **gusta(lector, prolog)** se resuelve como verdadera y se obtiene la respuesta **Quien= lector.**

A continuación, Prolog continúa revisando hacia atrás para ver si encuentra más respuestas. El hecho anterior nos dice **gusta(ralph, prolog)** y se encuentra así la segunda respuesta.

Esencialmente, el proceso de "backtracking" retrocede desde la conclusión de un comando de la forma "conclusión si condición", reduciendo problemas en subproblemas que cumplen con condiciones dadas hasta el momento.

VI. OBJETOS Y RELACIONES.-

Los bloque básicos de construcción del Prolog son las relaciones, específicamente relaciones entre objetos. Un objeto es cualquier cosa que puede ser representada simbólicamente en los programas. Ej: casa, banana, amor. Todos estos, aunque sean numbre propios, empiezan con minúscula. Empezar una palabra con mayúscula significa que es una variable, y no un objeto. Para especificar relaciones la sintaxis es **relación(objeto).** La relación es conocida como el "predicado" y el objeto como el "argumento".

VII. ESTRUCTURA DEL PROGRAMA.-

Un programa en Prolog está dividido en 4 secciones: **domains, predicates, goals y clauses.**

```
domains
      nombre = symbol

predicates
      persona(nombre)

goal
      person(Quien),
      write(Quien), nl,
      fail.

clauses
      persona(maria)
      persona(jorge)
```

En la sección **domains** se declaran los objetos. Prolog quiere primero saber sobre los tipos de argumentos que van a usar 1os predi.cados.

La segunda sección de un programa en Prolog, **predicates,** contiene las definiciones de los predicados, los cuales representan relaciones entre los objetos definidos en la sección domains.

La sección de **clauses** tiene hechos y reglas basadas en las relaciones y los objetos. Los argumentos son símbolos, no variables y por lo tanto deben estar en minúscula.

Opcionalmente se encuentra la sección **goal.** De no encontrarse, se presentará en pantalla una ventana donde se pueden realizar consultas **a** la **"Base** de Datos". En nuestro ejemplo la manera de ejecutar "backtracking" y extraer todas las respuestas es mediante el comando **fail.** Este for-za a Prolog a encontrar todas las soluciones y sólo parar cuando ya nadie mds cumpla la pregunta.

REFERENCIAS BIBLIOGRAFICAS

J.P.Sánchez y Beltrán. SISTEMAS EXPERTOS, UNA METODOLOGÍA DE PROGRAMACIÓN. Macrobit, 1990.

Benchimol-Levine-Pomerol. LOS SISTEMAS EXPERTOS EN LA EMPRESA. Macrobit, 1990.

Peter Jackson. INTRODUCTION TO EXPERT SYSTEMS. Addison-Wesley, 1986.

Borland International. TURBO PROLOG 2.0 USER'S GUIDE. Borland, 1988.

Borland International. TURBO PROLOG 2.0 REFERENCE GUIDE. Borland, 1988.