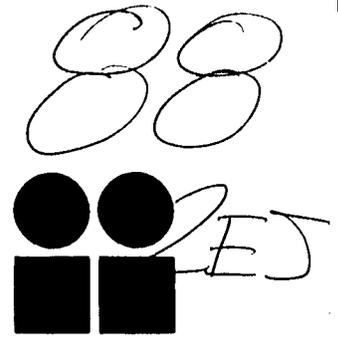




**UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO**



**Instituto de Ingeniería**

**FACULTAD DE INGENIERIA  
DIVISION DE INGENIERIA ELECTRICA-ELECTRONICA Y  
EN COMPUTACION**

**SISTEMA EXPERTO TUTORIAL DE REDES  
NEURONALES ARTIFICIALES**

**T E S I S**

**QUE PARA OBTENER EL TITULO DE**

**INGENIERO EN COMPUTACION**

**P R E S E N T A N :  
MA. ISABEL RAMIREZ PATRACA**



**DIRECTOR DE TESIS: Dr. Felipe Lara Rosano**

**México, D. F.**

**Octubre 1995**

**FALLA DE ORIGEN**

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



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

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

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

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

# **SISTEMA EXPERTO TUTORIAL DE REDES NEURONALES ARTIFICIALES**

**ARRIAGA COLINA ROSALBA  
RAMIREZ PATRACA MA. ISABEL  
VILLANUEVA RODRIGUEZ BEATRIZ**

El presente trabajo no se hubiera realizado sin la ayuda incondicional de nuestro Dir. de Tesis el Dr. Felipe Lara R., quién nos brindó sus valiosos conocimientos y apoyo en todo momento. Agradecemos también al Instituto de Ingeniería por permitirnos el uso de sus instalaciones y equipos, así como los que en él laboran, destacando el compañerismo y colaboración del Mto. Nicolás Kemper.

Nos es grato también mencionar que la Facultad de Ingeniería ha sido una etapa más en nuestra vida, y que todos los cursos, exámenes, trabajos y experiencias como estudiantes que de ella recibimos jamás se borrarán.

Gracias a la Universidad Nacional Autónoma de México por permitirnos llegar a una meta más en nuestra vida.

**Con Admiración y Respeto**

**Beatriz, Rosalba y Ma. Isabel**

El sueño de ser siempre el orgullo de quienes me dieron la vida, no deja de darme el aliento que alimenta mi Alma y el deseo de existir.

Por que siempre los amaré  
Gracias a mis PADRES.  
Ma. Abad y Amado

Por que en tí nunca dejará de existir una sonrisa, y una palabra de aliento en cualquier momento, y por el gran amor que te tengo.

Gracias a mi hermana y amiga VIOLETA.

El espíritu que me impulsa a superarme, es el mismo que te ama y te da las Gracias por tu apoyo y confianza como mi amigo, como mi novio y como mi ESPOSO.

Por que sé, que siempre contaré contigo.  
TE AMO DAVID.

Rosalba Arriaga Colina.

**A mis Papas Manuel e Irma, por haberme apoyado a lo largo de mis estudios y haber logrado la meta que me propuse.**

**A mis hermanas Paty, Carmen y Rosy.**

**A mi novio Jorge A. por su apoyo incondicional que me brindo en todo momento, con todo mi amor Isabel.**

**Ma. Isabel Ramírez Patraca.**

### **A MIS PADRES**

Ahora que he llegado a concluir parte de mi formación profesional, quiero agradecer y dedicarles este trabajo como testimonio de mi gran amor. Así como su apoyo y comprensión en todo momento sin pedir nada a cambio.

Gracias por su dedicación y empeño, ya que son la fuente de mi inspiración para continuar y ser cada día mejor tanto como hija, hermana y estudiante. Para que sus esfuerzos y sacrificios no fueran en vano.

“ La única energía que fortalece verdaderamente al hogar y a cada uno de los miembros es el Amor que siempre nos han brindado”.

Siempre están en mi corazón Eduardo y Lupita  
**GRACIAS**

### **A MIS HERMANOS**

Por sus amistad, cariño y sus buenos consejos que en todo momento han dado, me siento orgullosa de tener una familia unida y que en ella estén ustedes que siempre me brindaron su confianza.

Con cariño a todos **GRACIAS**

### **A MI ESPOSO**

Por tu gran amor y valioso apoyo en todo momento, que me inspiraron a seguir adelante son lo más hermoso, **GRACIAS**.

**TE AMO Gustavo**

**Beatriz Villanueva Rodríguez.**

## INDICE

### Introducción.

<b>1.- Inteligencia Artificial y Sistemas Expertos</b>	
1.1 Bosquejo Histórico	1
1.2 ¿Qué es la Inteligencia Artificial?	1
1.3 Campo de Estudio	2
1.4 Lenguajes y Herramientas de la Inteligencia Artificial	3
1.5 Sistemas Expertos	4
1.5.1 Definiciones del Sistema Experto	5
1.5.2 Ventajas de los Sistemas Expertos	6
1.5.3 Los Sistemas Expertos y su Entorno	8
1.5.4 Tipos de Sistemas Expertos	9
1.5.5 Arquitectura de un Sistema Experto	12
1.5.6 Desarrollo de un Sistema Experto	14
1.5.7 Análisis del Problema	15
1.5.8 El Ingeniero del Conocimiento	15
1.6 Mecanismos de Razonamiento	17
1.6.1 Elección de una Dirección de Solución	18
1.7 Sistemas Expertos Tutoriales	19
1.7.1 Tipos de Sistemas Expertos Tutoriales	21
1.7.1.1 Tutores con Iniciativa	21
1.7.1.2 Tutores de Entrenamiento	21
1.7.1.3 Tutores de Diagnóstico	22
1.7.1.4 El Concepto de Micromundos	22
1.7.2 Sistemas Expertos	22
1.7.3 Desarrollo de los Sistemas Expertos Tutoriales	23
<b>2.- Aspectos Pedagógicos</b>	
2.1 Introducción	26
2.2 Aspectos Pedagógicos	26
2.3 Enseñanza Tradicional	27
2.3.1 El Pensamiento Esponja	28
2.3.2 Ambiente Educativo	29
2.3.3 Desventajas de la Enseñanza Tradicional	30
2.4 Educación e Informática	30
2.4.1 Aprendizaje por Computadora	31
2.4.2 La Computadora como Vehículo para el Aprendizaje	32
2.4.3 ¿Para que Software Educativo?	33

<b>3.- Redes Neuronales Artificiales</b>	
<b>3.1 Redes Neuronales Artificiales Elementales</b>	<b>35</b>
3.1.1 Bosquejo Historico	35
3.1.2 Definiciones	36
3.1.3 La Neurona Artificial	37
3.1.4 El Perceptrón	38
3.1.5 La Adaline	41
3.1.6 La Madaline	43
<b>3.2 Modelos de Aprendizaje</b>	<b>44</b>
3.2.1 Tipos de Aprendizaje	44
3.2.2 Aprendizaje por Retropropagación de error	46
<b>3.3 Desarrollo de Redes Neuronales Artificiales</b>	<b>49</b>
<b>3.4 Simulación de Redes Neuronales Artificiales con BrainMaker</b>	<b>55</b>
3.4.1 Introducción	55
3.4.2 Aplicación de Reconocimiento de Patrones	58
3.4.2.1 Reconocimiento Optico de Caracteres Numéricos	58
<b>3.5 Conceptos Básicos de Electrónica para la implementación de Redes Neuronales Artificiales.</b>	<b>60</b>
<b>4.-Desarrollo del Sistema Experto Tutorial SETURNA.</b>	
<b>4.1 Desarrollo y características de SETURNA</b>	<b>67</b>
4.1.1 Alcance de la Aplicación	67
4.1.2 Adquisición del Conocimiento	67
4.1.3 Herramientas de Desarrollo	68
4.1.4 Arquitectura de SETURNA	68
4.1.5 Programas de Enlace	71
4.1.6 Comunicación con el Usuario	71
4.1.7 Requerimientos	71
4.1.8 Operación de SETURNA	71
<b>5.-Conclusiones</b>	
<b>Bibliografía</b>	
<b>Apendice A</b>	
<b>Apendice B</b>	

---

---

## INTRODUCCION

El cerebro humano está compuesto por células especiales llamadas neuronas. Las neuronas son especiales, por que cuando ellas mueren no se reproducen ni se reemplazan como todas las demás células. Este fenómeno puede explicarse porque las neuronas guardan o retienen información. Se estima que el número de neuronas en el cerebro humano cubre un rango arriba de 100 billones de neuronas y allí hay más de cientos de diferentes tipos de neuronas. Las neuronas son separadas dentro de grupos llamados Redes Neuronales . Cada Red Neuronal contiene varios miles de neuronas que son altamente interconectadas. Por esto el cerebro puede ser visto como una colección de Redes Neuronales.

Una Red Neuronal Artificial (RNA) es un modelo que emula a una Red Neuronal Biológica. Los conceptos son usados para implementar la solución por software de procesos masivos en paralelo que involucra elementos de proceso (también llamados neuronas artificiales o neuronodos) interconectadas en una arquitectura de red.

Extensivas investigaciones de la neurobiología y la sicología han desarrollado grandes avances, pero aún, hay bastantes cuestiones acerca de como el cerebro y la mente trabajan. Esta es justamente una de las razones por la cual los modelos de neurocomputadoras no son muy parecidos a los sistemas biológicos. Sin embargo, investigaciones y desarrollos en el área de RNAs están produciendo interesantes y útiles sistemas que tratan de imitar las características de los sistemas biológicos.

En los últimos años ha habido un desarrollo explosivo en la investigación básica y aplicada alrededor de las Redes Neuronales Artificiales. Estos sistemas, también llamados *sistemas de procesamiento distribuido en paralelo, sistemas conexionistas, redes neuronales y neurocomputadoras*, constituyen la base de la sexta Generación de computadoras anunciada ya por los japoneses.

Grandes cantidades de métodos y algoritmos en máquinas de aprendizaje han sido desarrollados. Más de ellos aún están en investigación en los laboratorios. Aquí damos algunos ejemplos:

- °**Neurocomputadoras:** Esta aproximación o método puede ser usado para adquisición de conocimiento y para inferencia.
- °**Aprendizaje inductivo:** Este método es usado en la adquisición de conocimiento .
- °**Razonamiento en caso-basado y razonamiento analógico:** Este método es utilizado en la adquisición de conocimiento y en inferencia.
- °**Métodos estadísticos:** Aunque si bien mas conveniente en situaciones de el conocimiento pobre, los métodos estadísticos han sido usados en adquisición de conocimiento y solución de problemas.

---

---

° **Aprendizaje basado en explicación:** Esta aproximación, asume que existiendo bastante teoría de racionalización de porque una muestra es o no miembro prototipo de una clase. Este método es prometedor pero aún no ha sido utilizado en el mundo real.

Un método diferente de sistemas inteligentes indujo la construcción de computadoras basadas en procesamiento en paralelo, rápida recuperación de grandes cantidades de información , y la habilidad de reconocimiento de patrones basado sobre la experiencia. La tecnología que permite llevar a cabo estos resultados es llamada *Neurocomputadoras o Redes Neuronales Artificiales (RNA)*.

Las Redes Neuronales Artificiales son una tecnología de procesamiento de información inspirada por estudios del cerebro y sistemas nerviosos.

## **OBJETIVO**

Elaborar un Sistema Tutorial Computarizado para el aprendizaje de Redes Neuronales Artificiales.

# **CAPITULO 1**

---

---

## **INTELIGENCIA ARTIFICIAL Y SISTEMAS EXPERTOS**

## **1. INTELIGENCIA ARTIFICIAL Y SISTEMAS EXPERTOS**

### **1.1 BOSQUEJO HISTORICO**

El origen de la Inteligencia Artificial (IA, nombre debido a John McCarthy) se remota a la conferencia sobre teoría que tuvo lugar en 1956 en el Darmouth College. En ella aparecieron sistemas con capacidad para desarrollar juegos (juegos de damas de Samuels) y demostrar teoremas (The logic theorist de Newell and Simon) creándose el nombre de Inteligencia Artificial y el concepto de Sistemas Inteligentes.

En el año de 1981 en la Conferencia Mundial de Vancouver sobre Inteligencia Artificial, se celebró un homenaje a los pioneros de aquella conferencia de Darmouth con motivo de que la Inteligencia Artificial había cumplido 25 años tanto de software como de hardware.

El desarrollo de conocimientos que constituyen hoy la Inteligencia Artificial se puede describir en tres etapas.

**Etapla inicial:** (1956-1970) Que comprende la creación de técnicas básicas para representar el conocimiento inteligente tanto a nivel de métodos como de lenguajes.

**Etapla de Prototipos:** (1970-1981) En la que desarrollan proyectos más complejos pero todavía limitados a centros de investigación.

**Etapla de Difusión Industrial:** (1981-1994) Se desarrollan proyectos comerciales por parte de empresas privadas.

### **1.2 ¿QUE ES LA INTELIGENCIA ARTIFICIAL?**

El concepto de Inteligencia Artificial (IA) se utiliza para describir un campo de investigación sobre la estructura y las funciones del comportamiento inteligente, cuyo objetivo es el de resolver problemas a partir de sistemas que actúen en forma similar a los organismos que poseen esta característica.

La base teórica de la IA fue tomando forma desde principios de los años cuarentas, sin embargo, no fue sino hasta la siguiente década cuando los equipos de cómputo existentes tuvieron el nivel adecuado para poder programar los procesos involucrados en sus investigaciones.

Los programas de computación que interesan a la IA son primeramente procesos simbólicos que implican complejidad, incertidumbre y ambigüedad. Son procesos que requieren búsqueda, pues no admiten soluciones algorítmicas. Por tanto, la IA trata de resolver problemas y tomar decisiones similares a las que los seres humanos afrontan continuamente en su relación con el mundo. Para esto, se propone una definición que no esta en absoluto aceptada universalmente.

**Inteligencia Artificial:** "Es el estudio de cómo lograr que las computadoras realicen cosas que, por el momento, las personas hacen mejor".

Esta definición es, un tanto efímera debido a que hace referencia al estado actual de la ciencia de las computadoras. Marvin Minsky propuso otras alternativas y creó la IA como un estudio centrado en computadoras y algoritmos para generar programas capaces de dotar a una máquina con la capacidad de hacer tareas inteligentes como las requeridas para jugar ajedrez, comprender lenguaje natural y ver lo que la rodea.

Tiene que ver con el estudio del comportamiento inteligente. Su meta final es el desarrollo de una teoría que explique el comportamiento en seres naturalmente inteligentes, y que guíe la creación de entidades artificiales, capaces de mostrar comportamiento inteligente. Es una ciencia y una ingeniería.

### **1.3 CAMPO DE ESTUDIO**

Así como es verdad que uno de los fines que persigue la IA es imitar algunas de las funciones cerebrales, también es cierto que muchas de esas funciones están muy lejos de ser imitadas por las computadoras. Por el contrario, uno de los objetivos inmediatos de la Inteligencia Artificial es dotar a los sistemas inteligentes de los sentidos para resolver rápidamente una cuestión, sin necesidad de emprender una ardua búsqueda entre el sin fin de posibilidades, guiados por el sentido común.

De hecho, ya se han obtenido resultados parciales y se han desarrollado máquinas que juegan ajedrez cuyos programas han sido sumamente exitosos, al grado que son capaces, con gran destreza, de evaluar las condiciones del juego en el tablero, analizar las posibilidades y valorar las estrategias. Sin duda el ajedrez exige una sobrada capacidad intelectual, dada la impresionante arborización de jugadas posibles que se presentan en un duelo de este tipo; pero las computadoras logran procesar solo la información relevante y discriminar muchas opciones que no tendrían éxito. Sin embargo, las condiciones actuales imponen un mayor uso de recurso para enseñar a las máquinas a seleccionar y tomar decisiones.

Para avanzar en la IA, simulando la inteligencia humana, será necesario además construir en las máquinas esquemas de todas las cosas, con el requisito de que sean genéricas y flexibles, como es en nosotros. Todo lo anterior sugiere la relación simbólica que se ha establecido entre las neurociencias y la IA, en la que ambas extraen conocimientos que en ocasiones resultan insospechados.

La IA ha conformado un extenso campo de investigación pura y aplicada, dentro del que se cuentan las siguientes áreas (Lara y Gelman, 1989):

- *Solución heurística de problemas.*
- *Representación del conocimiento.*
- *Sistemas Expertos.*
- *Redes Neuronales.*
- *Recuperación inteligente de la información.*
- *Demostración automática de conjeturas.*
- *Percepción y reconocimiento de formas.*
- *Aprendizaje.*
- *Procesamiento de lenguaje natural y traducción automática.*
- *Robótica.*
- *Programación automática.*

Todavía quedan problemas por resolver para lograr que las computadoras adquieran algunos de los principios que las convertirán de verdad en entes inteligentes. Uno de ellos es el aprendizaje. En nosotros se lleva a cabo toda una serie de cambios moleculares y estructurales que facilitan la transmisión de la información en el cerebro y permiten adquirir conocimientos nuevos, fenómeno que será muy difícil de inducir en las máquinas. No obstante, el ingenio del ser humano puede realizar otras máquinas como esa misma, crear inteligencia en las computadoras.

## **1.4 LENGUAJES Y HERRAMIENTAS DE LA INTELIGENCIA ARTIFICIAL**

La informática ha desarrollado lenguajes de programación específicos de alto nivel para los diferentes campos de aplicación. Lo mismo ha ocurrido con la Inteligencia Artificial. LISP, PROLOG, Logic Tree, ExSys Professional y Level5 Object son algunos ejemplos de los lenguajes de programación y de herramientas de la IA. Por ejemplo, con LISP se han elaborado software para expresar conocimientos, formulando Sistemas Expertos y desarrollando ayudas básicas de programación.

El hecho de utilizar a las computadoras para afrontar problemas cuya solución implica un cierto grado de comportamiento inteligente, constituye un nuevo paradigma en el desarrollo de las ciencias computacionales, ya que rompe con la idea tradicional de que los equipos de cómputo electrónico son útiles al proceso de toma de decisiones únicamente en su carácter de procesadores numéricos capaces de manipular con rapidez y efectividad grandes cantidades de información. En contraposición, la I.A pretende hacer funcionar eficientemente a la computadora en campos de trabajo donde por el estado tecnológico actual es superada por las capacidades naturales del ser humano (Charniak y McDermott, 1985).

## 1.5 SISTEMAS EXPERTOS.

Los Sistemas Expertos (SE) son resultado de pretender simular o reproducir el comportamiento de un solucionador de problemas inteligentes en un programa de computadora. Los primeros intentos se dirigieron al desarrollo de solucionadores de problemas generales, más tarde se determinó que estos programas serían débiles, a menos que el conocimiento específico acerca del problema a solucionar se sumara a la guía para llegar a la solución.

La investigación en IA se ha desplazado hacia el saber específico en las últimas décadas. El proyecto que inició este desplazamiento fue DENDRAL (Hayes-Roth 1983), un SE que permitió deducir estructuras químicas a partir de los datos disponibles de los fisicoquímicos. Este proyecto que inició en 1965; demostró que para que funcionara de un modo práctico, sencillo y especializado, el Sistema debería tener conocimientos considerables de fisicoquímica, es decir poseer grandes cantidades de conocimiento específico.

Los Sistemas Expertos (SE), a diferencia de los sistemas convencionales, pueden manejar los dos tipos de conocimientos (Pagnoni 1985):

a) **El formal:** Este tipo de conocimiento se puede acceder con relativa facilidad, ya que está integrado por definiciones, hechos, teoría de literatura, etc.

b) **El informal:** También conocido como heurístico, es aquel conocimiento que abarca "reglas de dedo" desarrolladas por expertos a lo largo de su vida profesional para enfrentar con efectividad problemas complejos, vagamente definidos, y situaciones de incertidumbre que no se ajustan a un método riguroso. La extracción, articulación y computarización de éste conocimiento, es la tarea fundamental al construir un SE. Para hacer posible esta tarea los SE se basan principalmente en tres formas de representar el conocimiento que son las siguientes:

**1.- Reglas de Producción.** También conocidas como juicios hipotéticos, son una estructura lógica formada por un conjunto de condiciones que deben ser cumplidas y una parte concluyente formada por una proposición que se verifica al analizar las condiciones.

**2.- Conceptos Estructurados.** Se pueden definir como un bloque de conocimientos que contiene la información relevante respecto de un objeto, representada en un formato preciso. Esta información se puede clasificar como: *El nombre del objeto, los géneros próximos del objeto en una estructura lógica definida, las cualidades o atributo del objeto, y las condiciones para acceder la información del objeto y/o modificarlas.*

**3.- Redes Semánticas.** En ésta forma de representación del conocimiento de los conceptos se representan como nodos de una red y las relaciones que se establecen entre los conceptos como los arcos de la red. Puede haber un gran número de tipos de relaciones entre los conceptos: *género, pertenencia, propiedad, etc.*, hay una gran variedad de arcos relacionales.

#### **1.5.1 DEFINICIONES DEL SISTEMA EXPERTO.**

Los Sistemas Expertos son programas de computadora en los que, a diferencia de los programas tradicionales, no se ha vertido una solución dada a un problema, sino el conjunto de conocimientos y reglas de operación de un experto humano, en torno a un problema específico, que le permiten al programa, a semejanza del experto humano, buscar la mejor solución, entre un gran número de posibilidades, atendiendo a la naturaleza del problema y a la situación contextual en el que éste se da.

- (Parsaye y Chignell 1988) "Un S.E es un programa de computadora que haciendo uso del conocimiento y la inferencia realiza una tarea difícil de algún campo en particular, que generalmente sólo puede llevar a cabo en un experto humano, ya que la solución implica su pericia".

- (Lara 1992) "Los S.E son programas de computadora en los que, a diferencia de los programas tradicionales, no se ha vertido una solución dada a un problema, sino el conjunto de conocimientos y reglas de operación de un experto humano, en torno a un problema específico, que le permiten al programa, a semejanza del experto humano, buscar la mejor solución, entre un gran número de posibilidades, atendiendo a la naturaleza del problema y a la situación contextual en el que éste se da".

- (Pinson 1981) "Se puede definir un S.E en forma funcional como un sistema que permite la resolución de problemas en un dominio específico (medicina, geología, química, etc.), utilizando una base de conocimientos adquirida de los expertos del dominio y un mecanismo de razonamiento característico de los expertos".

- (Cuenca 1986) "Un S.E es un sistema informático que incorpora, en forma operativa, el conocimiento de una persona experimentada, de forma que es capaz de responder como esta persona, como de explicar y justificar sus respuestas".
- (Marik 1987) "Los S.E son programas sofisticados de computación que manipulan conocimientos de expertos para resolver eficiente y efectivamente problemas de una área específica".

Después de haber leído las definiciones anteriores se puede definir a grandes rasgos:

**"Un SE es como un programa de computadora que utiliza métodos artificiales de raciocinio y un conjunto de conocimientos sobre un área específica para resolver problemas poco estructurables, emulando los procedimientos que seguiría un experto humano en dicha área".**

Los Sistemas Basados en el Conocimiento (SBC) usan representación explícita del conocimiento separando la base de conocimiento del mecanismo de inferencia. Los SE pertenecen a los SBC, con la capacidad de explicar y justificar su comportamiento. Es condición crucial para un SE tener una interfaz explicativa, ya que ésta sirve para que el usuario dé validez al razonamiento del sistema en una consulta. El sistema debe poder contestar (por lo menos) por qué hace ciertas preguntas y cómo llegó a algunas conclusiones (Kim y Connor 1988).

En un comienzo los S.E se programaban con lenguajes convencionales. En la actualidad, se han desarrollado los lenguajes llamados Shells que son esqueletos de SE listos para ser llenados de conocimiento. La ventaja (y la desventaja) de estos lenguajes es que fueron desarrollados para determinado tipo de conocimiento especializado.

### **1.5.2 VENTAJAS DE LOS SISTEMAS EXPERTOS**

Los Sistemas Expertos son sistemas auxiliares que pueden ofrecer ayuda efectiva en los puestos de trabajo existentes. Los trabajos rutinarios se realizan con menos fallas, las grandes masas de datos se analizan más rápidamente y las conclusiones complejas se captan con una mayor claridad. Los Sistemas Expertos no son una amenaza, sólo al parecer técnicamente refinados y, por lo tanto, casi inteligentes. Las cintas de transporte han desplazado a los portadores de cargas, las barreras automáticas a los vigilantes de paso de nivel, la trilladora a muchos labriegos. Por otro lado, no se puede dar la sustitución de profesionales que requieren intuición o tacto por máquinas, tal es el caso de vendedores, profesores o bien gente de alta calificación como científicos, médicos, ingenieros, pues esto resulta inimaginable. Esto no impide que tales personas prescindan de los SE, pues pueden verse apoyadas por unos buenos sistemas auxiliares - y este es precisamente el objetivo de los Sistemas Expertos.

**Consecuencia:** Cuanta más responsabilidad, competencia e intuición requiera un puesto de trabajo, tanto más complejos deben ser los medios auxiliares para apoyar la labor. Parece entonces, que los Sistemas Expertos tienen un futuro técnicamente prometedor. De aquí la importancia de mencionar las ventajas del uso de Sistemas Expertos:

- a) **Autonomía.** Una vez que el Sistema Experto ha sido diseñado y completado, se hace autónomo, es decir, independiente de la presencia física del especialista;
- b) **Reproducibilidad.** El Sistema Experto en sí y el conocimiento que abarca son reproducibles a voluntad de su autor. Esto implica que, en caso de ser necesario, el producto puede reproducirse para dar servicio a miles de usuarios. Por otro lado, en contraposición con la lentitud del proceso de formación de mentores especialistas, la reproducción de un Sistema Experto es del orden de unos minutos solamente;
- c) **Bajo costo de adquisición y operación.** En tanto que disponer permanentemente de un experto supone altos costos, un Sistema Experto requiere únicamente pagar por su diseño y construcción. Este costo después se distribuye entre todos los usuarios del sistema, por lo que el costo por copia es muy reducido;
- d) **Facilidad de distribución.** Por naturaleza del sistema, pueden cubrirse necesidades en localizaciones geográficas muy dispersas y bajo condiciones de trabajo difíciles;
- e) **Mínimos requerimientos de hardware.** Los requerimientos de hardware se limitan a una computadora PC compatible con memoria RAM de 640k, que es el tipo de más amplio uso en las empresas;
- f) **Flexibilidad para modificaciones y expansión.** El Sistema Experto puede ser actualizado en el contenido de información con fines de ampliación o profundización con sólo acceder a sus archivos de texto, ya que éstos no forman parte integrante de las bases de conocimiento.

### 1.5.3 LOS SISTEMAS EXPERTOS Y SU ENTORNO.

En combinación con los Sistemas Expertos se aplica hoy, de forma intensiva, el desarrollo más reciente de las técnicas informáticas. Los ordenadores suelen ser puestos de trabajo con una gran capacidad de cálculo, pantallas de alta resolución y grandes posibilidades de interconexión. Se exigen lenguajes de programación adecuados, como el LISP o el PROLOG, buenos sistemas de "tratamiento de ventanas" y sistemas operativos potentes como el UNIX.

Estos requisitos técnicos se dan por regla general, en las llamadas *Work-stations*. A principios de los 90 también PCs de gama alta cumplieron estas condiciones.

El mayor capital de un departamento de desarrollo de Sistemas Expertos está en los *Shells* y en las herramientas, con las que se crean estos Sistemas. Un shell es la parte que queda cuando se le extrae el conocimiento almacenado en la base de conocimientos - shell significa concha. De esta forma es posible utilizar un shell con varias bases de conocimientos. La esperanza de poder generalizar los Shells es, desgraciadamente, utópica. Los problemas que pueden solucionarse con Sistemas Expertos, son demasiado distintos entre sí. Por ello hace falta realizar adaptaciones específicas para cada proyecto. No es el Shell el que marca la forma de procesar el conocimiento basándose en sus posibilidades técnicas, sino el experto, que decide cómo y en qué secuencia se procesa el conocimiento, pues el Sistema Experto tiene que adaptarse al comportamiento del experto y no a la inversa. El experto ha de poder reconocer su forma habitual de trabajo, a menudo resulta ser también el usuario del Sistema Experto. El experto no es sustituido, sino sólo apoyado en parte de su trabajo diario por la experiencia acumulada en la base de conocimientos.

Es evidente que deben ser especialistas los que se encarguen del desarrollo de sistemas expertos (Fig. 1.1). En el espacio lingüístico anglosajón se ha acuñado el concepto *Knowledge Engineer*, que en castellano significa : *Ingeniero del Conocimiento*. El ingeniero del conocimiento es el mediador entre el experto, que pone a disposición sus conocimientos, y el ingeniero de software, que procede finalmente a la implementación. El ingeniero del conocimiento analiza la viabilidad y rentabilidad de los proyectos.

Es el especialista que extrae el conocimiento mediante preguntas, lo estructura y lo convierte en formas de representación del conocimiento interpretables por un ordenador, elige los medios auxiliares necesarios para su actividad, sean métodos, hardware, lenguajes o Shells. También es el responsable de la correcta utilización del Sistema Experto y de la primera formación del usuario final.



Fig 1.1 PARTICIPANTES EN EL DESARROLLO DE UN SISTEMA EXPERTO.

El ingeniero del conocimiento es, ante todo, informático, y se le exige que tenga intuición y tacto; pues no es inusual que los expertos trabajen de forma intuitiva y no sepan volcar de forma inmediata su conocimiento. El ingeniero del conocimiento sabe que las reglas que reflejan la experiencia de los expertos se pueden modificar y refinar varias veces antes de que sean aceptadas como correctas.

Los Sistemas Expertos no han de ser vistos como sistemas aislados. En una fase posterior de aplicación se integrarán probablemente en un entorno de procesamiento de datos ya existente, y se utilizarán los datos o resultados de otros programas. A veces, un Sistema Experto cubre sólo una parte pequeña aunque decisiva, de una aplicación "clásica" mayor, por lo tanto es importante que ambos sistemas puedan interactuar entre ellos.

Esta capacidad de integración implica una planificación precisa de interfaces. Sin embargo, puede ocurrir que, para ahorrar costos, esta consideración se deja fuera de la planificación sin tener en cuenta que una reimplementación posterior resulta siempre considerablemente más cara.

No cabe duda, de que con la tecnología de los Sistemas Expertos se pueden ampliar las posibilidades de la tecnología convencional de proceso de datos, traspasando algunas de las barreras hasta ahora muy complejas.

#### 1.5.4 TIPOS DE SISTEMAS EXPERTOS

Las principales aplicaciones ó tipos de SE dentro de la Ingeniería se pueden dividir de acuerdo a la tarea a desarrollar en: (Hayes' Roth 1983).

**1.- Interpretación:** Inferen descripciones y situaciones a través de sensores de datos. Esta categoría incluye entendimiento de lenguaje, análisis de imágenes, interpretación de señales o dilucidar estructuras químicas y muchas clases de análisis inteligente. Un sistema de interpretación explica los datos observados asignándoles un significado simbólico para describir la situación o el estado actual del sistema.

**2.- Predicción:** Inferen las consecuencias de situaciones dadas. Esta categoría incluye pronósticos el clima, predicciones demográficas o de tráfico, estimaciones de cosechas o pronósticos militares. Un sistema de predicción típicamente emplea un modelo dinámico paramétrico con valores de parámetros ajustados a una situación dada. Las consecuencias inferidas del modelo forman las bases para las predicciones. Si se ignora la probabilidad de las estimaciones, los sistemas de predicción pueden generar un gran número de posibles escenarios.

**3.- Diagnóstico:** Inferen un mal funcionamiento en un sistema por observación. Esta categoría incluye diagnóstico médico, electrónico, mecánico o de programación entre otros. Los sistemas de diagnóstico típicamente relacionan las irregularidades de un comportamiento con sus posibles causas, usando una de dos técnicas.

Un método esencialmente utiliza una tabla de asociación entre comportamientos y diagnósticos. El otro método combina el conocimiento del diseño del sistema con el conocimiento de efectos potenciales en el diseño, implantación o componentes para generar posibles malos funcionamientos en forma consistente con las observaciones.

**4.- Diseño:** Configuran objetos bajo restricciones. Los sistemas de diseño desarrollan configuraciones de objetos que satisfagan las restricciones de los problemas de diseño. Esos problemas incluyen diseño de circuitos impresos, diseño de construcciones e inclusive problemas de costos. Los sistemas de diseño generan descripciones y objetos con varias relaciones con otros y verifican que estas configuraciones cumplan con las restricciones establecidas.

**5.- Planeación:** Diseñan acciones. Estos sistemas se especializan en problemas de diseño concernientes a objetos que desarrollan funciones. Incluyen programación automática en robots, proyectiles, rutas, comunicaciones, experimentos y problemas de planeación militar. Los sistemas de planeación emplean modelos de comportamiento para inferir los efectos de actividades planeadas.

**6.- Monitoreo:** Comparan observaciones del comportamiento de un sistema en características que parezcan ser cruciales para que un plan sea exitoso. Estas características cruciales o puntos vulnerables corresponden a efectos potenciales en el plan. Generalmente los sistemas de monitoreo identifican estos puntos vulnerables en dos tipos. Un tipo de vulnerabilidad corresponde a condiciones asumidas cuya violación nulificaría el plan en forma racional. Otro tipo de vulnerabilidad surge cuando efectos potenciales del plan violan una restricción de planeación.

Muchos sistemas de monitoreo por computadora existen para plantas nucleares, tráfico aéreo, etc., aunque no existe en realidad un sistema experto de este tipo.

**7.- Depuración:** Prescriben remedios a malos funcionamientos. Estos sistemas utilizan las capacidades de planeación, diseño y predicción para crear especificaciones o recomendaciones para corregir un problema diagnosticado. Existen sistemas de depuración

para programación pro-computadora en la forma de bases de conocimiento inteligentes y editores de texto, pero ninguno califica como sistema experto.

**8.- Reparación:** Ejecutan un plan para administrar una solución, para algún problema diagnosticado. Estos problemas incorporan las capacidades de depuración, planeación y ejecución. Los Sistemas Expertos asistidos por computadora se utilizan en el campo de los automóviles, aviones y mantenimiento de equipos de cómputo entre otros. Los Sistemas Expertos apenas están ingresando a este campo.

**9.- Instrucción:** Diagnostican y depuran el comportamiento de estudiantes. Típicamente estos sistemas comienzan construyendo una descripción hipotética del conocimiento del estudiante que se interpreta como el comportamiento del estudiante. Posteriormente se diagnostican las debilidades en su conocimiento y se identifica un remedio apropiado. Finalmente se planea un tutorial que interactúe con el estudiante para cubrir sus necesidades de conocimiento.

**10.- Control:** Interpretan, predicen, reparan y monitorean el comportamiento de sistemas. Un Sistema Experto de control gobierna el comportamiento en conjunto de un sistema. Para hacer esto, el sistema debe interpretar en forma repetitiva la situación actual, predecir el futuro, diagnosticar las causas de los problemas, formular un plan que remedie la situación y monitorear su ejecución para asegurar el éxito de la operación en su conjunto. El tipo de sus aplicaciones incluye el control del tráfico aéreo, control de misiles e inclusive manejo de negocios entre otros.

### 1.5.5 ARQUITECTURA DE UN SISTEMA EXPERTO

A pesar de la diversidad de arquitecturas que se manejan alrededor de los sistemas expertos, es posible distinguir los elementos básicos comunes que los forman (Fig 1.2).

- **Bases de Conocimiento:** Contiene el saber específico en la disciplina, de la cual el Sistema es Experto. Consiste en un conjunto de hechos (datos) y de reglas programadas. No contiene información específica de un problema en particular. El conocimiento plasmado en la base juega un papel más importante en la calidad y habilidad del sistema.

- **Interfaz con el Usuario:** Durante el proceso de solución del problema, un experto humano requiere de comunicarse frecuentemente con el usuario, ya sea para recabar información significativa de la situación o para informarle del estado de la solución. En un SE, el bloque funcional conocido como interfaz del usuario es el encargado de proporcionar esta comunicación, mediante el uso de una serie de recursos visuales especialmente estructurados.

- **Interfaz con el Experto:** Es aquella que permite captar información del exterior (proporcionada por el experto) e introducirla en forma adecuada en la base de conocimiento.

- **Mecanismo de Razonamiento:** También llamada máquina de inferencia que con base en el conocimiento almacenado en la base de conocimientos, infiere nuevos hechos y relaciones que amplían el conocimiento y resuelven el problema planteado, si éste es soluble en este entorno.

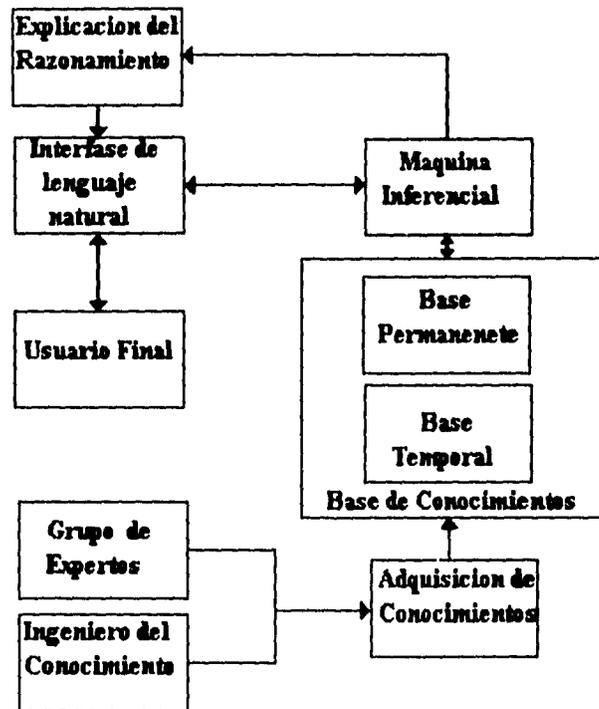


Fig. 1.2 Estructura de un sistema experto

- **Interfaz Explicativa:** Durante una consulta, no es común cuestionar al experto humano acerca de cómo obtuvo sus conclusiones o el por qué de las preguntas que realiza, ya que se tiene un alto nivel de confianza en sus procedimientos. En el caso de SE, resultaría difícil que el usuario mostrará una confianza similar frente a un programa de computadora. Por esta razón, el SE incluye una interfaz explicativa que permite interrogar al programa sobre la metodología que ha seguido para arribar a la solución y sobre la necesidad de las preguntas que realiza. Desde este punto de vista, la interfaz explicativa adquiere una enorme importancia, ya que al otorgar transparencia a la operación del SE, proporciona seguridad al usuario.

### 1.5.6 DESARROLLO DE UN SISTEMA EXPERTO

Un proceso de desarrollo abarca todos los pasos de planificación y las medidas durante la creación del software. Para este proceso se ha acuñado un término propio, el de Técnica de Procesos (TP). La técnica de procesos de software fue una reacción a la llamada crisis del software de los años sesenta. La disciplina por aquél entonces aún muy joven no parecía estar en situación de conjugar debidamente la calidad, el coste y los plazos con los deseos del cliente. Solo tras la introducción de una técnica de procesos de software con una lista de obligaciones para clientes y para fabricantes del software se pudo salir de aquel callejón sin salida.

En los primeros años de expansión de los Sistemas Expertos se solían dejar de lado los principios básicos de la técnica de procesos, ya que se trabajaba más deprisa sin dedicar mucho tiempo a las planificaciones. Con ayuda de herramientas cómodas se desarrollaban versiones de demostración en poco tiempo, para ampliarlas luego paso a paso. Tenían lugar, en aquéllos momentos, el paso a una nueva técnica de desarrollo de software. Esta fase de entrada ha sido hoy ampliamente superada pasando ahora a primer plano el punto de vista de la ingeniería de software. En la Fig.1.3 se presenta en forma gráfica el desarrollo de un SE

La técnica de procesos convencional no puede ser aplicada simplemente al desarrollo de Sistemas Expertos. Las nuevas posibilidades en software y hardware solo pueden aprovecharse al máximo con una Técnica de Procesos convenientemente modificada.

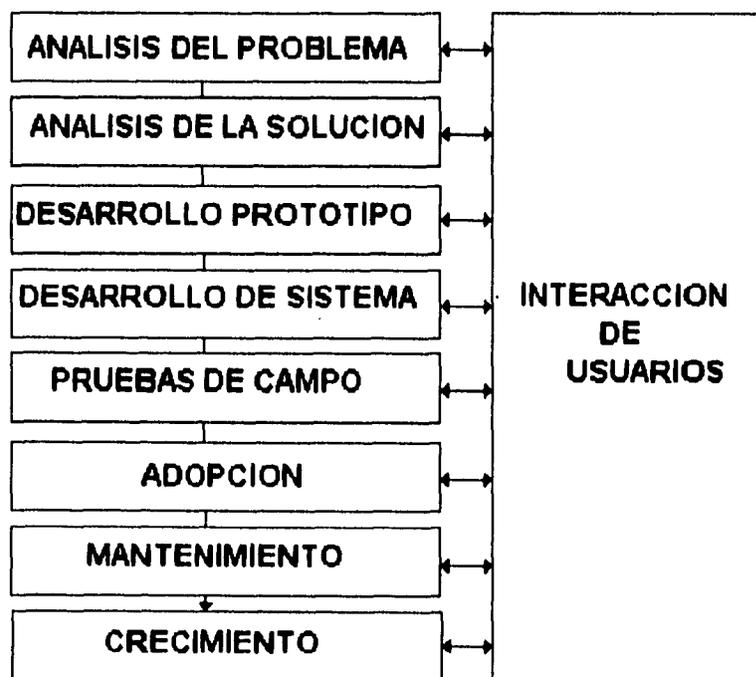


Fig.1.3 Desarrollo de un Sistema Experto

### 1.5.7 ANALISIS DEL PROBLEMA

En el primer subproceso la importancia se centra en la viabilidad y necesidad como principio de un Sistema Experto. Para ello hay que aclarar los siguientes puntos:

- Forma de trabajo del experto,
- Tipo de alcance del conocimiento utilizado para la resolución del problema,
- Entorno de aplicación,
- Usuarios
- Costes y rentabilidad

Al final de este subproceso se toma la decisión sobre la idoneidad de la técnica de Sistemas Expertos para solucionar el sistema planteado.

Algunas preguntas no podrán tener una respuesta clara en esta fase. Por ello es posible que se modifique la decisión sobre la utilización de la técnica de Sistemas Expertos a lo largo de la planificación.

### 1.5.8 EL INGENIERO DEL CONOCIMIENTO

Es la persona que trabaja con un experto humano para identificar y refinar el conocimiento que se necesita para resolver un tipo particular de problemas.

La esencia del trabajo del ingeniero del conocimiento consiste en su habilidad para obtener el conocimiento del experto humano (Fig. 1.4) durante una serie de entrevistas y sesiones de observación para estudiar conjuntamente como resuelve el sistema los casos presentados u determinar como mejorarlo (Fig. 1.5).

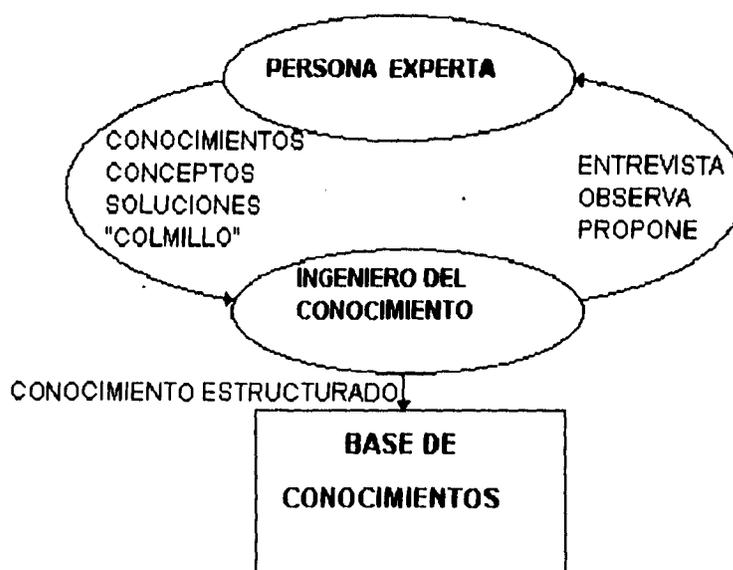
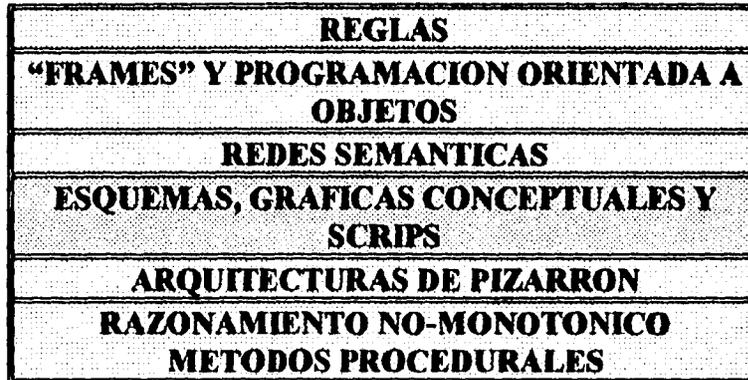
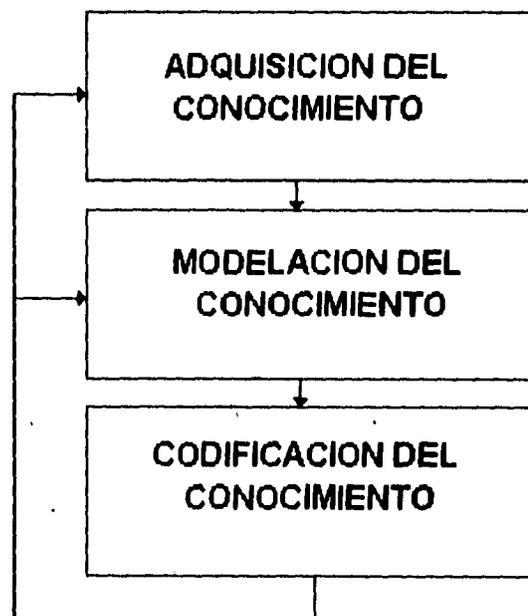


Fig 1.4 Adquisición del conocimiento



**Representación del conocimiento**



**Fig 1.5 Tareas de la Ingeniería del conocimiento**

## 1.6 MECANISMOS DE RAZONAMIENTO

Dada la importancia del mecanismo de razonamiento en un SE se hará una descripción más amplia:

El mecanismo de razonamiento, al cual se hace referencia también como "*máquina de inferencia*" o "*estructura de control*", es un conjunto de rutinas y estructuras tanto para la producción como para el control de decisiones, así como para el manejo de preguntas al usuario. Esta parte del SE no debe contener, por lo menos teóricamente, información acerca del dominio específico del problema y debe contar con cierta independencia para permitir un grado de generalidad.

A pesar de lo anterior, la base de conocimientos y el mecanismo de razonamiento están íntimamente ligados, dado que la primera presenta la organización del conocimiento en base al cual el segundo va a trabajar. Existe entonces una estrecha relación entre la organización del sistema experto y su control.

Los mecanismos de razonamiento no pueden ser completamente independientes del tipo de problemas a resolver, sin embargo, estas rutinas de razonamiento preferentemente no deben ser tan especializadas, pues no podrían entonces aplicarse a otro tipo de problemas.

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

- *Estrategias de razonamiento.*
- *Estrategias de explicación.*
- *Estrategias de interpretación.*

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

El SE para un problema debe buscar una solución. esta tarea le corresponde directamente al mecanismo de razonamiento. El acceso directo a una solución única no es posible en la mayoría de los casos. Generalmente son utilizadas técnicas de búsqueda, así como generadores de soluciones. Estos últimos permiten al sistema tener una visión general de todas las posibles soluciones al problema, permitiéndole probar cada una de ellas hasta encontrar la más apropiada.

Entre el método de búsqueda de soluciones, el más simple es el de analizar todas las alternativas una por una. A este método se le denomina de "*fuera bruta*". El problema con este método es la existencia, en algunas ocasiones, de espacios de búsqueda de gran tamaño.

### 1.6.1 ELECCION DE UNA DIRECCION DE SOLUCION.

a) **Encadenamiento hacia adelante:** Cuando se tiene un conjunto de datos o ideas básicas como punto de partida, el encadenamiento hacia adelante resulta una técnica natural para direccionar las soluciones de los problemas. Esta metodología ha sido utilizada por Sistemas Expertos en el área de análisis de datos, diseño, diagnóstico y formación de conceptos.

*Ejemplo:* Sistema AM (Lenat 1976), que descubre conceptos matemáticos en base a la tarea de formación de conceptos. El sistema usa encadenamiento hacia adelante iniciando el proceso con ideas elementales de la teoría de conjuntos. Construye un espacio de búsqueda con todas las posibles conjeturas que pueden ser generadas de esas ideas elementales. Elige las conjeturas más interesantes y prosigue en una línea de razonamiento.

b) **Encadenamiento hacia atrás:** Esta aproximación es aplicable cuando nuestro punto de partida es una meta o hipótesis. Planificación es un buen ejemplo para este tipo de aproximación, debido a que la función del planificador es construir un plan para poder alcanzar las metas deseadas. Un planificador considerará su meta sin el consumo de recursos excesivos o la violación de restricciones. Si hay una meta en conflicto, el planificador tiene que establecer prioridades.

c) **Encadenamiento hacia adelante y hacia atrás:** Cuando el espacio de búsqueda es grande, la teoría de doble búsqueda suele ser eficiente. El método consiste en tomar un estado inicial y las metas o hipótesis siguiendo un proceso de convergencia para poder igualar las soluciones en un punto intermedio. El método que se utiliza es parecido al de relajación. Esta aproximación también es muy útil cuando el espacio de búsqueda puede dividirse jerárquicamente. En tales casos, la búsqueda se combina apropiadamente en términos de "Top down" y "Bottom up". Una búsqueda de tal naturaleza se aplica en forma particular a problemas complejos, incorporando además incertidumbre.

*Ejemplo:* Sistema HERSAY II (Erman 1980). Este sistema cuya función es interpretar señales con el propósito de entender el habla, divide el problema jerárquicamente desde arriba en diferentes niveles, con oraciones en el tope y parámetros que miden señales en la parte inferior. El sistema hace un procesamiento "Top down" y "Bottom up" con una aproximación de relajación para ampliar y combinar los candidatos parciales.

**d) Manejo de eventos:** Esta aproximación para encontrar soluciones de problemas es parecida al encadenamiento hacia adelante excepto que se basa en el estado actual del problema. La técnica va tomando las etapas subsecuentes en base a nuevos datos o en respuesta a un cambio de situación. La técnica de manejo de eventos es muy apropiada para operaciones en tiempo real, tal como monitoreo o control.

**Ejemplo:** Sistema VM (Ventilator Manager) (Cherniack 1977), cuya función es interpretar en tiempo real el significado clínico de los datos que arrojan un sistema de monitoreo fisiológico de aliento.

## **1.7 SISTEMAS EXPERTOS TUTORIALES.**

También conocidos como Sistemas Tutoriales Inteligentes. Estos sistemas se definen de la forma:

**(Kearsley 1987).**

**"Son programas de computadora que utilizan técnicas de inteligencia artificial para ayudar a una persona a aprender".**

Los Sistemas Expertos Tutoriales (SET) se ubican en la intersección de las ciencias de la computación, las psicología cognitiva y la investigación educacional.

El hecho de que la investigación de SET se extienda sobre tres diferentes disciplinas, significa que hay diferencias muy grandes en terminología, metas de investigación, marcos teóricos y distinto énfasis entre investigadores, además de que esto requiere de un entendimiento mutuo de las tres disciplinas involucradas.

Por muchos años, los SET fueron confiados al laboratorio de investigación. Actualmente, con el crecimiento y la aceptación del campo de la IA, ha sido factible el introducir programas de esta disciplina en las áreas prácticas de entrenamiento y educación.

Los SET han tomado muchas formas pero esencialmente han preparado los componentes principales de un mecanismo de instrucción, de tal forma que le permite al estudiante y al sistema la flexibilidad de un ambiente de aprendizaje que se asemeja a lo que actualmente ocurre cuando el estudiante y el profesor se sientan juntos e intentan enseñar y aprender.

A pesar de la diversidad de arquitecturas que se manejan alrededor de los sistemas es posible distinguir los elementos básicos comunes que los forman:

**1.- Módulo experto:** Consiste de conocimientos sobre un dominio específico del cual el sistema intenta enseñar al estudiante. Este módulo se utiliza para generar el conocimiento instruccional, así como para evaluar el desempeño del estudiante. Los métodos de IA más representativos son las redes semánticas sistema de reglas de producción, representación procedural y la construcción de marcos o guiones.

**2.- Módulo del estudiante:** Se utiliza para estimar el estado de conocimientos del estudiante y hacer hipótesis acerca de las estrategias de razonamiento utilizadas para alcanzar el estado actual de conocimientos.

Como la mayoría de los SET representan el conocimiento del estudiante como un subconjunto de la base de conocimientos de un sistema experto, el modelo se construye comparando el desempeño del estudiante con el desempeño de un sistema experto en la misma tarea. A esta técnica se le llama el "*Modelo de Sobreposición*".

Otra técnica es la de representar las tareas erróneas o mal aprendidas. Las cuales no son subconjuntos de la base de conocimientos expertos como variantes de este conocimiento. A esta técnica se le conoce como "*Modelo de Depuración*".

El modelado del conocimiento del estudiante y el comportamiento de aprendizaje, básicamente utilizan dos rutinas:

- *Trazar entre la estructura de conocimientos aquellas áreas en las que el estudiante ha dominado o ha intentado aprender; y*
- *Aplicar patrones de reconocimiento al histórico de respuestas del estudiante para hacer inferencias acerca del entendimiento de la habilidad y el proceso de razonamiento utilizado para llegar a esa respuesta.*

Para mantener el modelo del estudiante, existen cuatro fuentes principales de información (Clancey 1979):

- i) El comportamiento del estudiante es la solución de problemas, observado por el sistema.
- ii) Cuestionamientos directos al estudiante.
- iii) Suposiciones basadas en la experiencia de aprendizaje del estudiante.
- iv) Suposiciones basadas en alguna medida de la dificultad del material presentado.

Sin embargo, la mayoría de los sistemas utilizan únicamente las dos primeras fuentes de información para mantener el modelo del estudiante.

**3. Módulo tutorial:** Es un conjunto de especificaciones acerca de cuál es el material que el sistema debe presentar y cómo y cuándo debe hacerlo.

En los SET existentes, las estrategias de instrucción básicamente se representan por dos métodos:

i) **El método Socrático**, provee a los estudiantes de cuestionamientos para guiarlos a través de un proceso de depuración de sus propias malas interpretaciones. En el proceso de depuración, se asume que el estudiante razona acerca de lo que sabe y lo que no sabe y de esta forma modifica sus interpretaciones (Carbonell 1970 y Stevens 1977).

ii) **El método de entrenamiento**, provee a los estudiantes un ambiente en el cual los compromete en actividades como juegos por computadora con el fin de aprender habilidades relacionadas o habilidades generales para la solución de problemas. La meta del programa es tener a los estudiantes entretenidos y aprender como consecuencia de la diversión (Burton 1979).

Los tres componentes no están totalmente desarrollados en cada sistema dado la complejidad y el tamaño de los SET. La mayoría de ellos se enfoca en el desarrollo de un solo componente de lo que constituye un sistema completo utilizable (Clancey 1979).

### **1.7.1 TIPOS DE SISTEMAS EXPERTOS TUTORIALES.**

#### **1.7.1.1 TUTORES CON INICIATIVA.**

Estos programas representan a los más antiguos SET, en donde, se compromete al estudiante a una conversación en dos sentidos y se intenta enseñarle vía el método Socrático con un descubrimiento guiado. Este tipo de sistemas se utiliza con mayor frecuencia en la enseñanza de habilidades conceptuales o procedurales.

Ejemplo: Sistemas SCHOLAR y SOPHIE.

#### **1.7.1.2 TUTORES DE ENTRENAMIENTO.**

Un entrenador observa el desempeño del estudiante y lo provee de consejos que le ayudan a mejorar su desempeño. Los sistemas de entrenamiento se utilizan en programas del tipo solución de problemas como son simulaciones y juegos.

Ejemplo: Sistemas WEST y WUSOR.

### **1.7.1.3 TUTORES DE DIAGNOSTICO**

Este tipo de sistemas intentan depurar el trabajo del estudiante. Los sistemas manejan un catálogo de errores que identifican los conceptos mas entendidos o faltantes que un estudiante puede tener al momento de resolver un problema. Los tutores de diagnóstico son apropiados para casi cualquier tipo de situación de resolución de problemas, aunque son sencillos de implantar para problemas con soluciones cerradas. Ejemplo: Sistemas BUGGY y PROUST.

### **1.7.1.4 EL CONCEPTO DE MICROMUNDOS**

Este tipo de sistemas involucran el desarrollo de herramientas computacionales que le permiten al estudiante explorar dominios como la geometría, la física o la música. Ejemplo: LOGO.

## **1.7.2 SISTEMAS EXPERTOS**

Este tipo de sistemas pueden ser utilizados como herramientas de trabajo y para proveer prácticas en la solución de problemas y toma de decisiones. Claramente se tiene un gran potencial de aplicaciones en el área aunque el uso de Sistemas Expertos ya sea en entrenamiento o educación no ha sido difundido.

La diferencia con los Sistemas Expertos convencionales es que los SET incluyen un módulo adicional que contiene las reglas que controlan el proceso de enseñanza.

A pesar de sus grandes diferencias, la mayoría de estos sistemas tratan con elementos y componentes similares. Por ejemplo, el elemento más importante en el desarrollo de un SET es la forma apropiada de representar el conocimiento de l que se trata. Otro elemento de gran importancia es como modelar el comportamiento actual del estudiante.

El compartir los conceptos y metodología proveen los SET un objetivo común y una fundamentación teórica. Esta situación es análoga al período de la aviación en la que los aviones de propulsión a chorro sustituyeron a los aviones de hélice. La mayoría de los conocimientos acerca de volar fueron aún relevantes pero tuvieron que ser replanteados en términos de la nueva tecnología.

### 1.7.3 DESARROLLO DE LOS SISTEMAS EXPERTOS TUTORIALES

Los SET tienen sus raíces durante los años 20's con el desarrollo de lo que se llamó "Las máquinas de enseñanza". Estas máquinas fueron un intento de construir dispositivos interactivos de enseñanza. Skinner en los años 50's en sus trabajos sobre la instrucción, aportó las bases de la metodología para la instrucción lineal programada tan difundida durante los años 50's y 60's. La instrucción programada fue el modelo para muchos esfuerzos iniciales en SET.

Uno de los esfuerzos más representativos fue desarrollado en la Universidad de Stanford por el investigador Suppes (1968), en donde se demostró que en un ciclo regular añadiendo un poco de instrucción y sesiones de práctica con una computadora, se podrían incrementar significativamente las habilidades del estudiante.

Durante los 70's se desarrollaron varios proyectos tanto en Universidades como Institutos de entrenamiento (la mayoría se iniciaron en los sesentas): En los más significativos se encontraron SOLO, en la Universidad de Pittsburgh, con el fin de enseñar el uso de computadoras como herramientas personales y los sistemas PLATO y TCCIT que se desarrollaron con la facilidad de proporcionar sistemas de instrucción de bajo costo. Estos sistemas fueron probados en Institutos y Universidades y eventualmente se convirtieron en productos comerciales. Su principal beneficio fue el permitir a un gran número de investigadores el obtener experiencia práctica, así como la apertura de una nueva era en los SET con la aparición de las microcomputadoras.

Otro desarrollo importante en los SET se dio en la estructura de los componentes de instrucción. En sus primeros años todos los componentes estaban combinados y almacenados en un mismo archivo. Esto implicaba que cualquier modificación al contenido o a las reglas de instrucción derivaba en prácticamente un sistema nuevo.

A principios de los 70's Seidel desarrolló un proyecto para la Armada Americana llamado IMPACT en el cual el contenido de la materia (textos y gráficas) y las reglas de instrucción, estaban separados en diferentes archivos de datos. Asimismo, este sistema permitía que una variedad de actividades sucedieran en forma simultánea.

El inicio de los SET lo marcó el sistema SCHOLAR desarrollado para enseñar geografía sudamericana (Carbonell 1970). La base de datos de SCHOLAR es compleja pero bien estructurada con la forma de una red de conceptos, datos y procedimientos. Los elementos de la red son unidades de información definiendo palabras y eventos en la forma de arboles multiniveles. El método de enseñanza es el diálogo Socrático. El sistema primero diagnostica los conceptos erróneos o mal interpretados y luego presenta material que fuerza al estudiante a ver sus propios errores. Los mecanismos de inferencia para contestar las preguntas de los estudiantes y evaluar sus respuestas son independientes del contenido de la red semántica y es aplicable a diferentes áreas de dominio.

Como extensión a SCHOLAR surge el sistema WHY (Stevens 1977) diseñado para enseñar la causas del fenómeno de lluvia repentina, un complejo proceso geofísico que está en función de muchos factores no relacionados WHY implanta el método tutorial Socrático Heurístico para describir las estrategias globales utilizadas por los seres humanos para guiar los diálogos.

Otro avance importante en la aplicación de técnicas de IA en el campo de los SET fue el crear nuevos ambientes educacionales a través del control total de la experiencia aprendida por el estudiante. Como ejemplo el Sistema LOGO.

Un aspecto interesante de los Sistemas Tutoriales fue el desarrollo de características auto-mejorables. Estos tipos de sistemas tenían dos grandes componentes: Un programa de enseñanza adaptativo que se expresa en reglas de producción y un componente auto-mejorable que lleva a cabo cambios experimentales en las reglas de producción del programa de enseñanza. Este trabajo fue particularmente importante por su naturaleza adaptativa. Como ejemplos de esta línea de investigación existen el tutor cuadrático auto-mejorable y el tutor de integración simbólica.

# **CAPITULO 2**

---

---

## **ASPECTOS PEDAGOGICOS**

## **2. ASPECTOS PEDAGOGICOS.**

### **2.1 INTRODUCCION.**

La educación del futuro desde la perspectiva de la "pedagogía del lenguaje total", será planteada como aquella que le brinde al individuo la oportunidad de ubicarse dentro de las condiciones reales que configuran a la sociedad y ya que los medios de comunicación forman parte muy importante en el concepto de conceptualización del mundo que se da en el individuo, la escuela del futuro deberá incorporar los medios de comunicación social como un instrumento para expresar la realidad, ofreciendo así al alumno la posibilidad de incorporarse a un mundo verdadero.

Larry Shorter nos dice que hoy en día la educación es decisión del individuo, y según sus necesidades, intereses e ideales es permanente o continua. La educación en el futuro deberá estar al alcance de todos, lo cual significa un aprendizaje activo, de investigación y exploración continua.

### **2.2 ASPECTOS PEDAGOGICOS**

El aprendizaje es un factor que influye de modo importante en todos los seres vivos. Permite al hombre adaptarse a las modificaciones del ambiente para poder sobrevivir. La lucha por la supervivencia requiere realizar muchos ajustes y adaptaciones. Desde el momento de la concepción, la experiencia empieza a dar forma al organismo en desarrollo.

El término aprendizaje tiene una gran amplitud que abarca cambios tan distintos como la respuesta de un modo diferente a una señal, la adquisición de una habilidad, la alteración de la manera de percibir una cosa, el conocimiento de un hecho y el desarrollo de una actitud frente algo. Las consecuencias últimas del aprendizaje van, inclusive, mucho más allá del individuo, puesto que las formas que presentan los idiomas, las religiones, las culturas, las leyes y las costumbres, son producto directo de modificaciones practicadas por la experiencia humana.

El aprendizaje nunca es observado directamente, es siempre una inferencia derivada de la observación de un cambio, o de una serie de cambios.

Es difícil definir el concepto de educación por que la palabra "educación" se emplea de múltiples maneras. En su aplicación mas común equivale a asistir a la escuela y trae a la mente toda gama de actividades que se llevan a cabo en los jardines de la infancia, escuelas, colegios, institutos y universidades. En este sentido su significado es vago, puesto que designa cualquiera de las diversas clases de aprendizaje.

En este contexto se podría decir que "la educación también implica actitudes, creencias que se aprenden a través de la participación en la vida social. Así pues, educación designa el proceso general por el cual aceptamos los valores de nuestra sociedad, y por esta razón podemos decir que la educación es un proceso que dura toda la vida.

Una tercera concepción de la educación se refiere a la percatación y curiosidad intelectual por todo lo que ocurre en la tierra y la consiguiente búsqueda por satisfacer la curiosidad.

### 2.3 ENSEÑANZA TRADICIONAL

Aunque difieren mucho en sus puntos filosóficos, Platón y Aristóteles plantearon las bases fundamentales en la que se sustentan la perspectiva tradicional de la educación.

Ambos afirman la prioridad del intelecto y establecen la noción de educación como trascendencia lograda por una élite intelectual, donde las masas se mantienen en un estado físico y material suficientemente cómodo, pero de vasallaje mental.

Aunque sobre este aspecto han habido cambios substanciales y se han integrado otros elementos, la educación tradicional se basa en la filosofía de Platón y Aristóteles en el aspecto que se subraya la autoridad del maestro y se considera que su función consiste en impartir a sus alumnos el conjunto requerido de la asignatura.

"La tarea del maestro es la base y condición del éxito de la educación; a él le corresponde organizar el conocimiento, aislar y elaborar la materia que ha de ser aprendida, en una palabra, trazar el camino y llevar por él a sus alumnos".

"La escuela queda separada de la vida exterior y lo que ocurre en ella se ve primordialmente como una preparación para el futuro, más que como un enriquecimiento del presente". "La escuela se construye en un mundo aparte, al margen de la vida diaria, en un reintentado reservado y preservado del mundo exterior".

La escuela tradicional concede importancia a los conocimientos y a la cultura en general y sostiene que " la mejor forma de preparar al individuo para la vida, es formar su inteligencia, su capacidad de resolver problemas, sus posibilidades de atención y de esfuerzo.

Los conocimientos son valorados por su utilidad para ayudar al individuo en el progreso de su personalidad: edificando sólidos conocimientos se favorece el desarrollo global del individuo". De esta forma, la escuela tradicional significa "*por encima de todo método y orden*".

Algunas observaciones hechas en diferentes sistemas educativos permiten detectar que los educandos, casi siempre, permanecen sentados física y mentalmente escuchando, acatando ordenes y, por supuesto, aprendiendo a adivinar las respuestas correctas a formulaciones sesgadas. En otras palabras, están aprendiendo a manipular, a percibir que resulta mejor la aceptación pasiva que la crítica, o que el aprendizaje memorístico es la mejor forma para triunfar.

### 2.3.1 EL PENSAMIENTO ESPONJA

Si la educación se concibe como la absorción de datos, escuchando pasivamente en una aula en la que el profesor habla *ex-catedra* en todo el momento, como un dueño único y absoluto de la verdad, los estudiantes no tienen nada que hacer. O más bien, si tienen algo que hacer: acumular memorísticamente hechos, cifras, nombres, citas célebres. Todo ello sin relación alguna entre sí.

El mejor estudiante, será aquel que sea capaz de repetir exactamente cuanto el profesor ha dicho durante horas y horas de transmisión verbalista: pues, además se cree que a mayor número de palabras escuchadas y memorizadas, será mayor el índice de aprendizaje.

Los resultados previsibles en este aprendizaje pueden ser desastrosos. Por una parte, los educandos lograrán ser muy buenos burócratas o muy buenos consumistas, obedientes lo mismo a la voz demagógica de las autoridades que a los reclamos publicitarios de los mensajes radiofónicos, etc.

Por otra parte, se aprende que nada importa lo que se diga, siempre y cuando esto agrade al oído de los interlocutores del sistema familiar, escolar o laboral. Este tipo de mensajes implícitos en el ambiente es el que más repercusiones tiene sobre las personas.

En efecto, cuando durante treinta años se repiten los mismos programas y, clase tras clase, se dictan los mismos apuntes es obvio que los estudiantes se estimularán y motivarán para mantener alerta su pensamiento crítico y su espíritu de investigación. Más aún, al ir tomando notas, en el estudiante puede nacer la vocación de "maestro" y pensar: "Estos serán los apuntes que dictaré a mis alumnos. Están perfectos".

### 2.3.2 AMBIENTE EDUCATIVO

Es necesario crear un ambiente educativo: laboral, familiar, escolar, en el cual, quienes lo forman:

a) *Confíen* profundamente de que no serán reprimidos, sino alentados para poner en juego su propia capacidad de resolver problemas y de realizar críticas desde su propia perspectiva científica e ideológica. Esta confianza mutua no implica que siempre se acierte en la solución. También se aprende por ensayo y error. Es más educativo crear un sistema en el que se respire confianza, no sólo por principios éticos o pedagógicos sino por la necesidad de responder a las demandas de una sociedad con agudas tensiones y un cambio acelerado.

b) *Disfruten* a todo lo largo del proceso de investigación, pues la alegría también influye en una mayor capacidad de aprendizaje.

La perspectiva de ser responsables del proceso de investigación deberá ser alentada no dando respuestas a todas las preguntas que se formulen a quien supuestamente sabe más, aunque el educador tenga sinceros deseos de transformar a sus discípulos.

c) *Distingan* claramente entre lo importante, lo esencial y lo secundario, lo accidental. En otras palabras, hay que aprender a *mirar* ( en profundidad) no simplemente a ver (superficialmente), pues de otra manera no es posible *capturar* la diferencia entre lo importante y lo trivial.

d) *Sean flexibles* tanto en lo que respecta a sus propias "verdades científicas" como en lo que respecta al valor de sus respuestas. Aunque confiados en su propio criterio, investigan y emiten opiniones fundamentadas una vez que tuvieron acceso a toda la información, directa o indirecta, disponible.

En otras palabras, comprenden que las verdades científicas son modelos provisionales para interpretar la realidad y que, por lo tanto, las respuestas son relativas, pues dependen de diversos enfoques teóricos, así como de diferentes sistemas metodológicos y perspectivas.

Al no cambiar actitudes en este sentido se corre el riesgo de ser manipuladores, al servicio de ideologías dominantes. De aquí que resulte más difícil establecer diálogo entre profesor y alumno. En consecuencia para formar la conciencia crítica resulta necesario cambiar los ambientes educativos diversos en los que hay que *aprender a ver* de un modo u otro.

Obviamente, no podrán introducirse innovaciones relevantes si no se modifican las actitudes de los "educadores", ya que sus actitudes, creencias y prejuicios determinan el clima de estos ambientes de aprendizaje.

### **2.3.3 DESVENTAJAS DE ENSEÑANZA TRADICIONAL.**

Es más educativo considerar a los estudiantes como objetos que como personas. Desde el punto de vista de *Skinneriano*, los individuos únicamente son máquinas, manipuladas mediante reforzamientos positivos o negativos.

Además, no conviene involucrarse con los estudiantes ni establecer con ellos relaciones profundas de amistad, porque ¿cómo un amigo puede ser verdugo, juez y examinador?. En consecuencia habrá que mantenerlos a distancia y conservar relaciones de "respeto".

Los estudiantes-objeto, aprenderán a considerarse como objetos y no sujetos de su proceso educativo, el cual, esencialmente, es un proceso de auto-desarrollo. Además si la propia institución educativa no se encuentran ni solidaridad ni amistad ¿cómo podríamos pensar que asimilen estos valores?. Este ambiente, en consecuencia, puede ser educativo o des-educativo.

## **2.4 EDUCACION E INFORMATICA**

La solución que se propone a este problema es la planeación educativa basada en la educación continua como principio básico, cuyo propósito sea el pleno desarrollo personal, y su resultado sería una sociedad individualmente más creativa y más satisfactoria.

Los planteamientos teóricos parecen interesantes, el problema fundamental es cómo ponerlos en práctica y además crear la infraestructura, y generar los recursos para llevar a cabo la tarea de poner la educación al alcance de todos. Algunas de las respuestas se presentan a continuación:

- 1.- Crear programas de extensión Universitaria que establezcan fuentes alternas de educación, con base a las necesidades sentidas y manifiestas.
- 2.- Integrar grupos de expertos en áreas de educación, pedagogía y telecomunicaciones, para utilizar la infraestructura existente, así como nuevos desarrollos para ofrecer posibilidades reales de educación continua, educación a distancia, etc;
- 3.- Evaluar la estructura de la curricula actual de las Instituciones de educación superior y las necesidades presentes y futuras de la sociedad en su conjunto. Referente a la educación superior, desarrollarla en la modalidad de continua, con una tecnología adecuada, para ofrecer alternativas reales.

La educación en cómputo e informática es uno de los grandes retos que enfrenta la sociedad de hoy, el lograr la formación e información en cómputo e informática para toda la comunidad es una vía indispensable para entender a la sociedad actual, sus estructuras, su interés, y todas las relaciones que se dan en un ámbito cada vez más tecnificado, y que incorporan innovaciones tecnológicas a un ritmo vertiginoso.

La educación actual se plantea consciente e inconscientemente la preparación del hombre para una sociedad futura, una sociedad en la que la electrónica, la computación y la informática juegan un papel relevante.

El cómputo y la informática han sido señalados como un conjunto de conocimientos para incorporarse en el proceso educativo; sin embargo, muchas de las experiencias han tenido pobres resultados, principalmente debido a que se han hecho como una introducción de modernidad mediante la incorporación de equipos y programas de cómputo, pero con escasa o nula vinculación con las necesidades reales en su conjunto y mucho menos con la práctica profesional, lo cual resulta como un parche de modernidad con pobres resultados en el proceso educativo.

#### 2.4.1 APRENDIZAJE POR COMPUTADORA

Las computadoras están apareciendo rápidamente en las escuelas de todo el mundo. En todos los países el mayor problema es el entrenamiento de los maestros para el uso de estas. Debido al rápido incremento en el número de computadoras en las escuelas, la pregunta es: ¿Qué sucede con las computadoras cuando llegan a las escuelas? Se tienen dos problemas:

- 1) La preparación de los maestros para usar el hardware.
- 2) Proveer adecuados materiales para la enseñanza con la computadora.

Casi la totalidad de los maestros simplemente *no tienen acceso a ningún programa apropiado* al empleo de las computadoras en la educación, ya que no tienen tradición en esta área.

La solución al problema del entrenamiento de los maestros es desarrollar un buen material para el aprendizaje basado en el uso de computadoras *sobre la computadora*, solo para maestros. Los maestros aprenderían *en la computadora*, no respecto a la computadora.

La instrucción por computadora como un término general "*Instrucción*", como se aplica entre otras áreas, es ser capaz de leer y escribir de modo competente, ser capaz de realizar la aritmética esencial y de entender a la ciencia en un sentido general.

Estamos interesados en los maestros como maestros auxiliando a los estudiantes en el proceso de aprendizaje. Por lo que podemos enfocarnos al uso de la computadora en sus propias clases.

#### **2.4.2 LA COMPUTADORA COMO VEHICULO PARA EL APRENDIZAJE**

La clave está en que la computadora por si misma proporciona una solución posible y factible. La computadora es un excelente instrumento para el aprendizaje y puede usarse para auxiliar tanto de los maestros como de los estudiantes. Programa de cómputo interactivos aunados al acceso de solución de problemas por computadora, pueden proporcionar una gama de posibilidades no alcanzables de ninguna otra forma. Cada persona puede ser tratada como un individuo en particular, con todas las características que le distinguen de cualquier otra persona que se considere. Cada persona puede avanzar a su propio ritmo. Además, la experiencia del aprendizaje puede ser activa, con el educando como participante en lugar de como simple espectador.

En casi todas las Universidades la mayor parte del tiempo de los estudiantes se dedica a la lectura de libros y tal vez las conferencias estén en segundo lugar. La computadora sólo cuenta con una pequeña fracción del tiempo de aprendizaje del estudiante.

Las computadoras llegan a las universidades por su utilidad como herramientas de investigación. En todos los campos universitarios los investigadores son los principales usuarios. El uso administrativo ha crecido bastante rápido. Y mientras el uso educativo a crecido, mucho de él es extremadamente simplista; además, su crecimiento no ha coincidido con otros usos de la computadora.

A pesar de que los catedráticos y administradores universitarios rara vez lo aceptan, la gran mayoría de los cursos de todas las universidades se basan casi totalmente en un pequeño conjunto de libros de texto.

### 2.4.3 ¿PARA QUE SOFTWARE EDUCATIVO?

Aunque este tipo de software puede tener algún valor didáctico, su objetivo principal no es auxiliar el proceso enseñanza-aprendizaje, en los procesos administrativos y de investigación de otra índole.

El software calificado como educativo es aquel que tiene como objetivo intervenir en el proceso de enseñanza-aprendizaje, ya sea en la forma tradicional, apoyando la transmisión o ejercitación de un conocimiento particular de un tema y materia determinada, o bien de una forma más innovadora, tratando de provocar en el alumno un interés genuino por la investigación. En este último caso el software plantea problemáticas y no contenidos particulares, que el alumno habrá de manipular a través de variables, advirtiendo asimismo, las transformaciones surgidas de su interacción con el software durante el proceso simulado.

Este software proporciona al alumno una guía para la comprensión de un concepto, a partir de ejemplos particulares y tangibles que le permitan interaccionar y observar los efectos de esa interacción en diferentes niveles de representación del fenómeno.

Como instrumento de medición, el software y la computadora presenta un fuerte apoyo a las ciencias experimentales, ya que con éstos se pueden desarrollar experimentos sobre algunos conceptos que especialmente en los niveles educativos básicos son transmitidos como dogma.

Ante estas aseveraciones se pueden plantear algunas respuestas al problema planteado.

Es evidente que no podemos pensar que lógicamente el software educativo sustituye al maestro. Tampoco podemos pensar que representa una solución a todos los problemas de la enseñanza de las ciencias, especialmente de las ciencias básicas, ni mucho menos creer que constituye una forma rápida de aprender un tema en particular, sin el menor esfuerzo.

En cambio, sí podemos considerar el software educativo como un poderoso instrumento de apoyo, dinamización y revisión de nuestros actuales conceptos de aprendizaje, y de las formas actuales de transmisión y generación de conocimiento.

# **CAPITULO 3**

---

---

## **REDES NEURONALES ARTIFICIALES**

### 3. REDES NEURONALES ARTIFICIALES

#### 3.1 REDES NEURONALES ARTIFICIALES ELEMENTALES.

##### 3.1.1 BOSQUEJO HISTORICO

En la década de los 40's y principios de los 50's, varios investigadores entre los que destacan McCulloch y Pitts Neumann (1956) y Culbertson (1956) elaboraron modelos matemáticos de neuronas y Redes Neuronales. En la década de los 50's, varios investigadores, entre ellos Farley y Clark (1954) y Rochester, Holland, Haibt y Duda (1956, combinaron los resultados obtenidos por los matemáticos, biólogos y los psicólogos y desarrollaron modelos de simulación en computadora de neuronas y Redes Neuronales, dando lugar a la forma actualmente más generalizada de trabajar con estos sistemas: su simulación mediante software, en una computadora digital común.

Pronto se obtuvieron éxitos muy promisorios. Frank Rosenblatt desarrolló el **Perceptrón** (Rosenblatt 1958), que fue la primera red neuronal artificial especificada con toda precisión y orientada computacionalmente. Como era una máquina que podía aprender y demostrar comportamiento adaptativo complejo, atrajo de inmediato la atención de los investigadores. Su procedimiento de convergencia de aprendizaje fue un avance definitivo sobre la teoría de Hebb. Asimismo, Rosenblatt desechó el enfoque de teóricos anteriores, que veían al cerebro como una computadora lógica.

En vez de ello, lo consideró como un asociador y clasificador, cuya misión era asociar respuestas de clasificación a estímulos específicos. En 1962 Rosenblatt publicó su libro *Principles of Neurodynamics* (Rosenblatt 1962) en el que presentó formalmente el Perceptrón como modelo para construir Redes Neuronales Artificiales.

Los perceptrones se aplicaron rápidamente a resolver problemas tales como la predicción climatológica, la interpretación de electrocardiogramas y otros. Tal parecía que se había hallado la clave para comprender el funcionamiento cerebral, emulando las Redes Neuronales naturales mediante redes complejas de perceptrones.

Sin embargo, pronto se comprobó que las redes con una capa de perceptrones eran incapaces de resolver problemas tan simples como la simulación de una compuerta lógica de tipo **O** exclusivo y, tras una investigación sobre las limitaciones de los perceptrones, Minsky y Pappert publicaron el libro *Perceptrons* (Minsky & Pappert 1969) donde se hacían patentes estas limitaciones.

Como consecuencia, los fondos para nuevas investigaciones se congelaron y la mayoría de los investigadores reorientaron su objeto de estudio. Sólo un par de investigadores aislados como Kohonen en Finlandia, Stephen Grossberg y James Anderson en Estados Unidos, continuaron sus esfuerzos en este campo, dando lugar lentamente a que, a través de los años, (Kohonen 1972), (Anderson 1972) y (Grossberg 1976), emergiera un nuevo cuerpo teórico alrededor de las Redes Neuronales multicapas, que superó las limitaciones encontradas por Minsky y dio nuevo impulso al desarrollo de Redes Neuronales Artificiales.

### 3.1.2 DEFINICIONES

Quizás, la definición de RED NEURONAL más rigurosa sea la debida a Hecht Nielsen [ Hecht-Nielsen (1988) ]:

“Una *red neuronal* es una estructura de procesamiento de información paralela y distribuida, formada por elementos de procesamiento interconectados mediante canales unidireccionales de información denominados conexiones. Cada elemento de procesamiento tiene una conexión de salida con diferentes ramas (tantas como sea necesario) portadoras de la misma señal.

Esta señal de salida será de un tipo matemático cualquiera. Todo el procesamiento que se hace en un elemento debe ser completamente local, por ejemplo: dependerá solo de los valores actuales de las entradas al elemento y de posibles valores almacenados en memoria local”.

Una definición más simple, pero menos rigurosa, es la de Simpson [ Simpson, 1987]: “Una red neuronal es un grafo dirigido no lineal con pesos en las ramas que es capaz de almacenar patrones cambiando los pesos de las ramas y de reconocer patrones con entradas incompletas y desconocidas”.

Los elementos clave de casi todas las definiciones son: Representación distribuida, Operaciones locales, Procesamiento no lineal y, por supuesto, formando la columna vertebral las matemáticas. Las matemáticas más utilizadas incluyen ecuaciones diferenciales, sistemas dinámicos, álgebra lineal, probabilidad y estadística.

### 3.1.3 LA NEURONA ARTIFICIAL.

La neurona artificial es una unidad procesadora con cuatro elementos funcionales: (fig.3.1)

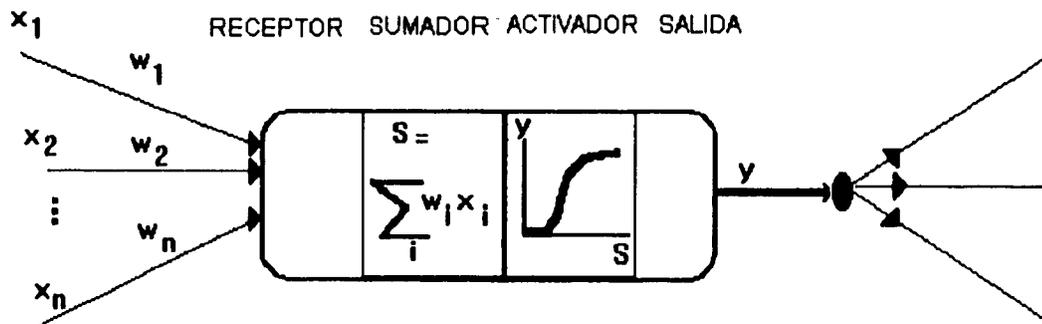


FIG.. 3.1 Esquema de una Neurona Artificial.

1.- El elemento receptor, a donde llega una o varias señales de entrada  $x_i$ , que generalmente provienen de otras neuronas y que son atenuadas o amplificadas cada una de ellas con arreglo a factor de pesos  $w_i$  que constituye la conectividad entre la neurona fuente de donde provienen la neurona de destino en cuestión.

2.- El elemento sumador, que efectúa la suma algebraica ponderada de las señales de entrada, ponderándolas de acuerdo con su peso, aplicando la siguiente expresión:

$$S = \sum w_i x_i \quad (3.1)$$

3.- El elemento de función activadora, que aplica una función no lineal de umbral (que frecuentemente es una función escalón o una curva logística) a la salida del sumador para decidir si la neurona se activa, disparando una salida o no.

4.- El elemento de salida que es el que produce la señal, de acuerdo con el elemento anterior, que constituye la salida de la neurona.

Este modelo neuronal es el utilizado en casi todas las Redes Neuronales artificiales, variando únicamente el tipo de función activadora.

A continuación se presentarán los modelos más simples de Redes Neuronales Artificiales.

### 3.1.4 EL PERCEPTRON

El **Perceptrón**, desarrollado por Rosenblatt (1958), consiste en una neurona procesadora, con sus elementos de entrada, sumador, activador y de salida, a la cual llegan señales de entrada  $x_i$ ,  $i = 1, 2, \dots, n$  cada una con una  $a$  través de una línea con conductividad o peso asociado  $w_i$ . El elemento sumador efectúa entonces una suma ponderada de las entradas, en tanto que el activador emplea una función escalón de umbral: si la suma ponderada es mayor o igual a un valor de umbral  $U$ , da una salida  $y$  de tal manera que:

$$\begin{aligned} y &= 1 && \text{si } S > U \\ y &= 0 && \text{si } S < U \end{aligned}$$

La red neuronal más simple construida con perceptrones tiene dos capas: *una capa receptora de entrada*, en la que la salida de cada neurona reproduce simplemente su entrada (**fig.3.2**) y *una capa de salida* formada por perceptrones como los descritos, totalmente conectados con la capa de entrada, a través de líneas de comunicación con conductividades o pesos ajustables

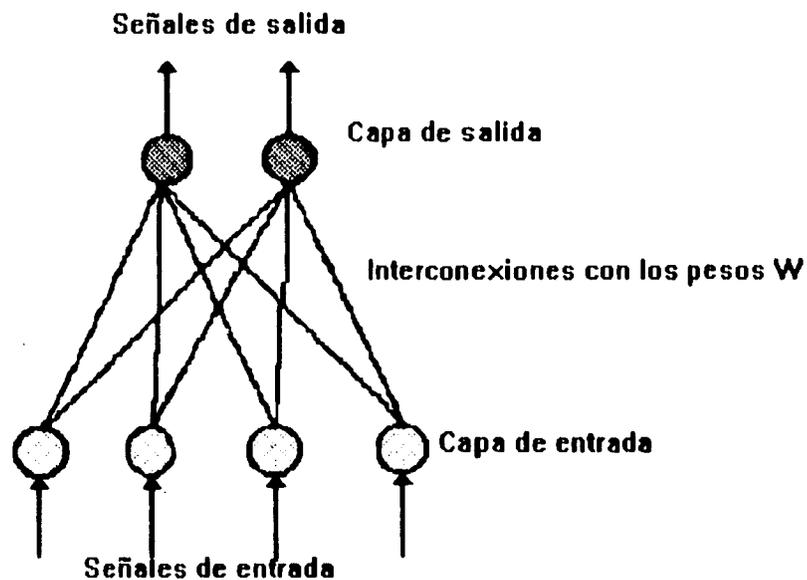


Figura 3.2 Perceptron de dos capas

Así, cada neurona de entrada está conectada con cada neurona de salida a través de una línea de comunicación con una conductividad o pesos ajustables. La ley de aprendizaje del perceptrón ajusta estos pesos, de manera que se obtenga con mayor probabilidad la salida deseable correspondiente a un cierto conjunto de entradas.

El perceptrón es entrenado presentándole un conjunto de *patrones de entrenamiento* en forma repetida. Cada patrón de entrenamiento es una pareja formada por un vector de entrada  $x$  y su vector de salida  $y$  deseable. La dimensión del vector de entrada es igual al número de neuronas de la capa de entrada, en tanto que la dimensión del vector de salida es igual al número de neuronas de la capa de salida.

Al presentarle el vector de entrada al perceptrón, sus neuronas de entrada lo asumen. Las salidas de la red se comparan con el vector de salida  $y$  y la diferencia obtenida se utiliza para reajustar los valores de los pesos  $w$  de las interconexiones.

Este reajuste se hace de modo que sea más probable que la red dé la respuesta apropiada la siguiente vez. El entrenamiento prosigue hasta que todas las respuestas de la red se aproximan en forma aceptable a las deseadas.

Para el reajuste de los pesos, existen diferentes reglas propuestas posteriormente por diferentes autores (Duda & Hart 1973 ; Rosenblatt 1962), basadas en la Regla Delta de Widrow y Hoff (1960). . En una de las más sencillas, el nuevo peso  $w_1$  es igual al peso anterior  $w_0$  más una cantidad proporcional a la diferencia entre la salida deseada  $t$  y la salida real  $y$ .

$$W_1 = W_0 + \eta ( t - y ) \quad (3.2)$$

donde  $\eta$  es una constante de proporcionalidad menor que la unidad que se llama *razón de aprendizaje*.

Si el vector de entrada es de ceros y unos, hay una fórmula derivada en la que la razón de aprendizaje se multiplica por la entrada correspondiente  $x$ . De este modo, el peso se modifica solo cuando la entrada vale 1, es decir, cuando está activa. Así, la fórmula anterior queda en la forma:

$$W_1 = W_0 + x \eta ( t - y ) \quad (3.3)$$

De acuerdo con estas fórmulas, el nuevo peso es mayor que el anterior, si la salida deseada es mayor que la actual y menor en el caso contrario.

Antes de comenzar el entrenamiento, los pesos se fijan aleatoriamente. Durante el entrenamiento, los patrones de entrenamiento se presentan a la red una y otra vez (a veces cientos y hasta miles de veces), hasta que los pesos ya no se modifican. En este caso se dice que la red ha convergido, en cuyo caso o ha aprendido con éxito o se declara incapaz de aprender todas las respuestas correctas. El desempeño de la red se puede medir mediante *el error cuadrático promedio*, dado por la expresión: (Dayhoff 1990)

$$e = \sqrt{\frac{\sum_j \sum_p [t_{jp} - x_{jp}]^2}{n_p n_o}}$$

(3.4)

donde:

$n_p$  = número de patrones de entrenamiento.

$n_o$  = número de neuronas en la capa de salida.

La primera sumatoria se toma sobre todos los patrones en el conjunto de entrenamiento y la segunda sumatoria se toma sobre todas las neuronas de salida.

Las limitantes más grandes del perceptrón es que, aunque puede construirse Redes Neuronales de varias capas con él, no permite más que una sola capa de pesos adaptativos y además sus neuronas de salida solo pueden clasificar *patrones linealmente separables*, es decir, patrones que puedan ser separados en clases mediante una sola línea (fig.3.3).

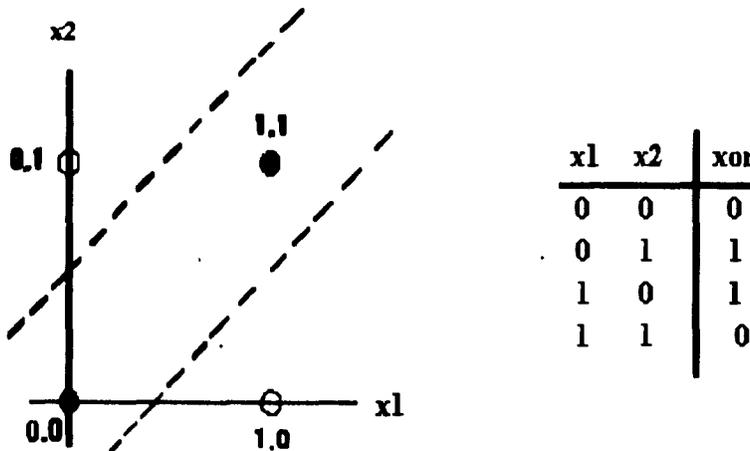


Figura 3.3 Función exclusivo (xor) no linealmente separable con una línea

La función **O** exclusivo es un ejemplo clásico de un problema de clasificación de patrones que no es linealmente separable, puesto que no es posible dibujar una sola recta que agrupe de un lado las entradas con salida 1 y del otro las entradas con salida 0.

3.1.5 LA ADALINE.

La Adaline es un modelo de neuronas desarrollado por Bernard Widrow y Marcian E. Hoff (Widrow & Hoff 1960). Su nombre proviene de Adaptive Lineal Element, cuando los modelos neuronales cayeron en desgracia.

La Adaline es análoga a un perceptrón, ya que calcula la suma ponderada de sus entradas, siendo sus diferencias las siguientes:

a) A las entradas que provienen del exterior o de otras neuronas, se les agrega otra entrada adicional  $x_0$  con un valor constante de bias o polarización, que en general es igual a +1 y que, modulada por su peso  $w_0$ , que representa el umbral de disparo de la neurona.

b) La señal de salida +1 o -1, según que la suma ponderada  $S$  sea mayor o menor que el umbral de disparo. Para definirla, se utiliza una función activadora equivalente a una función escalón (fig. 3.4)

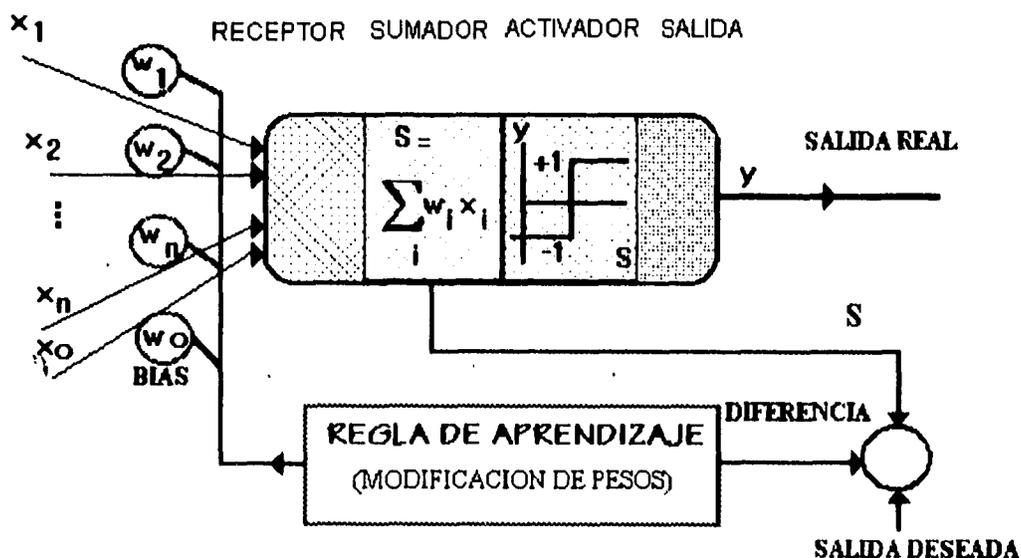


FIG.. 3.4 Esquema de aprendizaje de una Adaline.

c) El error que se utiliza para la regla de aprendizaje es la diferencia entre la salida  $S$  del sumador ponderado y la salida deseada, en vez de tomar la diferencia entre la salida total (que es la salida del cuantizador) y la salida deseada.

La razón de ello es que aquélla es matemáticamente más fácil de manipular, por ser una función cuadrática de los pesos de entrada y, además, el cuadrado del error total es una función monótonica del cuadrado de este error, por lo que minimizarlo significa minimizar el error total. Esto significa, además que la Adaline sigue aprendiendo, aún después de dar una respuesta final correcta, en tanto la salida del sumador no sea igual a la salida total. Este aprendizaje adicional abrevia considerablemente el tiempo total de aprendizaje, en relación con el perceptrón.

La regla de aprendizaje, llamada también la **Regla Delta**, es la siguiente: (Windrow & Hoff 1960)

$$\Delta w_{ij} = Y_i \eta (t_j - S_j) \quad (3.5)$$

donde  $\Delta w_{ij}$  es el incremento en el peso entre la neurona  $i$  y la neurona  $j$ ,  $Y_i$  es la salida de la neurona  $i$ ,  $\eta$  es la tasa de aprendizaje  $t_j$  es la salida deseada de la neurona  $j$  y  $S_j$  la salida del sumador de la neurona  $j$ .

Al definir el mecanismo de aprendizaje, Windrow y Hoff minimizan el cuadrado del error antes definido, utilizando el *método del gradiente máximo*. Al tratarse de una superficie cuadrática, el mínimo encontrado es un mínimo global. Asimismo, los autores muestran que las derivadas parciales definidas al plantear la minimización, son proporcionales al error mencionado, por lo que la medición de este error permite calcular la dirección hacia adónde moverse para corregirlo.

Es fácil pasar de un elemento a una red de varios elementos, en virtud de que la ley de aprendizaje es local, esto es, una neurona dada sólo debe tener información respecto a su propia entrada y salida, sin preocuparse por las demás.

### 3.1.6 LA MADALINE

La Madaline (*Mlti-layer Adaline*) también fue desarrollada por Widrow. Es una red neuronal donde un número de adalines proporcionan las entradas a un procesador madaline, que da una salida igual a la de la mayoría de las adalines de entrada (fig.3.5).

Esto es, la madaline hace un conteo de los "votos" y le da la razón a la mayoría. En esta forma, si la mayoría de las adalines tienen una salida igual a +1 (o -1), la madaline también dará +1 (o -1).

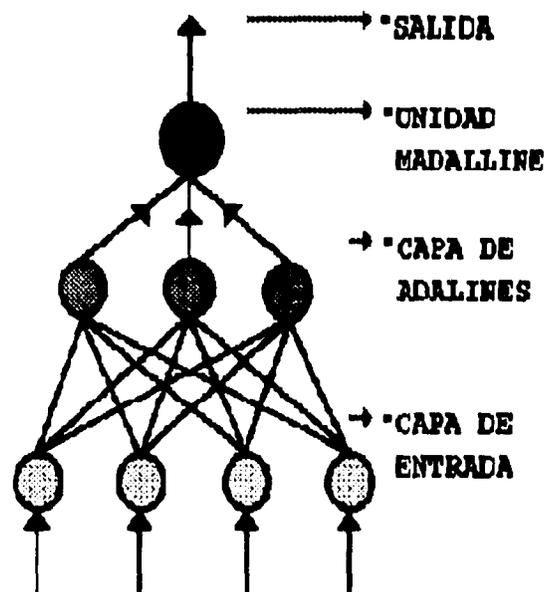


FIG.. 3.5 Sistema Madaline.

El entrenamiento de una Madaline es similar al de una Adaline. Durante el aprendizaje, si la salida de la Madaline es igual a la deseada, nada cambia. Si la salida es diferente, el peso de la adaline cuya suma fue más próximo a cero en la dirección equivocada es modificado. Por lo tanto, la regla asigna mayor responsabilidad a la adaline que más fácilmente puede asumirla. Solo una adaline modifica sus pesos: los de las otras permanecen sin cambio.

## 3.2 MODELOS DE APRENDIZAJE

### 3.2.1 TIPOS DE APRENDIZAJE

Una RNA aprende de experiencias. El proceso usual de aprendizaje involucra tres tareas (Figura 3.6):

1. Calcula las salidas.
2. Compara las salidas con los valores deseados.
3. Ajusta los pesos y repite el proceso.

El proceso de aprendizaje comienza poniendo los pesos por una u otra regla o al azar. La diferencia entre la salida actual y la salida deseada ( $u$  o  $u_T$ ), es llamada delta ( $Z$ ). El objetivo es minimizar la delta (o mejor aún reducirla hasta cero). La reducción de delta se hace a través del cambio de los pesos.

El proceso de información con una RNA consiste en el intento de reconocer patrones (reconocimiento de patrones). Durante las etapas de aprendizaje, los pesos de interconexión cambian en respuesta a los datos de enseñanza presentados al sistema.

Una RNA puede calcular la delta de diferentes maneras dependiendo del algoritmo de aprendizaje que sea usado.

Se ha mencionado ya que una de las características fundamentales de las Redes Neuronales es su adaptabilidad y su susceptibilidad de aprendizaje, a través de la modificación de los pesos de las interconexiones entre las diferentes neuronas.

A continuación mostramos un pequeño diagrama de flujo que muestra el funcionamiento generalizado de un algoritmo de aprendizaje:

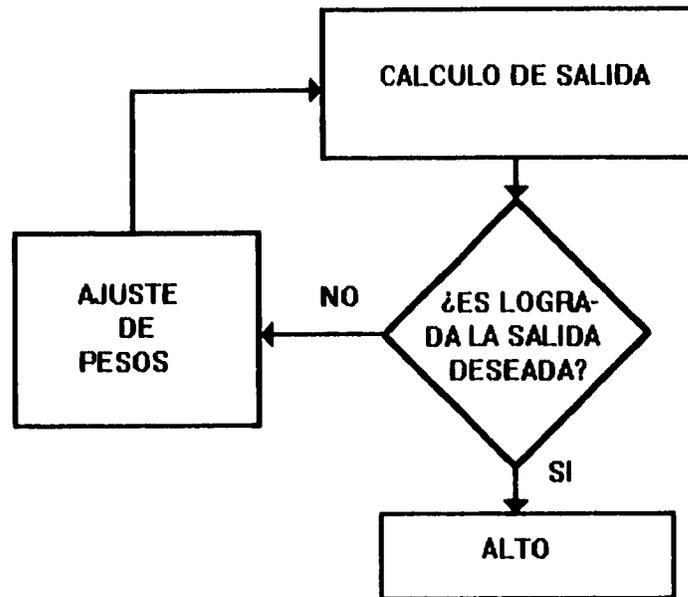


FIG.. 3.6 Proceso de Aprendizaje de una RNA.

Una importante consideración en una RNA es el uso apropiado de algoritmos para aprender. Este tipo de algoritmos es llamado Algoritmos de aprendizaje. Hay dos formas fundamentales de aprendizaje:

**1.- Aprendizaje supervisado**, que requiere la presencia de un tutor externo y una serie de patrones de aprendizaje. El tutor conoce el vector de respuesta correcto ante cada vector de entrada y, con la respuesta real de la red genera un vector de error, que se retroalimenta de esta. La red con base en el vector de error, actualiza los pesos de sus interconexiones de manera que el error tienda a desaparecer.

Con un número suficiente de sesiones de entrenamiento, la red converge, produciendo las respuestas deseadas. Este tipo de aprendizaje es el aplicado tanto en el Perceptrón, como en la Adaline, la madaline y las Redes de Grossberg.

**2.- Aprendizaje no supervisado**, que utiliza datos de entrenamiento no etiquetados previamente y no necesita tutor externo. Los datos son presentados simplemente a la red, que de acuerdo con ellos configura cúmulos internos que comprimen los datos de entrada en cierto número de categorías de clasificación. Este tipo de aprendizaje es el aplicado en las Redes de Resonancia adaptativa y en las Redes de Kohonen.

### 3.2.2 APRENDIZAJE POR RETROPROPAGACION DEL ERROR.

Es importante mencionar los diferentes tipos de aprendizaje que hay actualmente para la enseñanza en Redes Neuronales Artificiales, y son los siguientes:

- Aprendizaje por Retropropagación del error.
- Aprendizaje en Redes de Contrapropagación.
- Aprendizaje mediante métodos estadísticos:
  - Entrenamiento de Boltzman.
  - Entrenamiento de Cauchy.

Esto es con el fin de que usted como lector este enterado de ellos. El algoritmo de aprendizaje por Retropropagación del error es el más ampliamente utilizado. Es una técnica muy popular que es fácilmente de implementar. Una red con retropropagación incluye una o más capas ocultas. Externamente provee patrones correctos que son comparados con la salida de la red durante el entrenamiento (esto es un entrenamiento supervisado), y la retroalimentación es usada para ajustar los pesos hasta que todos los patrones son correctamente categorizados por la red.

El procedimiento para ejecutar el algoritmo de aprendizaje es como sigue: se inicializan los pesos y otros parámetros dándoles valores aleatorios. Entonces lee el vector de entrada y la salida deseada. Calcula la salida actual vía calculando hacia adelante a través de las capas; cambia los pesos por cálculos hacia atrás desde la salida de la capa a través de las capas ocultas. El procedimiento se repite para todos los vectores de entrada hasta que las salidas deseadas y las actuales coinciden dentro de una tolerancia predeterminada. Se da la cantidad de los cálculos para una iteración, una red grande puede tomar un largo tiempo de entrenamiento. Investigaciones actuales están apuntando a desarrollar algoritmos que mejoren este proceso.

El modelado de sistemas inteligentes mediante Redes Neuronales decayó durante algunos años, hasta que en 1986 David Rumelhart, Geoffrey Hinton y Ronald Williams en el libro *Parallel Distributed Processing: Explorations in the Microstructures of Cognition* (Rumelhart, Hinton & Williams 1986) introdujeron el **algoritmo de retropropagación**, que permitía ajustar los pesos de varias capas de interconexiones, con arreglo a las diferencia entre la salida real y la deseada.

Una vez publicado el libro, se encontró que ya anteriormente David Parker en 1982 había desarrollado el algoritmo de retropropagación (Parker 1982). Poco tiempo después se encontró que incluso desde 1974 Paul Werbos había propuesto este algoritmo en su tesis doctoral. (Werbos 1974). Por consiguiente, Rumelhart y compañía se hubieran ahorrado mucho trabajo si hubieran sabido de estos trabajos previos.

El algoritmo de retropropagación del error se explicará a continuación con base en una red neuronal de tres capas: la capa de entrada, la capa intermedia u oculta y la capa de salida.

La capa de entrada es la única que recibe señales de entradas externas, en tanto que la capa de salida es la que da las respuestas observables de la red. Entre las capas de entrada y de salida se encuentra la capa intermedia u oculta, formada por neuronas procesadoras.

La red está completamente interconectada, es decir, cada neurona de la capa intermedia está conectada con cada una de las neuronas de entrada y con cada una de las neuronas de salida. Sin embargo, esto no es imprescindible. Tampoco se permiten interconexiones entre neuronas de la misma capa.

El algoritmo de retropropagación del error involucra dos fases: a) la fase de propagación hacia adelante y la fase de propagación hacia atrás.

En la fase de propagación hacia adelante, se le presenta a la red un vector de entrada a través de sus neuronas de entrada. Estas neuronas dan una respuesta igual a su entrada y dirigida a cada una de las neuronas  $j$  de la capa intermedia. Cada una de estas, al recibir esas entradas, calcula su suma ponderada  $S$  de acuerdo con los pesos de las interconexiones de entrada. La suma ponderada  $S$  pasa al elemento activador no lineal, que presenta como salida una función sigmoide  $f(S)$  de esa suma ponderada  $S$ . Esta salida constituye, a su vez, una entrada para cada neurona  $i$  de la capa de salida que, a su vez calcula una suma ponderada de sus entradas, la cual pasa por su activador no lineal para constituir la salida de la red. Con esto termina la fase de propagación hacia adelante.

La fase de propagación hacia atrás comienza con la comparación de la salida real de la red con la salida deseada, calculándose el error  $\delta_j$  para cada neurona  $j$  de salida. Este error es igual a:

$$\delta_j = (t_j - y_j) f'(S_j) \quad (3.6)$$

donde:

$t_j$  = la salida deseada de la neurona  $j$

$y_j$  = la salida real de la neurona  $j$

$f'(x)$  = la derivada de la función activadora sigmoide  $f(x)$  en el punto  $x$ .

$S_j$  = la suma ponderada de las entradas a la neurona  $j$ .

Después, se calcula el valor del error  $\delta_j$  para cada neurona intermedia  $j$  en la forma siguiente:

Se hace la suma ponderada de los errores de salida, utilizando los procesos de interconexión  $w_{kj}$  entre esta neurona  $j$  intermedia y cada neurona  $k$  de salida, y esta suma ponderada se multiplica por la derivada  $f'$  de la función activadora sigmoideal  $f(S)$ , con el fin de "escalar" el error y forzar una corrección, más grande cuando la suma ponderada queda cerca de la zona de pendiente máxima de la curva sigmoideal

$$\delta_j = ( \sum d_k w_{kj} ) f' (S_j) \quad (3.7)$$

A continuación se calcula el incremento  $\Delta w_{ji}$  a los pesos, de acuerdo con la formula:

$$\Delta w_{ji} = \eta \delta_j a_i \quad (3.8)$$

donde:

- $\delta_j$  = el valor de error de la neurona posterior
- $a_i$  = el valor de salida real de la neurona anterior
- $\eta$  = la tasa de aprendizaje

Esta regla se conoce también como la **regla delta generalizada** (Rumelhart & McClelland 1986).

En muchos casos es deseable dar a cada neurona una polarización o bias igual a +1, pero con peso ajustable por entrenamiento, con el fin de neutralizar el efecto de inicio lento de la función logística, con lo que se acelera la convergencia del proceso de aprendizaje. Otras ocasiones la convergencia se acelera introduciendo un término en las ecuaciones de ajuste, llamado momentum, y que es proporcional al cambio de peso sufrido en la interacción anterior. La expresión (21) queda entonces en la forma:

$$\Delta w_{ji} (n) = \eta \delta_j a_i + \alpha \cdot \Delta w_{ji} (n-1) \quad (3.9)$$

El valor de  $\alpha$  varia entre 0 y 1, pero comúnmente se adopta un valor alrededor de 0.9.

### 3.3 DESARROLLO DE REDES NEURONALES ARTIFICIALES.

Aunque el desarrollo de procesos de RNAs es similar a la estructura de metodologías de diseño de sistemas de información basadas en computadoras tradicionales, algunos pasos son únicos en las aplicaciones de Redes Neuronales Artificiales o tienen consideraciones adicionales. En el proceso descrito aquí, nosotros asumimos que los pasos preliminares del desarrollo de sistemas, de este tipo es determinando los requerimientos de la información y llevar a cabo el análisis del proyecto. Este tipo de pasos es genérico hacia cualquier sistema de información. Como se muestra en la **Figura 3.7**, el proceso de desarrollo para la aplicación de una RNA se puede dar en diez pasos.

- ° **Paso 1.** En este paso se coleccionan los datos que van a servir para construir la red.
- ° **Paso 2.** Se separan los datos en conjuntos de entrenamiento y conjuntos de verificación.
- ° **Paso 3.** Se define la estructura de la red ( Tipo de Red ).
- ° **Paso 4.** Se selecciona el algoritmo de aprendizaje.
- ° **Paso 5.** Se inicializan los pesos y los parámetros de la red.
- ° **Paso 6.** Se transforman en datos en el tipo y formato requerido por la RNA. Esto puede ser hecho por medio de software para el procesamiento de datos. El almacenamiento de datos y técnicas de manipulación son procesos necesarios para diseñar conveniente y eficientemente el reentrenamiento de la RNA cuando sea necesario.
- ° **Paso 7.** Comienza el entrenamiento, determina y revisa los pesos.
- ° **Paso 8.** Termina y prueba. En los dos pasos anteriores se entrenan y prueban en un proceso iterativo la presentación de los datos de entrada y los datos de salida deseados hacia la red. La red calcula las salidas actuales y ajusta los pesos hasta que las salidas actuales van a la par con las deseadas.
- ° **Paso 9.** En el proceso un conjunto de pesos se hace estable. Ahora la red reproduce las salidas deseadas. La red esta lista para usarse como un sistema independiente o como parte de otro sistema de software.
- ° **Paso 10.** Se valida la red presentándole los datos de verificación.



FIG.. 3.7 Diagrama de flujo del proceso de desarrollo de una RNA.

**Colección de datos y preparación.**

Los primeros dos pasos en el proceso de desarrollo de una RNA involucran la colección de datos y separación de los mismos dentro de conjuntos de entrenamiento y conjuntos de prueba. Los casos de entrenamiento son usados para ajustar los pesos. Los casos de prueba son usados para la validación de la red.

La estructura de la RNA y el algoritmo de aprendizaje determinan el tipo de datos, que pueden ser del tipo binario o continuo.

### **Estructura de Redes.**

Diferentes modelos de RNA y algoritmos han sido desarrollados y estudiados. Pero se puede decir que hay tres arquitecturas representativas que son:

- Sistemas de Memoria Asociativa.
- Capas ocultas.
- Doble capa.

**Sistemas de Memoria Asociativa.** Memoria asociativa se refiere a la habilidad de reproducir situaciones completas desde una información incompleta. Estos sistemas correlacionan los datos de entrada con la información almacenada en memoria. La información puede ser desde datos incompletos hasta ruido a la entrada. Los Sistemas de Memoria Asociativa pueden detectar similitudes entre las nuevas entradas y los patrones almacenados.

**Capas ocultas.** Aplicaciones prácticas complejas requieren una o más capas ocultas entre la capa de entrada y la capa de salida y consecuentemente un gran número de pesos. Mas comercialmente las Redes Neuronales Artificiales incluyen tres, y algunas veces cuatro o cinco capas y cada una contiene desde 10 hasta 1,000 elementos de proceso (neuronas). Algunas RNAs experimentales utilizan millones de elementos de procesamiento. El uso de más de tres capas es raro y más en sistemas comerciales porque el cálculo crece con cada capa y el incremento se hace exponencial.

**Estructura de doble capa.** Este tipo de estructura no requiere del conocimiento de un preciso número de clases en el entrenamiento de datos. Se utiliza el método de aproximación por retropropagación para ajustar los parámetros .

### **Preparación.**

En la preparación para el entrenamiento es necesario primero decidir que algoritmo de aprendizaje se usará. Esta decisión está relacionada con el tipo de herramientas desarrolladas en software con que se cuenta. Seleccionar un algoritmo de aprendizaje es necesario pero no suficiente; antes de comenzar el entrenamiento, varios parámetros tienen que ser determinados. Unos parámetros determinan la velocidad de aprendizaje. Esta puede ser alta o baja. Otros parámetros están en un valor de umbral que determinan la forma de la salida.

La elección de la estructura de la red ( número de nodos y capas ), además de la selección de las condiciones iniciales de la misma, determinan la cantidad del tiempo para el entrenamiento. Por lo tanto, esas elecciones son importantes y requieren de cuidadosas consideraciones en el inicio del proceso.

La última tarea de preparación es transformar el entrenamiento y probar los datos en el formato requerido por la red y su algoritmo. Este paso es especialmente importante cuando el software desarrollado es usado.

### **Entrenando la Red**

La fase de entrenamiento consiste de la presentación del conjunto de datos a la red, para que los pesos puedan ajustarse y producir la salida deseada para cada entrada. Los pesos son ajustados después que cada vector de entrada es presentado.

### **Implementación de la Red.**

La implementación de la RNA frecuentemente requiere de interfaces propias con otros sistemas de información en una computadora base y entrenamiento de los usuarios. Si la RNA es parte de un sistema muy grande, la red podría necesitar interfaces convenientes hacia otros sistemas de información, dispositivos de entrada y salida, y manuales de operación para usuarios. Los sistemas pueden necesitar la manipulación de entradas y salidas de subsistemas como digitalizadores de señales y módulos de conversión. Buena información y el entrenamiento del usuario son necesarios para asegurar una exitosa integración dentro de las operaciones principales. Un conveniente procedimiento tiene que ser planeado para actualizar los conjuntos de entrenamiento y la inicialización periódica de reentrenamiento de la red. Esto incluye la habilidad de reconocer e incluye nuevos casos que son descubiertos cuando el sistema es usado rutinariamente.

### ***Paradigmas de la Neurocomputadora:***

En la construcción de una RNA, se tienen que tomar varias decisiones como:

- Tiempo del entrenamiento y datos de prueba.
- Algoritmos de aprendizaje
- Topología: número de elementos de procesamiento y sus configuraciones
- Función de Transformación a ser usada.
- Velocidad de aprendizaje para cada capa.
- Diagnóstico y validación de herramientas.

**Programando Redes Neuronales Artificiales**

Las Redes Neuronales Artificiales son básicamente aplicaciones de software que necesitan ser programadas. Como cualquier otra aplicación, una RNA puede ser programada con lenguajes de programación, un paquete de programación, o ambos.

Una gran parte de la programación esta dada con algoritmos de entrenamiento y funciones de transferencia y operaciones aritméticas. Esto tiene sentido, y varias docenas de herramientas desarrolladas están en el mercado. Algunas de esas herramientas son similares a los shells de Sistemas Expertos. Contienen ayuda para las herramientas, y cualquiera que sea el trabajo para desarrollar una RNA puede no ser fácil o simple. Otros productos son diseñados para trabajar con Sistemas Expertos como un producto híbrido. Pero tal vez, las herramientas existentes sean limitadas para la aplicación que nosotros le queramos dar a nuestra RNA, por lo tanto, tenemos la opción de utilizar y hacer nuestros propios programas, estos pueden hacerse en lenguaje C y otros. Enseguida mostramos a ustedes los paquetes actuales existentes en el mercado que hacen posible la simulación, entrenamiento y/o interface de una RNA:

**TABLA 3.1**

**TABLA REPRESENTATIVA DE HERRAMIENTAS DESARROLLADAS PARA NEUROCOMPUTADORAS.**

HERRAMIENTA	VENDEDOR	
BrainMaker	California Scientific Software	{ Grass Valley, CA }
ExploretNet	Hecht-Nielsen Neurocomputer Corp.	{ San Diego, CA }
Explorer NeuralWorks: Professional I,II Plus	NeuralWare	{ Pittsburgh, PA }
Plexi	Lucid Inc.	{ Menlo Park, CA }
NeuroShell	Ward Systems Group	{ Frederick, MD }
MacBrain	Neurix	{ Boston, MA }
N-NTEX, N-NET 600 OH }	AI Ware	{ Cleveland,
Nestor Development System	NESTOR Corp.	{ Providence, RI }

### **Hardware de RNA.**

Las aplicaciones de RNA involucra la simulación de software que corre en procesos secuenciales convencionales. Simulando una RNA matemáticamente definiendo los nodos y asignando los pesos a esta; lo uno o lo otro se puede hacer usando un CPU para cada una de las neuronas o bien un CPU para todas las neuronas. Esta simulación puede tomar largos tiempos de procesamiento. Los avances en tecnología de hardware podrían realzar grandemente el funcionamiento a futuro de sistemas de Redes Neuronales Artificiales por la explotación inherente en la ventaja de procesos masivos en paralelo. El mejoramiento de Hardware podría hacer frente a los altos requerimientos para memoria y velocidad de procesamiento y así de esta manera permitir acortar los tiempos de entrenamiento para grandes redes.

Cada elemento de procesamiento calcula las salidas de los nodos desde los pesos y las señales de entrada desde otros procesadores. En sí, la red de neuronas puede almacenar información que puede ser llamada para interpretar y clasificar futuras entradas a la red.

El trabajo de cálculo de una RNA puede consistir de cientos de miles de manipulaciones. Para incrementar la velocidad en los cálculos cuando computadoras convencionales son usadas, una de tres aproximaciones siguientes es aplicable:

1. **Maquinas rápidas.** Por ejemplo una máquina con el procesador 80486 suplementado por un coprocesador matemático puede facilitar el trabajo. Pero no mucho ( entre dos a diez veces más rápido ).
2. **Chips de Neuronas.** Hoy en día existen chips especiales que ejecutan cálculos muy rápidos, pero no pueden ser usados para entrenar la red. Así que, es necesario "entrenar apagado el chip". El problema es esperar a ser corregido pronto. La idea de un chip, es la de implementar la estructura de datos de la red en el chip - un chip analógico, como el de Intel 80170 que es una RNA electrónicamente entrenable, o bien un chip digital . Pero los chips de RNAs aún están en etapa de desarrollo.

3. **Tarjetas de aceleración.** estos son procesadores dedicados que pueden ser añadidos a computadoras convencionales, similares a un coprocesador matemático. Ellos son especialmente diseñados para una RNA, y son muy rápidos. ( por ejemplo un procesador de este tipo puede ser de 10 a 100 veces más rápido que el procesador 80386 de 20 Mhz ). Las tarjetas de aceleración son actualmente la mejor propuesta para la aceleración de cálculos. Algunos ejemplos son el acelerador BrainMaker , tablas Balboa/860, y NeuroBoard. Las tarjetas de aceleración son extremadamente útiles porque reducen el tiempo de entrenamiento, lo cual es muy usual. Por ejemplo una prueba con NeuroBoard muestra en un entrenamiento, la reducción del tiempo de siete minutos a un segundo.

### **3.4 SIMULACION DE REDES NEURONALES CON BrainMaker.**

#### **3.4.1 INTRODUCCION.**

Como se había mencionado anteriormente, actualmente las Redes Neuronales Artificiales todavía requieren de software para poder realizar su aprendizaje. El mercado en esta área no es muy amplio, pero se tienen varios paquetes que nos brindan facilidad de manejo y un no muy alto costo. BrainMaker es un completo sistema para diseñar, construir, entrenar, probar y echar a andar Redes Neuronales Artificiales. Es pues, nuestro propósito dar a usted una información general del funcionamiento de este sistema.

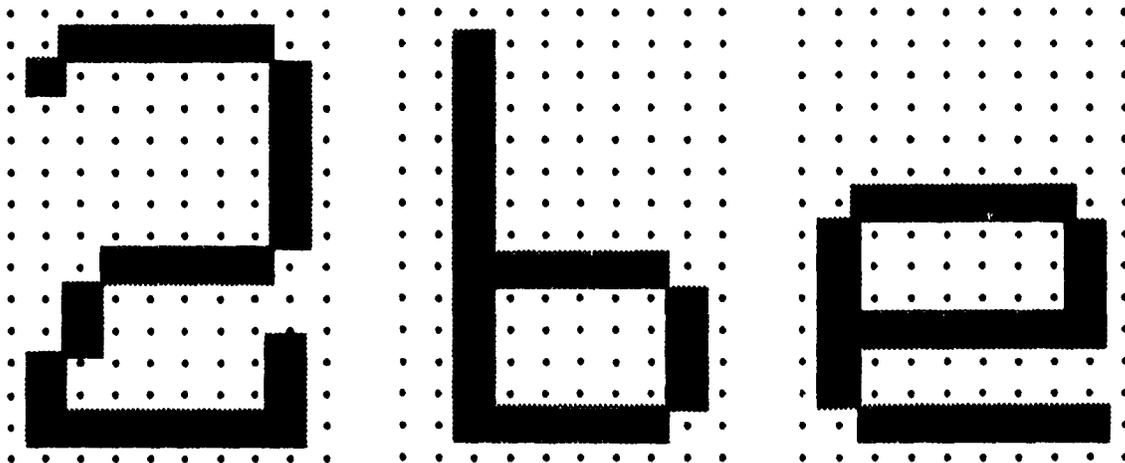
#### **Tipos de Entradas Disponibles en el BrainMaker.**

La entrada del BrainMaker puede ser uno de tres distintos tipos: gráficos, números, y símbolos.

La entrada numérica consiste de una lista de números, los cuales son las entradas hacia la primer capa de neuronas. La entrada simbólica consiste de una lista de símbolos los cuales son representados por cadenas arbitrarias de caracteres. La entrada de gráficos consiste de arreglos de dos dimensiones de elementos intensificados, o pixeles; cada pixel corresponde a la entrada de una neurona.

Aquí hay algunos ejemplos de los distintos tipos de entradas:

Numérico:	0.230	-2.7182	1.0000
Simbólico:	Feliz	Triste	Aburrido
Gráfico:			



Los datos de entrenamiento consisten del alternamiento de entradas (gráficos, números, o símbolos) y el entrenamiento de patrones (también gráficos, números, o símbolos).

**Entradas Numéricas.**

Muchas Redes Neuronales son utilizadas para analizar números. BrainMaker puede leer listas de números, los cuales son ordenados en líneas y columnas. Los números individuales son todos de libre formato; los siguientes números son entradas numéricas válidas para el BrainMaker:

0    1    .304    1e2    1.234e56    567.8e-99

BrainMaker utiliza doble precisión en estos números. El número más grande que puede ser leído por BrainMaker está alrededor de +10308 y el número más pequeño que puede ser leído por BrainMaker está alrededor de +10-308. La mayoría de los problemas entran confortablemente dentro de este rango.



### ***Traslación de Gráficos de Entrada.***

Cuando BrainMaker está leyendo gráficos desde un conjunto de entrenamiento, los caracteres en la pintura en su mayoría son trasladados dentro de números. Esto es porque la red asimismo entiende entradas numéricas y salidas numéricas. BrainMaker traslada un gráfico dentro de números viendo cada carácter arriba del gráfico en una tabla de traslación, y sustituyendo el número que encuentra. De esta manera un gráfico de 8 por 8 esta convertido a 64 números. La tabla de traslación esta almacenada en un archivo Char.Map.

### ***Decidiendo sobre el número de Neuronas.***

Una vez que se ha decidido sobre el formato del conjunto de entrenamiento; decidiendo sobre el número de neuronas, esta casi hecha la red. Si se están utilizando gráficos como entradas, necesita una neurona de entrada para cada elemento (pixel) del gráfico. Si el gráfico es de 6 por 13 pixeles, necesita  $6 \times 13 = 78$  neuronas de entrada. Si esta usando símbolos como entradas requiere una neurona como entrada única para cada símbolo. En la red acerca de la fruta, necesita una neurona de entrada para cada característica rojo, verde, amarillo, pequeño, mediano, redondo, y alargado, y nos da un total de 7 neuronas de entrada. Si se están utilizando como entradas números, se necesita una neurona para cada columna de números.

La decisión de cuántas neuronas ocultas poner es relativamente arbitraria. Una buena forma de saber es utilizar el promedio de las neuronas de entrada y las neuronas de salida. Otro buen número es 25. Otra aceptable manera de elaborar capas ocultas es hacer una capa con el mismo número de neuronas de entrada y una capa con el mismo número de neuronas de salida. Menos capas ocultas significan menos conexiones, menos capacidad de aprendizaje, y mayor rapidez de entrenamiento y funcionamiento. Generalmente, las capas ocultas pueden no ser pequeñas en la red, ni tampoco pueden ser muy grandes.

Frecuentemente se comprueba que dos o más capas ocultas son mejor que una.

## **3.4.2 APLICACIÓN DE RECONOCIMIENTO DE PATRONES.**

### ***3.4.2.1 RECONOCIMIENTO OPTICO DE CARACTERES NUMERICOS.***

En este ejemplo veremos como construir un archivo de entrada para especificar la aplicación. Deseamos elaborar una Red Neuronal que se capaz de reconocer los dígitos del 0 al 9. La podemos llamar la Red Dígitos. La red está asociada a imágenes (mapa de bits) de caracteres con el código ASCII como representante.

Cuando entrenamos una red, se dan un par de imágenes y símbolos. Las imágenes son las entradas, y los símbolos son los patrones de entrenamiento para esa red. Los 10 símbolos de salida pueden ser asignados cada uno a una neurona de salida. Cuando el gráfico de un dígito en particular es presentado a la red, debe activarse exactamente una neurona de salida.

Durante el curso del entrenamiento, la salida real de la red para cada dato es comparado con el patrón de entrenamiento. En su mayoría el camino para representar una única entrada es con un gráfico. Podemos elaborar solamente el dibujo de cada número y entrenar la red asociando el gráfico del número "0" con el primer símbolo de salida, el gráfico del número "1" con el segundo símbolo de salida, y así sucesivamente.

Podemos dibujar la pintura en un arreglo de 8 por 8 pixeles, para tener un total de 64 pixeles y 64 neuronas de entrada.

Cada patrón de entrenamiento puede ser uno de 10 símbolos únicos. En nuestro caso, solamente necesitamos distinguir cada una de las 10 posibilidades, así como de los símbolos seleccionados. Dentro de las 10 posibilidades disponibles tenemos que asociar cada pintura de entrada con solamente uno de los 10 símbolos, la red aprende a clasificar correctamente estos símbolos. Por notación convencional podemos utilizar solamente los números "0", "1", ..., "9", como nuestros símbolos nombres, por ejemplo "cero", "uno", ..., "nueve", o bien otro tipo de palabras o simples caracteres como "hola", "qqr", etc..

Aquí se presentan los primeros tres datos de entrada como gráficos de pixeles:

facts		
<pre> ..... ..X X X X ..X.....X ..X.....X ..X.....X ..X.....X ..X.....X ..X.....X ..X X X X 0                 </pre>	<pre> ..... .....X... ....X X... .....X... .....X... .....X... .....X... .....X... ..X X X X 1                 </pre>	<pre> ..... ..X X X X ..X.....X .....X... .....X X X .....X... .....X...X ..X X X X 2                 </pre>

El primer dato es una imagen del carácter "0", seguido por el símbolo 0. El hecho es que la red debe aprender a asociar las imágenes de "0" con el carácter 0 de ASCII.

Definiremos los símbolos de salida, en nuestro caso los símbolos son los números del 0 al 9. Para representar los 10 número, se utilizan 10 símbolos, que corresponden a 10 neuronas de salida.

Podemos observar como aprende la red, y decirle que despliegue las entradas y las salidas como están hechas.

Asimismo usaremos 25 neuronas ocultas (se selecciona el numero 25 arbitrariamente).

Con esto se está listo para iniciar el aprendizaje, corriendo BrainMaker. BrainMaker ahora despliega una caja de entrada a la izquierda de la pantalla, y una caja de salida cerca del centro. Para iniciar con el entrenamiento, se selecciona Train Network desde el menú que se encuentra en la parte superior de la pantalla.

Cada entrada aparece como una imagen en la caja de entrada, mientras el patrón de entrenamiento de esa entrada aparece como un bloque brillante abajo de uno de los 10 numero en la caja de salida. La salida actual de la red aparece justamente abajo del patrón de entrenamiento. El supervisor necesita indicar a la red si su respuesta fue correcta y si no cuál es la correcta.

Cuando en el entrenamiento todas las salidas de los datos son correctos, dentro de la tolerancia, que es de 0.1 por default, interrumpimos el entrenamiento.

El funcionamiento de BrainMaker realmente no es sofisticado y tiene la gran ventaja de poseer varias herramientas que nos facilitan su manejo. Pero la mejor manera de entenderlo bien es introducirse en él.

### **3.5 CONCEPTOS BASICOS DE ELECTRONICA PARA LA IMPLEMENTACION DE REDES NEURONALES ARTIFICIALES.**

Hoy en día los cambios en todos los aspectos de la vida se dan a cada instante, y la electrónica como la computación no se quedan atrás.

La implementación de Redes Neuronales Artificiales puede ser muy diversa dependiendo del tipo de Red que se quiera aplicar, el tipo de aprendizaje, las características de los patrones a reconocer, así como de los conocimientos que el ingeniero en electrónica tenga. Lo anterior no debe de alarmarnos, pues, como todos sabemos todo gran conocimiento parte de bases sencillas para llegar a lo complejo, y las Redes Neuronales Artificiales no son la excepción. Para esto, volvamos nuevamente al concepto elemental de una Neurona Artificial. Sabemos que cuenta con cuatro elementos que son: el receptor, el sumador, el activador y la salida.

Mostrando a ustedes un nuevo esquema de la neurona tenemos lo siguiente:

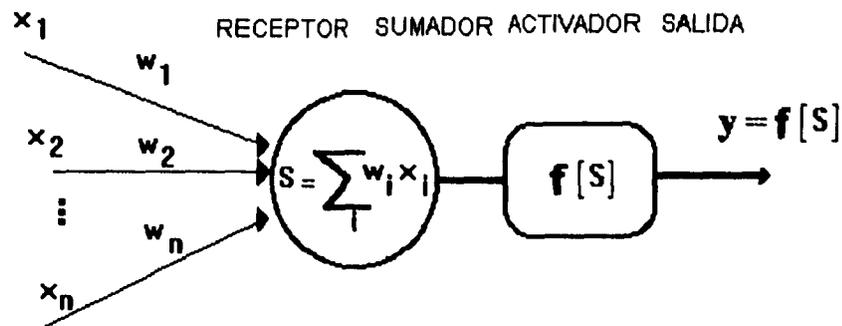


FIG.. 3.8 Esquema de una Neurona Artificial.

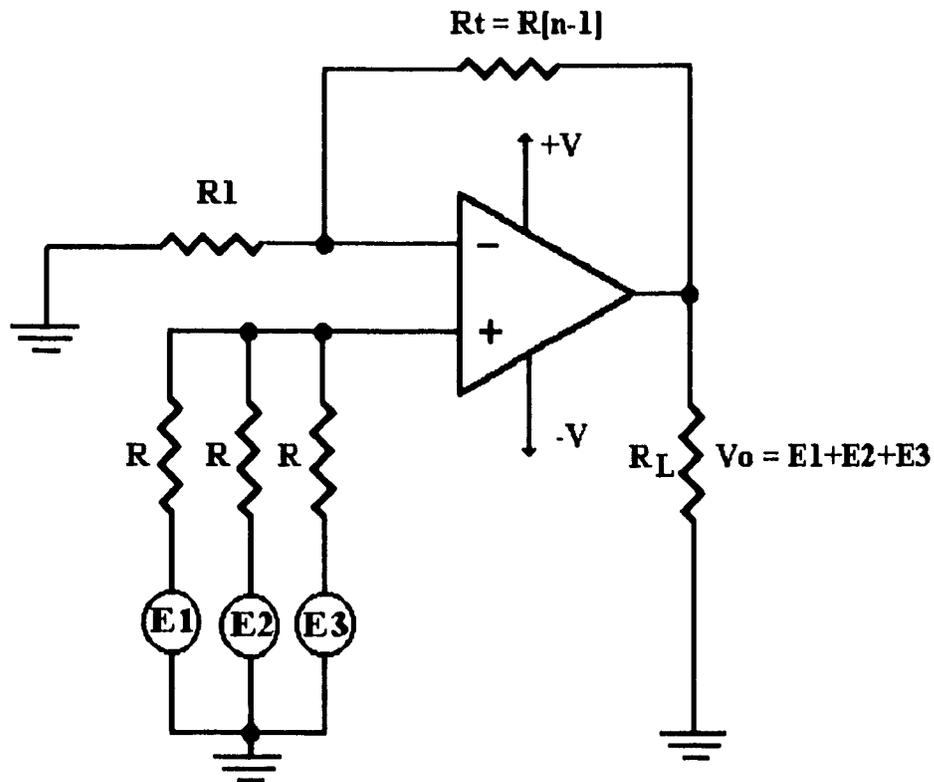
Ahora vayamos por partes. En el receptor las señales de entrada a la neurona son  $X_i$ , que generalmente provienen de otras neuronas y que son amplificadas o atenuadas cada una de ellas con un arreglo a un factor de peso  $W_i$ .

Como podemos observar el factor de peso físicamente se puede manejar de manera sencilla, utilizando potenciómetros, amplificadores operacionales, o bien llegar a utilizar algo más sofisticado como los DACs, que son resistencias controladas digitalmente.

El sumador efectúa la suma algebraica ponderada de las señales de entrada, aplicando la siguiente expresión:

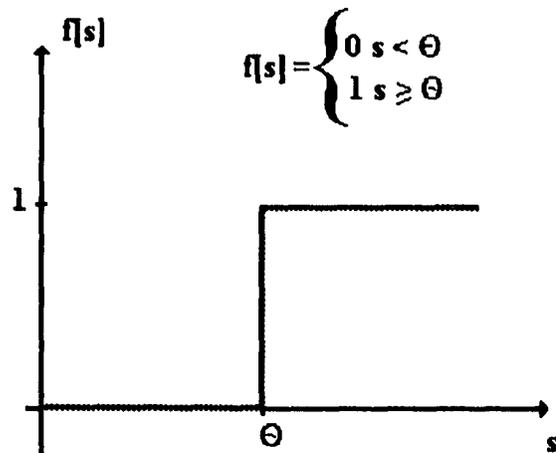
$$S = \sum W_i X_i \quad (3.10)$$

Electrónicamente la suma de señales se puede realizar con el simple arreglo de un amplificador operacional, como se muestra en la **Figura 3.9**:

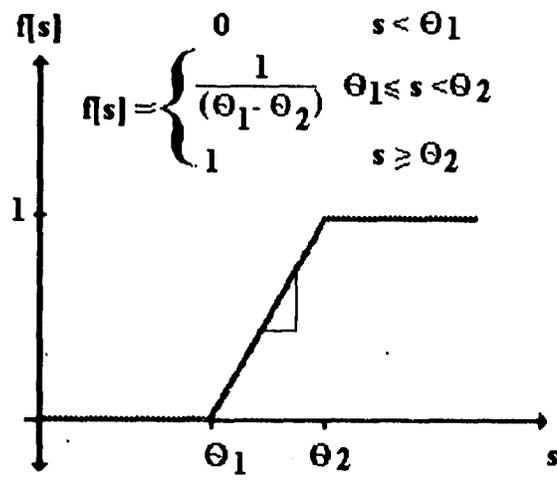


**FIG.. 3.9** Sumador no inversor de tres entradas ( $n=3$ )

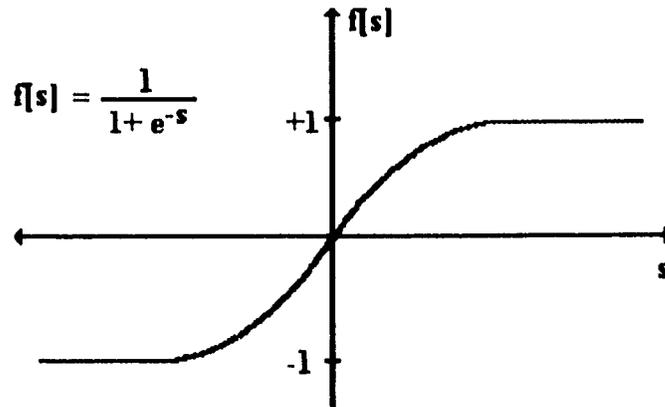
En seguida tenemos el elemento activador, como observamos la función suma calcula la simulación interna, o nivel de activación, de la neurona. Basado sobre este nivel la neurona puede o no producir una salida. La relación entre el nivel de activación y la salida puede ser lineal o no lineal. Este tipo de relación es expresada por funciones de transformación también llamadas funciones activadoras, y hay varias de ellas. Están la función escalón, la función rampa o más frecuentemente se usa una función sigmoide. Las señales mencionadas se muestran en la siguiente figura:



**Función Escalón**



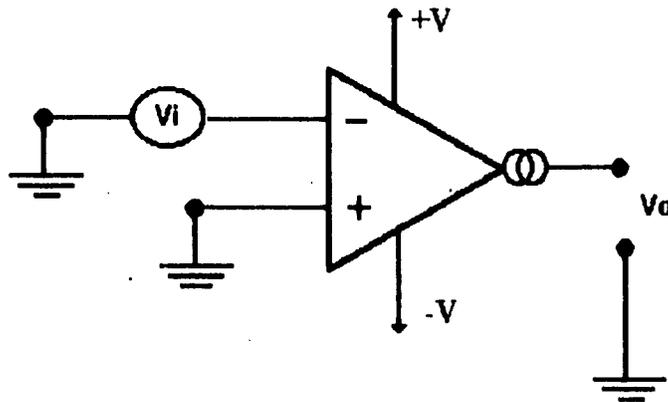
**Función Rampa**



**Función Sigmoide**

**FIG. 3.10 Funciones Activadoras.**

es importante señalar que la señal más utilizada es la sigmoide debido a que es derivable en todos sus puntos. Ahora bien para poder obtener una función escalón o una rampa electrónicamente se pueden utilizar circuitos integrados LM555, Amplificadores Operacionales, o multivibradores. Para la función sigmoide se requiere de un dispositivo especial, llamado Amplificador Operacional de Transconductancia (Figura 3.11).



**FIG.. 3.11 Amplificador Operacional de Transconductancia.**

El propósito de la transformación, es el de modificar los niveles de salida a un valor razonable (entre cero y uno). Esta transformación es hecha antes de alcanzar la salida del próximo nivel. Sin tal transformación, el valor de la salida puede ser muy grande, especialmente cuando en una red varias capas son involucradas. Algunas veces, en lugar de la función de transformación, un valor de umbral es usado. Por ejemplo, cualquier valor de 0.5 o menor es cambiado a cero; cualquier valor arriba e 0.5 es cambiado a uno.

Una transformación puede ocurrir a la salida de cada elemento procesado, o este puede ser ejecutado al final de la salida de la red.

Para que quede más claro el concepto de la transformación o función activadora daremos un pequeño ejemplo:

$$\begin{array}{l}
 X_1 = 3 \\
 X_2 = 1 \\
 X_3 = 2
 \end{array}
 \begin{array}{l}
 \xrightarrow{W_1 = 0.2} \\
 \xrightarrow{W_2 = 0.4} \\
 \xrightarrow{W_3 = 0.1}
 \end{array}
 \text{Procesando el elemento } \underline{f = 1.2}$$

La función sumatoria es:

$$f = 3(0.2) + 1(0.4) + 2(0.1) = 1.2$$

La función de transformación o función activadora es:

$$f_T = \frac{1}{1 + e^{-1.2}} = 0.77$$

A la **salida** simplemente tenemos el elemento que produce la señal, de acuerdo con el elemento anterior, se puede decir que es un simple buffer que transmite una respuesta. Esto se puede implementar con un Amplificador Operacional o con un buffer.

De esta manera con elementos simples podemos elaborar una Neurona Artificial, pero sólo una. Hacemos énfasis en este punto, porque para la elaboración de una red se necesitan cientos y hasta miles de estos elementos simples, además de un software de soporte para el manejo de pesos y otros parámetros.

Como hemos observado realmente la elaboración de una Neurona Artificial básica no es complicada. Pero una sola neurona realmente no puede hacer nada, pues requiere de más neuronas para tener señales de entrada y mandar la señal o señales de salida a otras neuronas. Lo anterior es solamente el principio de un campo tan amplio que requiere de biólogos, sicólogos, ingenieros de computación e ingenieros de electrónica, para el avance en sus investigaciones, y que actualmente tiene la atención de grandes científicos que creen y quieren lograr una máquina con la semejanza del cerebro humano.

# **CAPITULO 4**

---

## **DESARROLLO DEL SISTEMA EXPERTO TUTORIAL DE REDES NEURONALES ARTIFICIALES**

## **4. DESARROLLO DE SETURNA.**

### **4.1 DESARROLLO Y CARACTERISTICAS DE SETURNA.**

Todos sabemos la gran utilidad que las computadoras nos brindan hoy en día, y además que en cualquier industria, sector administrativo y todas las instituciones de nivel superior es indispensable el uso de éstas máquinas.

La aplicación de los Sistemas Expertos Tutoriales (SET), se basa principalmente en el mejor aprovechamiento de la información que los usuarios obtengan al hacer uso de estos.

Es por eso que es importante señalar que en la mayoría de los casos, hacemos uso de los Sistemas Expertos Tutoriales cuando no existen los suficientes Expertos de la materia en estudio, o si los hay están a un precio elevado, o no están disponibles; o bien cuando se requiere capacitar personal, difundir información sobre alguna investigación o simplemente hacer uso de un SET para aprender cualquier tema en casa.

#### **4.1.1 ALCANCE DE LA APLICACION.**

Hemos desarrollado un Sistema Experto Tutorial llamado SETURNA, el cuál está enfocado para ser empleado en la enseñanza de Redes Neuronales, en diferentes Instituciones Universitarias, que brinden este tipo de capacitación. Algunas de las Instituciones en donde se le puede dar aplicación inmediata a SETURNA son:

- UNAM Facultad de Ingeniería en la materia de Inteligencia Artificial que corresponde a la carrera de Ingeniería en Computación.
- También existe la posibilidad de ser empleado en el Posgrado de la Facultad de Ingeniería como material de apoyo, para la maestría con el mismo tema.

#### **4.1.2 ADQUISICION DEL CONOCIMIENTO.**

Para la obtención de información del conocimiento sobre Redes Neuronales Artificiales en el campo de Reconocimiento de Patrones se contó principalmente con la colaboración del Dr. Felipe Lara Rosano experto en la materia, además de Información obtenida de la siguientes Bibliotecas:

- Biblioteca de Posgrado de la Facultad de Ingeniería.
- Biblioteca Central de la Universidad Nacional Autónoma de México.
- Biblioteca Antonio Dovalí del edificio principal de la Facultad de Ingeniería.

#### 4.1.3 HERRAMIENTAS DE DESARROLLO.

La principal herramienta para el desarrollo de SETURNA, es un Shell orientado a objetos, que contiene diversas utilidades para la elaboración de botones, pantallas, mensajes, y otras aplicaciones, que nos dan la facilidad para crear una interface con el usuario, de ambiente agradable y fácil de manejar. Este Shell es LEVEL 5 OBJECT.

Además LEVEL 5 OBJECT resultó ser un Shell adecuado para la realización del sistema por tener las siguientes características:

1. *Es un Sistema Híbrido.* Es decir, que puede manejar tanto objetos, reglas y funciones. De este modo el conocimiento puede ser conducido de manera muy similar a como es manejado por un ser humano.
2. *Interfaz gráfica altamente desarrollada.* En este caso nos referimos al escaso uso del teclado, pues, la mayoría de la programación se puede realizar por medio del mouse.
3. *Utilización de Hipertextos e Hiperregiones.* Tiene la capacidad de manejar la información de manera no secuencial y de acuerdo al nivel de conocimiento, y los requerimientos del usuario, por lo que es posible realizar Hipertextos e Hiperregiones, que dan al ambiente de interface una mejor y más clara comunicación con el usuario.
4. *Interacción con programas externos.* LEVEL 5 OBJECT tiene también, la capacidad de interactuar con programas externos como son Bases de Datos, Hojas de Cálculo y Programas Ejecutables.

#### 4.1.4 ARQUITECTURA DE SETURNA.

El Sistema Experto SETURNA está comprendido de 3 módulos:

- El Módulo Uno comprende la Adquisición de datos de usuario
- El Módulo Dos contienen toda la información sobre Redes Neuronales Artificiales, y el índice temático.
- El Módulo Tres consta de la evaluación del usuario

1. *Adquisición de datos de usuario.* - En este módulo, SETURNA pide los datos necesarios al usuario; datos que servirán al Sistema para poder llevar un control del manejo de información por medio de una Base de Datos, en donde tendrá registrados a los usuarios, en todo el proceso del Sistema, y de esta manera llevar un registro de cada uno de ellos.

Los datos que se le piden al usuario son : Nombre , Apellido Paterno, Apellido Materno y una Clave de Acceso, está última puede constar de letras y números, pero no más de 6 caracteres, que posteriormente le servirá al usuario para entrar al Sistema tantas veces sea necesario y continuar con la sesión, además de que al ingresar nuevamente al Tutorial el usuario entrará en la sección que vio la última vez.

**2. Contenido Temático.-** Aquí encontramos toda la información sobre RNA que consta de 6 temas. Dentro de cada tema podemos encontrar hipertextos, que tiene la función de darnos el contenido de cada palabra o posible subtema, que esté indicado como tal; así también encontramos hiperregiones, las cuales nos mostrarán figuras o esquemas, que de alguna manera harán más completa la explicación del tema en que nos encontremos.

Los Temas de SETURNA son los siguientes:

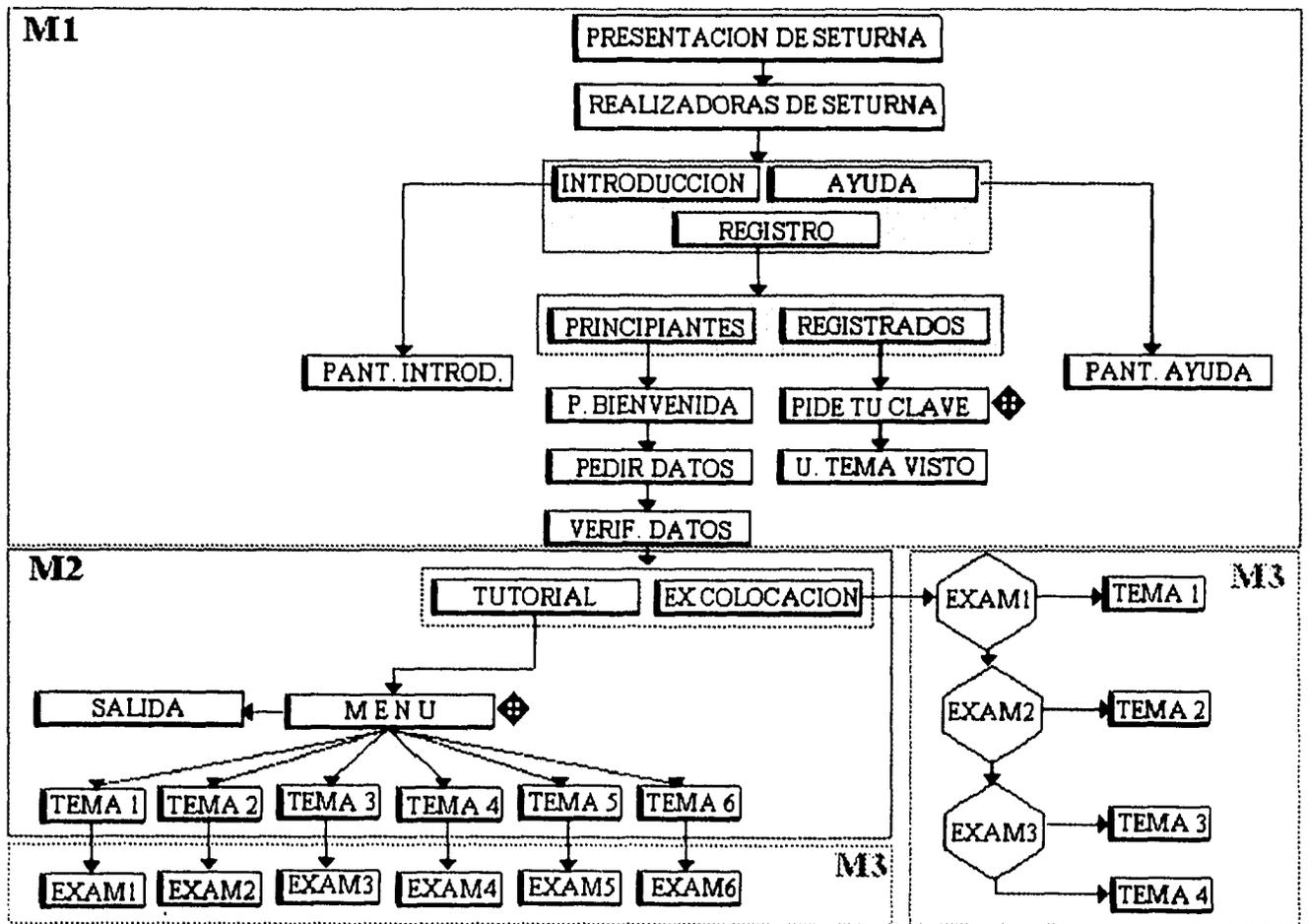
- Generalidades sobre Redes Neuronales
- Redes Neuronales Artificiales Fundamentales
- Modelos de Aprendizaje
- Aplicaciones en el Reconocimiento de Patrones

**3. Evaluación.-** En éste módulo se realiza una evaluación al usuario; de esta manera se dará cuenta del avance que tiene en los conocimientos adquiridos. Para este fin se cuenta con dos tipos de exámenes:

**a) Examen de colocación.-** Este examen se aplica con el fin de que sea utilizado por aquellos usuarios que tienen conocimientos sobre redes neuronales artificiales, y es aplicado al inicio del tutorial; de esta manera el usuario iniciará el tutorial dentro del tema que no domina. La finalidad de la aplicación de éste examen es hacer el tutorial activo, para que el usuario tenga un mejor aprovechamiento.

**b) Examen de evaluación .-** Este examen es aplicado al final de cada tema visto, y el usuario no podrá realizar este examen sin antes haber pasado por todos subtemas del tema, ya que se le harán preguntas de toda la información.

A continuación mostramos a ustedes un diagrama de bloques que nos muestran la Arquitectura del Sistema Experto Tutorial SETURNA:



ARQUITECTURA DE SETURNA.

#### **4.1.5 PROGRAMAS DE ENLACE.**

En cada uno de los módulos descritos, se contempló que SETURNA accesa a Bases de Datos diseñadas en DBASE III+, a fin de recuperar información de cada uno de los usuarios cuando sea necesario.

#### **4.1.6 COMUNICACION CON EL USUARIO.**

El sistema experto SETURNA pide información al usuario a través de pantallas de captura, las cuáles pueden ser grupos de botones, pantallas de despliegue o pantallas gráficas. Para introducir esta información se puede usar el teclado y/o ratón. Para brindar información, el sistema emplea pantallas de despliegue y pantallas gráficas.

#### **4.1.7 REQUERIMIENTOS**

Los requerimientos para el desarrollo del Sistema SETURNA son los siguientes:

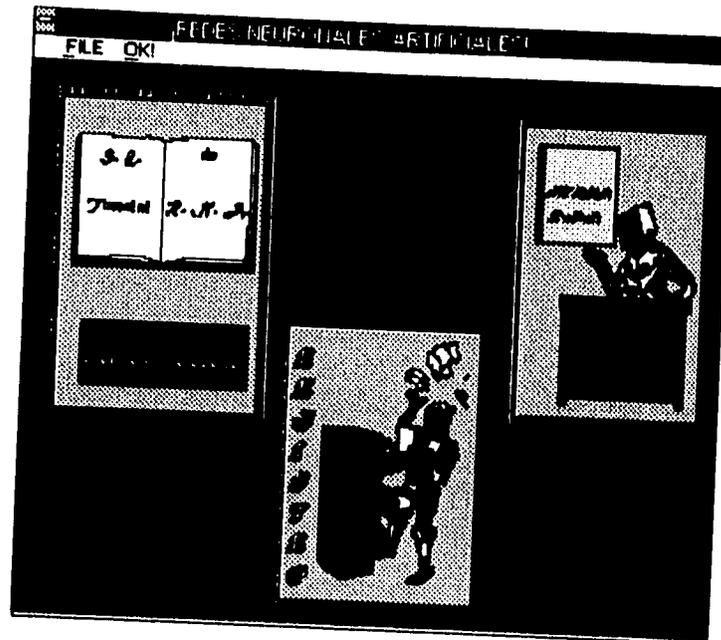
- Una Computadora PC 80386 o modelo más reciente.
- Un Disco duro de 40 Mbytes.
- 4 mbytes en RAM mínimo.
- Windows 3.0 o versiones más recientes.
- LEVEL 5 OBJECT 2.5.
- Mouse.
- Espacio en disco duro para el sistema de 10 Mb mínimo.

#### **4.1.8 OPERACION DEL SETURNA**

Como ya se mencionó el Sistema SETURNA esta estructurado por tres módulos:

1. Módulo Uno: Adquisición de Datos de Usuario.
2. Módulo Dos: Contenido Temático.
3. Módulo Tres: Evaluación.

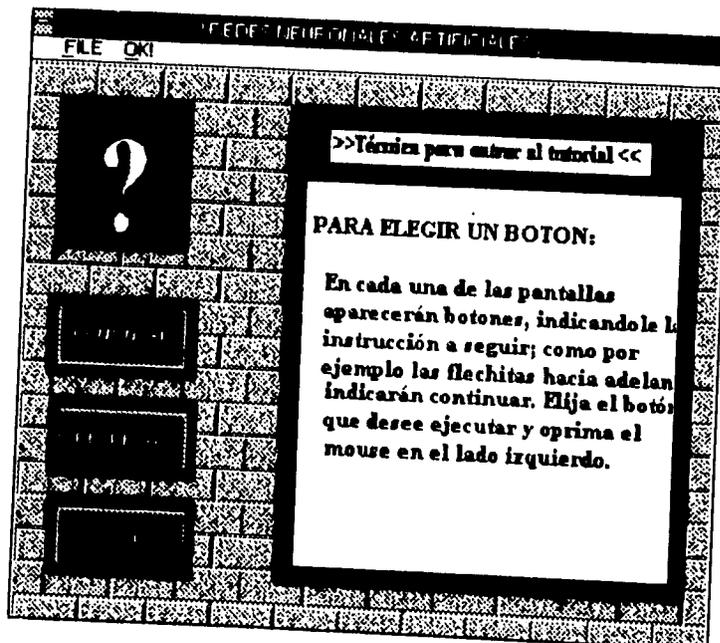
**1. Modulo de Adquisición de Datos de Usuario.** Dentro de éste modulo se encuentra un botón de AYUDA sobre el manejo del Tutorial, en donde usted encontrará información acerca de como resolver los exámenes y sobre la navegación en los diferentes Temas con los que éste cuenta, así también se encuentra un botón de INTRODUCCION al Sistema y un botón de REGISTRO el cuál conduce al usuario, a la parte de adquisición de datos. A continuación se muestra la pantalla correspondiente a lo que se mencionó:



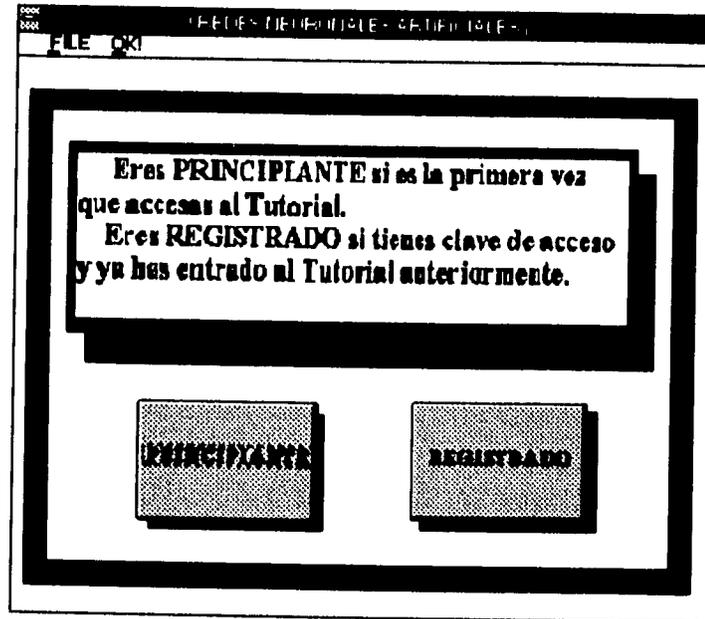
**Introducción.** Presentación de la introducción del sistema.

**Ayuda.** Es la ayuda del sistema, que nos menciona los procedimientos a seguir, para que el usuario no tenga ningún problema en el uso del Tutorial

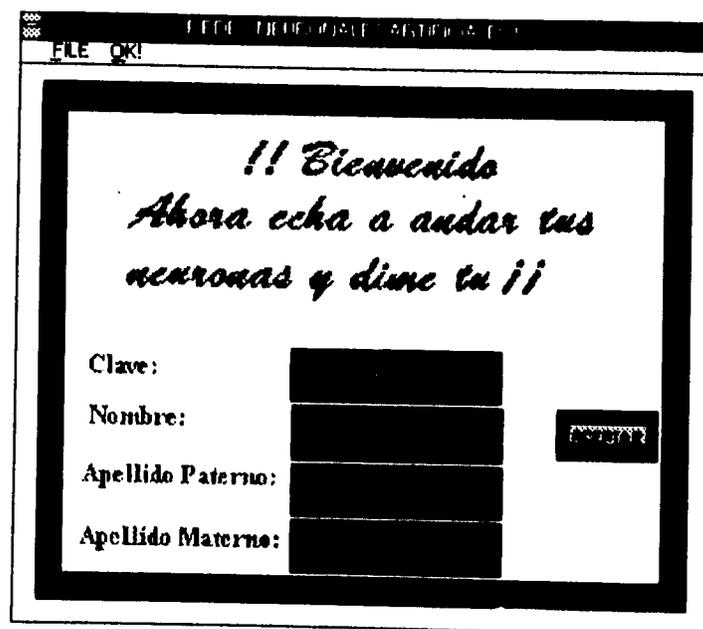
Enseguida se muestra la pantalla de Ayuda:



**Registro.** Al seleccionar este botón se nos presenta la siguiente pantalla que contiene 2 opciones:



**PRINCIPIANTES.** De la pantalla anterior, se seleccionará éste botón cuando es la primera vez que el usuario utiliza el Tutorial; y al oprimir este botón tendremos la siguiente pantalla.



Aquí, es en donde el usuario tiene la opción de introducir sus datos personales así como una Clave de acceso que él usuario desee, tomando en cuenta que dicha Clave no podrá tener más de 6 caracteres que puede incluir letras y números.

Si el usuario llegará a equivocarse al introducir sus datos o Clave, se cuenta con la siguiente pantalla en donde el usuario tiene la opción de corregir sus datos o cambiar la clave.

A screenshot of a graphical user interface window titled "REDES NEURONALES ARTIFICIALES". The window has a menu bar with "FILE" and "OKI". The main content area contains the following text and input fields:

- CLAVE: [input field]
- NOMBRE: [input field]
- APELLIDO PATERNO: [input field]
- APELLIDO MATERNO: [input field]

Below the input fields, there is a message: "Verifica tus datos si son correctos opríme ACEPTAR si no selecciona CORREGIR." To the right of this message is a button labeled "Aceptar".

**2. Módulo de Adquisición de Datos.** Una vez que el usuario a introducido sus datos, ya puede iniciar con el Tutorial, para esto tiene dos opciones que se presentan en la siguiente pantalla:

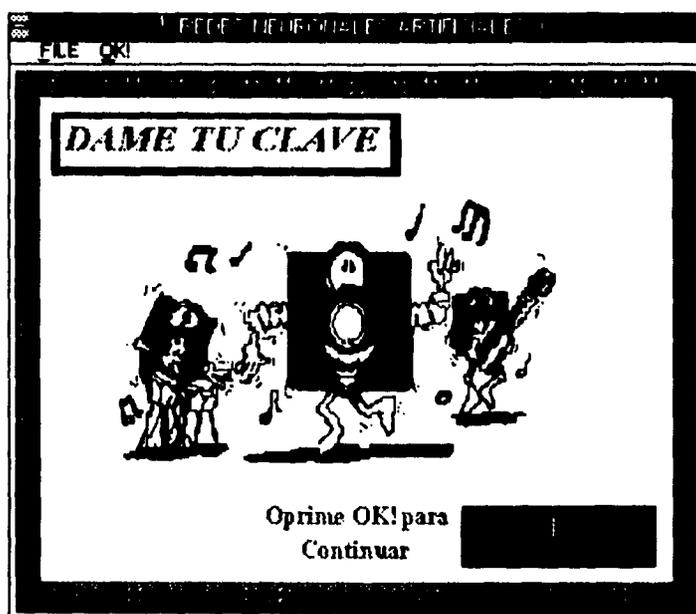
A screenshot of a graphical user interface window titled "REDES NEURONALES ARTIFICIALES". The window has a menu bar with "FILE" and "OKI". The main content area contains the following text and buttons:

- Para aprender Redes Neuronales Artificiales presiona TUTORIAL. [TUTORIAL button]
- Si tienes conocimientos de RNA te recomendamos realizar el Examen de Colocacion. [EXAMEN DE COLOCACION button]

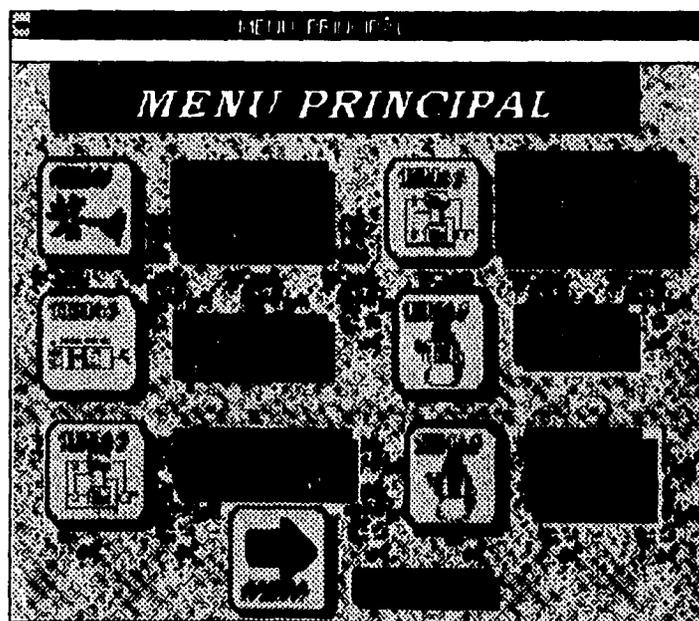
**Tutorial.** Al seleccionar este botón nos manda a la pantalla del Menú principal, en donde iniciamos a partir del TEMA 1 de SETURNA.

**Examen Colocación.** Se selecciona este botón, si el usuario tiene conocimientos sobre Redes Neuronales Artificiales, ya que se inicia el Tutorial con un examen; esto con el fin de evaluar al usuario, y saber en que tema puede colocarse.

**REGISTRADOS.** Si el usuario selecciona este botón, es porque ya tuvo una sesión previa, en donde introdujo sus datos, aparecerá entonces, una pantalla como la siguiente en donde se le pedirá la clave de acceso que introdujo con anterioridad.



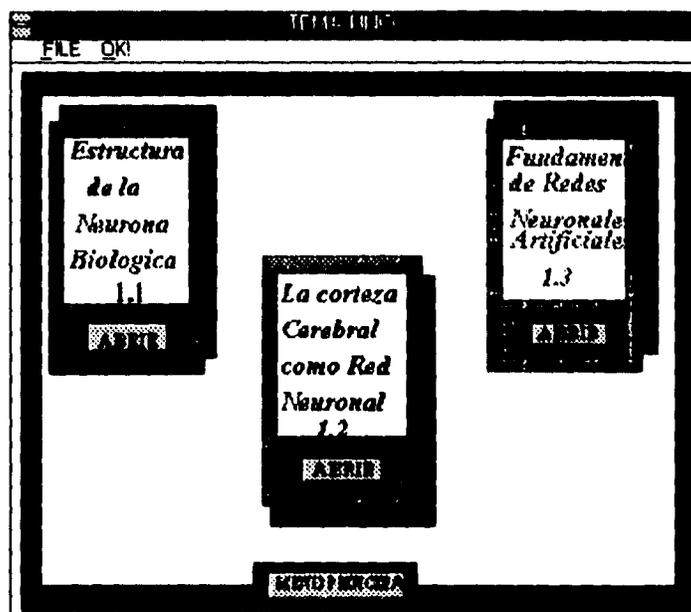
Una vez que el usuario esté dentro del Tutorial tendrá un Menú Principal



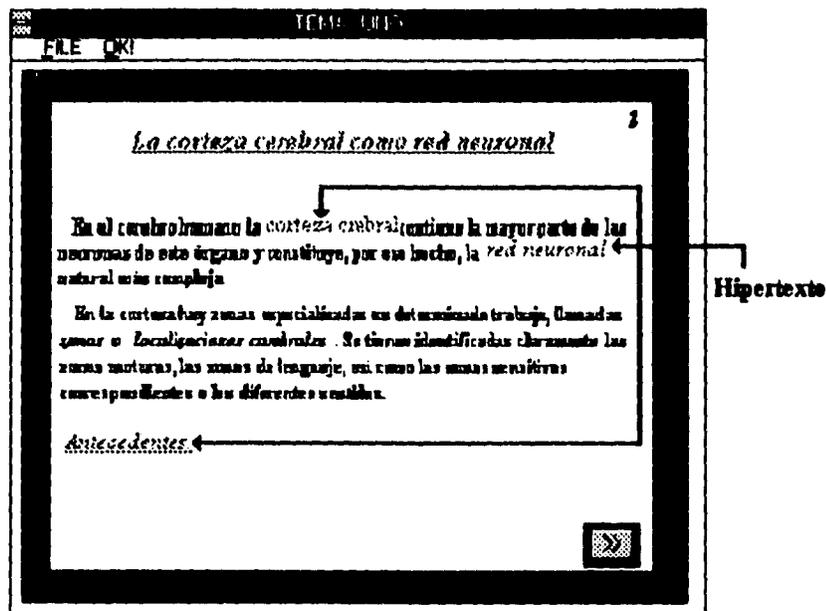
El Menú Principal consta de 7 botones, en 6 de ellos entramos a cada uno de los Temas del Tutorial y el séptimo botón nos conduce a la salida del Sistema.

Al entrar por primera vez al Sistema, notaremos que en el menú principal sólo podremos acceder al tema uno. Para continuar con los siguientes temas se presenta un examen al aprobar el examen de cada tema se tiene acceso al siguiente

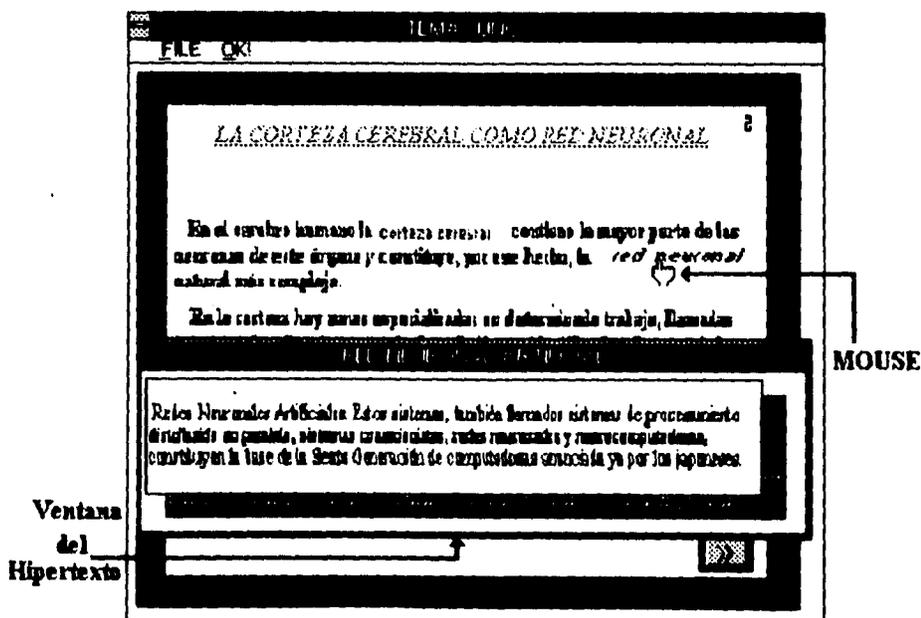
Dentro de los Temas se tiene una pantalla como la que se muestra a continuación, en donde se presentan los subtemas del tema que se esta viendo. Para poder entrar a cada uno de ello es suficiente con presionar el botón que se indica:



Al entrar a un subtema se nos puede presentar una pantalla como la siguiente, en donde tenemos textos marcados en verde y letras cursivas, que nos indican que existe un hipertexto.

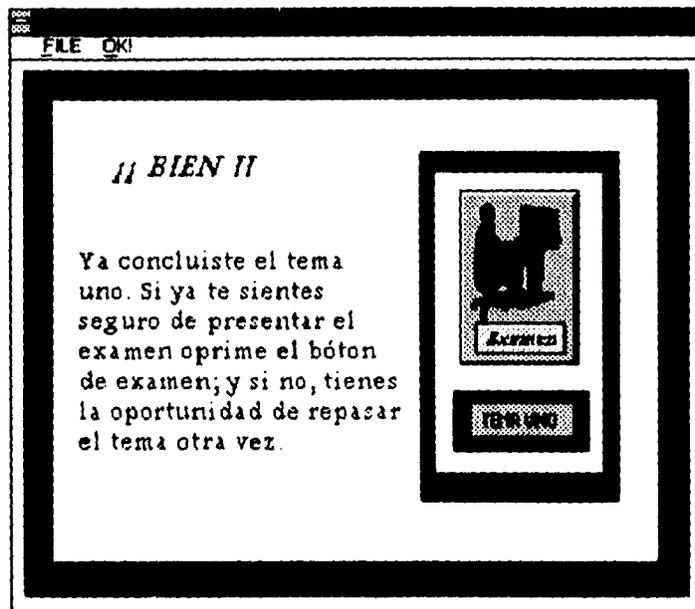


Quando sealamos un hipertexto el mouse toma la forma de una mano, al presionar nos aparece una ventana que pertenece al hipertexto, para quitar dicha ventana hay que hacer un clic con el mouse en cualquier parte fuera de la ventana del hipertexto e inmediatamente desaparecerá la ventana:



Dentro de los hipertextos se pueden tener textos o figuras refiriéndose al tema en el que nos encontramos.

**3. Módulo de Evaluación.** Al final de cada tema se presenta una pantalla que nos envía al examen, y que corresponde al Tema que se esta viendo, como se muestra a continuación:

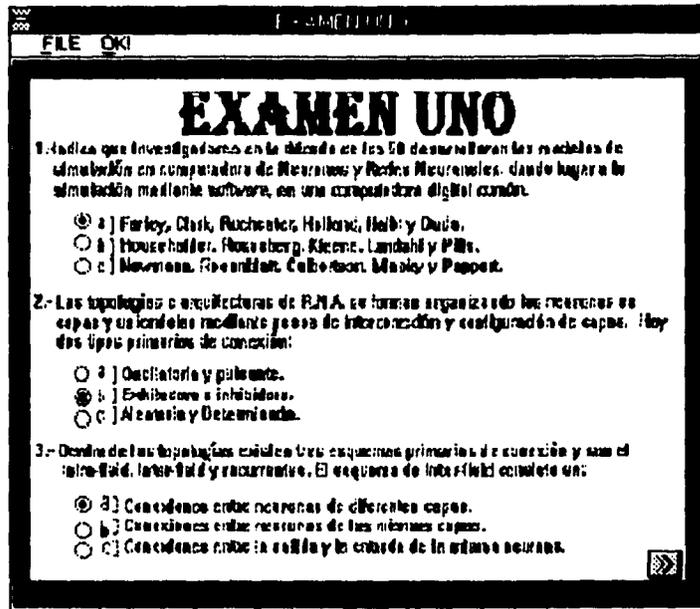


Esta pantalla envía al usuario al examen, que cuenta con 3 diferentes tipos para cada Tema, pues el usuario, sólo cuenta con 3 oportunidades para poder aprobar dicho Tema, en caso de no aprobar ninguno de los tres exámenes el usuario será borrado de la Base de Datos del Tutorial. Para poder ingresar nuevamente el usuario tendrá que hacerlo como principiante.

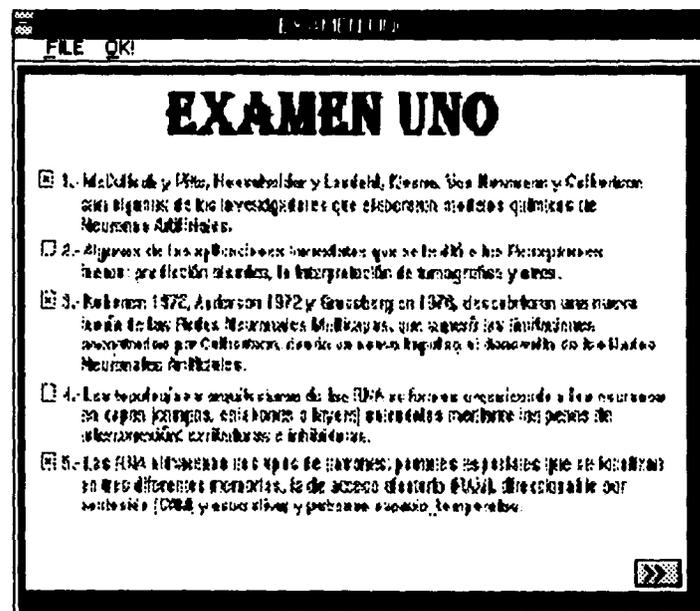
Dentro de los exámenes se tienen 2 formatos diferentes:

- a) El primero es por opción múltiple, en este formato el usuario tendrá que seleccionar una de las diferentes opciones que se le presentan. La selección de cada pregunta se hará colocando el mouse en la respuesta y darle un clic.

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA



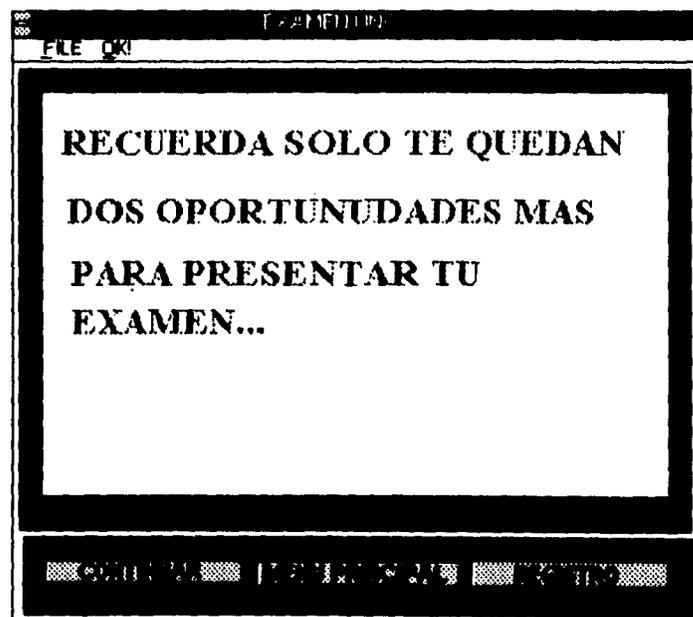
b) La segunda opción, es marcar falso o verdadero. Aquí el usuario marca con una cruz con el mouse aquellas respuestas que son verdaderas y dejara en blanco las preguntas que sean falsas; enseguida se muestra un ejemplo:



Una vez que se ha resuelto el examen al estudiante se le presenta la siguiente pantalla, en donde se le muestran sus calificaciones, y un mensaje que el Tutorial le manda:



Una vez que al usuario se le ha presentado su calificación, se le manda un mensaje, en donde le indican las oportunidades que le faltan para aprobar su examen, o bien el Tema con el que continua, en caso de haber aprobado.



La pantalla anterior tiene tres opciones

- 1) *Continuar*: en este caso puede continuar con el siguiente Tema o bien volver a iniciar el Tema si lo desea.
- 2) *Menú Principal*: aquí tenemos la opción de salir del sistema o pasar al siguiente tema.
- 3) *Registro*: este botón nos enviará a una pantalla, en donde se presentan los datos del usuario, sus calificaciones obtenidas hasta el momento y su promedio como se muestra a continuación:

The screenshot shows a window titled 'EXAMENES' with a menu bar containing 'FILE' and 'OK'. The main content area displays the following information:

NOMBRE:	
JORGE ALBERTO VARGAS NERI	
EXAMEN 1:	8
EXAMEN 2:	9
EXAMEN 3:	7
ULTIMO TEMA:	3
PROMEDIO:	8

The scores for EXAMEN 1, EXAMEN 2, EXAMEN 3, and PROMEDIO are each shown with a horizontal line underneath. There is a small 'OK' button in the bottom right corner of the window.

# CAPITULO 5

---

---

CONCLUSIONES.

## CONCLUSIONES.

La educación por computadora e informática es uno de los grandes retos a los que se enfrenta la sociedad de hoy, el lograr la formación e información de cómputo e informática para toda la comunidad, es una guía indispensable para entender a la sociedad actual. Su estructura, interés y todas las relaciones se dan en un ámbito cada vez más tecnificado que incorporan innovaciones tecnológicas a un ritmo vertiginoso.

La educación actual se planea consciente e inconscientemente para la preparación del hombre, hacia una sociedad futura; una sociedad en la que la electrónica la computación y la informática juegan un papel relevante.

Con la realización del Sistema se pudieron reafirmar los conocimientos de la programación orientada a objetos, al resolver los problemas que se fueron presentando a lo largo de la realización de SETURNA.

SETURNA sirve como modelo para la realización de otros Tutoriales similares que varíen en la información, ya que la programación no está directamente relacionada con el contenido del Tutorial.

El uso del Sistema da la posibilidad de hacer una evaluación del desempeño del usuario, de acuerdo a la metodología propuesta, sistematizada y periódica que permite al profesor mejorar sus actividades y enfocar su papel en asuntos específicos en el tema. Las evaluaciones que contiene, son en forma general la aplicación de exámenes que facilitan el manejo de información del usuario.

No se puede pensar que el software educativo sustituya al maestro, tampoco podemos pensar que representa una solución a todos los problemas de la enseñanza de las ciencias, especialmente de las ciencias básicas; ni mucho menos creer que sustituye una forma rápida de aprender un tema en particular sin el menor esfuerzo.

Consideramos que los puntos más sobresalientes del Sistema son los siguientes:

° La importancia de un Sistema Experto Tutorial (SET) en la educación actual es básico debido al gran avance tecnológico que se tiene día con día. Considerando la individualidad, como un factor importante en el desarrollo del alumno, además de la facilidad de manejo y distribución del SET.

° Para el desarrollo de SETURNA fue esencial manejar diferentes disciplinas, que son: Psicología, Diseño Gráfico e Ingeniería.

## CONCLUSIONES

---

- ° Lo principal para el desarrollo de un Sistema Experto es manejar los conceptos básicos de Inteligencia Artificial y Computación, así como tener los conocimientos del Experto Humano.
- ° Consideramos que el contenido de nuestro trabajo sobre Redes Neuronales Artificiales es sólo una pequeña parte, de una amplia rama en donde se involucran diferentes ciencias como son: biología, psicología, electrónica y computación.
- ° Actualmente el aprendizaje en Redes Neuronales Artificiales es através de software, aunque ya se están desarrollando circuitos electrónicos para este fin.
- ° Como ejemplo de aplicación de las Redes Neuronales Artificiales utilizamos BrainMaker, el cual realiza la simulación de una RNA para el Reconocimiento de Patrones (Numéricos, Simbólicos o Gráficos) por medio de software.

# APENDICE A

---

---

## APENDICE A

Listado de las reglas de SETURNA .

### TUTORIAL.KNB

#### Avanza

WHEN CHANGED

BEGIN

te OF variable := TRUE

END

#### Ayuda1 OF variable

WHEN CHANGED

BEGIN

ASK ayuda

BEGIN

texto OF variable := text OF tx1

END

END

#### Dato OF variable

WHEN CHANGED

BEGIN

title OF main window := " UN MOMENTO POR FAVOR "

CHAIN "DATOS"

END

#### Intro OF variable

WHEN CHANGED

BEGIN

ASK introducción

END

#### Pag

WHEN CHANGED

BEGIN

te OF variable := TRUE

END

**Re OF variable**

WHEN CHANGED

BEGIN

pagina OF variable := pagina OF variable - 1

BEGIN

IF pagina OF variable = 1 THEN

texto OF variable := text OF tx1

IF pagina OF variable = 2 THEN

texto OF variable := text OF tx2

IF pagina OF variable = 3 THEN

texto OF variable := text OF tx3

IF pagina OF variable = 4 THEN

texto OF variable := text OF tx4

IF pagina OF variable = 5 THEN

texto OF variable := text OF tx5

IF pagina OF variable = 6 THEN

texto OF variable := text OF tx6

IF pagina OF variable = 7 THEN

texto OF variable := text OF tx7

IF pagina OF variable = 8 THEN

texto OF variable := text OF tx8

IF pagina OF variable = 9 THEN

texto OF variable := text OF tx9

IF pagina OF variable = 10 THEN

texto OF variable := text OF tx10

END

END

**Regraso OF variable**

WHEN CHANGED

BEGIN

re OF variable := TRUE

END

**Te OF variable**

WHEN CHANGED

BEGIN

pagina OF variable := pagina OF variable + 1

```
BEGIN
  IF pagina OF variable = 2 THEN
    texto OF variable := text OF tx2
  IF pagina OF variable = 3 THEN
    texto OF variable := text OF tx3
  IF pagina OF variable = 4 THEN
    texto OF variable := text OF tx4
  IF pagina OF variable = 5 THEN
    texto OF variable := text OF tx5
  IF pagina OF variable = 6 THEN
    texto OF variable := text OF tx6
  IF pagina OF variable = 7 THEN
    texto OF variable := text OF tx7
  IF pagina OF variable = 8 THEN
    texto OF variable := text OF tx8
  IF pagina OF variable = 9 THEN
    texto OF variable := text OF tx9
  IF pagina OF variable = 10 THEN
    texto OF variable := text OF tx10
  END
END
```

## DATOS.KNB

### Buscame OF procedimientos

```
WHEN CHANGED
BEGIN
  action OF dB3_GIRLS IS open := TRUE
  FIND dB3_GIRLS
  WHERE klaveprincipal = klave OF dB3_GIRLS
  WHEN FOUND

  ASK avisol
  visible OK button OF main window := TRUE
  ASK datos
  FIND END
  action OF dB3_GIRLS IS pack := TRUE
END
```

**Grabadat OF procedimientos****WHEN CHANGED****BEGIN**

action OF dB3\_GIRLS IS open := TRUE

**FIND dB3\_GIRLS**

WHERE klaveprincipal = klave OF dB3\_GIRLS

**WHEN FOUND**

ASK aviso1

visible OK button OF main window := TRUE

ASK datos

**WHEN NONE FOUND**

action OF dB3\_GIRLS IS append record := TRUE

klave OF dB3\_GIRLS := klaveprincipal

nombre OF dB3\_GIRLS := name

paterno OF dB3\_GIRLS := appaterno1

materno OF dB3\_GIRLS := apmaterno2

kalif1 OF dB3\_GIRLS := kalificacion1

kalif2 OF dB3\_GIRLS := kalificacion2

kalif3 OF dB3\_GIRLS := kalificacion3

kalif4 OF dB3\_GIRLS := kalificacion4

kalif5 OF dB3\_GIRLS := kalificacion5

kalif6 OF dB3\_GIRLS := kalificacion6

kaltot OF dB3\_GIRLS := total

ultima OF dB3\_GIRLS := ultem

ultsubtema OF dB3\_GIRLS := ultsubtem

variable OF dB3\_GIRLS := record OF dB3\_GIRLS + 2

action OF dB3\_GIRLS IS pack := TRUE

action OF dB3\_GIRLS IS close := TRUE

**FIND END**

ASK opciones1

**END****Key****WHEN CHANGED****BEGIN**

action OF GIRLS 1 IS open := TRUE

**FIND dB3\_GIRLS**

WHERE key = klave OF dB3\_GIRLS

**WHEN FOUND**

access OF dB3\_GIRLS IS write := TRUE

variable OF dB3\_GIRLS := record OF dB3\_GIRLS + 1

klaveprincipal := klave OF dB3\_GIRLS

```

name := nombre OF dB3_GIRLS
apaterno1 := paterno OF dB3_GIRLS
apaterno2 := materno OF dB3_GIRLS
kalificacion1 := kalif1 OF dB3_GIRLS
kalificacion2 := kalif2 OF dB3_GIRLS
kalificación3 := kalif3 OF dB3_GIRLS
kalificacion4 := kalif4 OF dB3_GIRLS
kalificacion5 := kalif5 OF dB3_GIRLS
kalificacion6 := kalif6 OF dB3_GIRLS
total := kaltot OF dB3_GIRLS
ultem := ultima OF dB3_GIRLS
ultsubtem := ultsubtema OF dB3_GIRLS
manda OF variables := TRUE
action OF GIRLS 1 IS close := TRUE
WHEN NONE FOUND
action OF GIRLS 1 IS close := TRUE
shances OF variables := shances OF variables - 1
mensa := TRUE
FIND END
END

```

### **Mensajes**

**WHEN CHANGED**

**BEGIN**

**IF shances OF variables > 0 THEN**

**BEGIN**

visible OF mensaje2 := TRUE

output OF mensaje2 := mensaje1

**FOR (num := 1 TO 5) END**

**ELSE**

**FOR (num := 1 TO 5) exit OF application := TRUE**

**END**

### **Buscame Of procedimientos**

**WHEN CHANGED**

**BEGIN**

action OF dB3\_GIRLS IS open := TRUE

**FIND dB3\_GIRLS**

**WHERE klaveprincipal = klave OF dB3\_GIRLS**

```

WHEN FOUND
  ASK aviso1
  visible OK button OF main window := TRUE
  ASK datos
  FIND END
  action OF dB3_GIRLS IS pack := TRUE
END

```

### Grabadat OF procedimiento

```

WHEN CHANGED
  BEGIN
  action OF dB3_GIRLS IS open := TRUE
  FIND dB3_GIRLS
  WHERE klaveprincipal = klave OF dB3_GIRLS
  WHEN FOUND
    ASK aviso1
    visible OK button OF main window := TRUE
    ASK datos
  WHEN NONE FOUND
    action OF dB3_GIRLS IS append record := TRUE
    klave OF dB3_GIRLS := klaveprincipal
    nombre OF dB3_GIRLS := name
    paterno OF dB3_GIRLS := appaterno1
    materno OF dB3_GIRLS := apmaterno2
    kalif1 OF dB3_GIRLS := kalificacion1
    kalif2 OF dB3_GIRLS := kalificacion2
    kalif3 OF dB3_GIRLS := kalificacion3
    kalif4 OF dB3_GIRLS := kalificacion4
    kalif5 OF dB3_GIRLS := kalificacion5
    kalif6 OF dB3_GIRLS := kalificacion6
    kaltot OF dB3_GIRLS := total
    ultema OF dB3_GIRLS := ultem
    ultsubtema OF dB3_GIRLS := ultsubtem
    variable OF dB3_GIRLS := record OF dB3_GIRLS + 2
    action OF dB3_GIRLS IS pack := TRUE
    action OF dB3_GIRLS IS close := TRUE
  FIND END
  ASK opciones1
END

```

**MENU5.KNB**

**Buscar**

**WHEN CHANGED**

**BEGIN**

action OF GIRLS 1 IS open := TRUE

**FIND dB3 GIRLS 1**

**WHERE** variable OF dB3 GIRLS 1  $\diamond$  0

**WHEN FOUND**

klaveprincipal := klave OF dB3 GIRLS 1

name := nombre OF dB3 GIRLS 1

apaterno1 := paterno OF dB3 GIRLS 1

apmaterno2 := materno OF dB3 GIRLS 1

kalificacion1 := kalif1 OF dB3 GIRLS 1

kalificacion2 := kalif2 OF dB3 GIRLS 1

kalificación3 := kalif3 OF dB3 GIRLS 1

kalificacion4 := kalif4 OF dB3 GIRLS 1

kalificacion5 := kalif5 OF dB3 GIRLS 1

kalificacion6 := kalif6 OF dB3 GIRLS 1

total := kaltot OF dB3 GIRLS 1

ultem := ultema OF dB3 GIRLS 1

ulsubtem := ulsubtema OF dB3 GIRLS 1

manda OF variables := TRUE

action OF GIRLS 1 IS close := TRUE

**FIND END**

**IF** kalificacion1  $\geq$  7 **THEN**

**BEGIN**

enabled OF bt2 := TRUE

**IF** kalificación2  $\geq$  7 **THEN**

**BEGIN**

enabled OF bt3 := TRUE

**IF** kalificación3  $\geq$  7 **THEN**

**BEGIN**

enabled OF bt4 := TRUE

**IF** kalificacion4  $\geq$  7 **THEN**

**BEGIN**

enabled OF bt5 := TRUE

**IF** kalificacion5  $\geq$  7 **THEN**

**BEGIN**

enabled OF bt6 := TRUE

**END**

**END**

**END**

**Salir**

**WHEN CHANGED**

**BEGIN**

action OF dB3 GIRLS 1 IS open := TRUE

**FIND dB3 GIRLS 1**

**WHERE** variable OF dB3 GIRLS 1  $\diamond$  0

**WHEN FOUND**

action OF dB3 GIRLS 1 IS recall record := TRUE

access OF dB3 GIRLS 1 IS write := TRUE

variable OF dB3 GIRLS 1 := 0

action OF dB3 GIRLS 1 IS pack := TRUE

action OF dB3 GIRLS 1 IS close := TRUE

**FIND END**

exit OF application := TRUE

**END**

**Tema1 OF temas**

**WHEN CHANGED**

**BEGIN**

ultem := 1

action OF dB3 GIRLS 1 IS open := TRUE

**FIND dB3 GIRLS 1**

**WHERE** variable OF dB3 GIRLS 1  $\diamond$  0

**WHEN FOUND**

action OF dB3 GIRLS 1 IS recall record := TRUE

record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1

access OF dB3 GIRLS 1 IS write := TRUE

ultima OF dB3 GIRLS 1 := ultem

action OF dB3 GIRLS 1 IS pack := TRUE

action OF dB3 GIRLS 1 IS close := TRUE

**FIND END**

title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEMA UNO"

**CHAIN "tema1"**

**END**

**Tema2 OF temas**

**WHEN CHANGED**

**BEGIN**

ultem := 2

action OF dB3 GIRLS 1 IS open := TRUE

**FIND dB3 GIRLS 1**

**WHERE** variable OF dB3 GIRLS 1  $\diamond$  0

**WHEN FOUND**

action OF dB3 GIRLS 1 IS recall record := TRUE  
 record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1  
 access OF dB3 GIRLS 1 IS write := TRUE  
 ultema OF dB3 GIRLS 1 := ultem  
 action OF dB3 GIRLS 1 IS pack := TRUE  
 action OF dB3 GIRLS 1 IS close := TRUE

**FIND END**

title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEMA  
 DOS"  
 CHAIN "tema2"  
 END

**Tema3 OF temas**

**WHEN CHANGED**

**BEGIN**

ultem := 3  
 action OF dB3 GIRLS 1 IS open := TRUE

**FIND dB3 GIRLS 1**

**WHERE** variable OF dB3 GIRLS 1 <> 0

**WHEN FOUND**

action OF dB3 GIRLS 1 IS recall record := TRUE  
 record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1  
 access OF dB3 GIRLS 1 IS write := TRUE  
 ultema OF dB3 GIRLS 1 := ultem  
 action OF dB3 GIRLS 1 IS pack := TRUE  
 action OF dB3 GIRLS 1 IS close := TRUE

**FIND END**

title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEMA  
 TRES"  
 CHAIN "tema3"  
 END

**Tema4 OF temas**

**WHEN CHANGED**

**BEGIN**

ultem := 4  
 action OF dB3 GIRLS 1 IS open := TRUE

**FIND dB3 GIRLS 1**

**WHERE** variable OF dB3 GIRLS 1 <> 0

**WHEN FOUND**

action OF dB3 GIRLS 1 IS recall record := TRUE  
 record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1

```

access OF dB3 GIRLS 1 IS write := TRUE
ultima OF dB3 GIRLS 1 := ultem
action OF dB3 GIRLS 1 IS pack := TRUE
action OF dB3 GIRLS 1 IS close := TRUE
FIND END
title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEMA
CUATRO"
CHAIN "tema4"
END

```

**Tema5 OF temas**

WHEN CHANGED

BEGIN

ultem := 5

action OF dB3 GIRLS 1 IS open := TRUE

FIND dB3 GIRLS 1

WHERE variable OF dB3 GIRLS 1 <> 0

WHEN FOUND

action OF dB3 GIRLS 1 IS recall record := TRUE

record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1

access OF dB3 GIRLS 1 IS write := TRUE

ultima OF dB3 GIRLS 1 := ultem

action OF dB3 GIRLS 1 IS pack := TRUE

action OF dB3 GIRLS 1 IS close := TRUE

FIND END

title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEMA  
CINCO"

CHAIN "tema5"

END

**Tema6 OF temas**

WHEN CHANGED

BEGIN

ultem := 6

action OF dB3 GIRLS 1 IS open := TRUE

FIND dB3 GIRLS 1

WHERE variable OF dB3 GIRLS 1 <> 0

WHEN FOUND

action OF dB3 GIRLS 1 IS recall record := TRUE

record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1

access OF dB3 GIRLS 1 IS write := TRUE

ultima OF dB3 GIRLS 1 := ultem

action OF dB3 GIRLS 1 IS pack := TRUE

```

    action OF dB3 GIRLS 1 IS close := TRUE
    FIND END
    title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEMA
    SEIS"
    CHAIN "tema6"
    END

```

## TEMA1.KNB

### Actualiza

#### WHEN CHANGED

##### BEGIN

```

    action OF dB3 GIRLS 1 IS open := TRUE
    FIND dB3 GIRLS 1
    WHERE variable OF dB3 GIRLS 1 <> 0
    WHEN FOUND
    action OF dB3 GIRLS 1 IS recall record := TRUE
    record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
    name := nombre OF dB3 GIRLS 1
    kalificacion1 := kalif1 OF dB3 GIRLS 1
    subtem := ulsubtema OF dB3 GIRLS 1
    subsubtem := ulsub OF dB3 GIRLS 1
    action OF dB3 GIRLS 1 IS pack := TRUE
    FIND END

```

##### BEGIN

```

    IF subtem = 0.0 THEN
        ASK TEMA1
    IF subtem = 1.1 THEN
        ASK pag1estructura
    IF subtem = 1.2 THEN
        ASK pag2corteza cerebral
    IF subtem = 1.3 AND subsubtem = 0 THEN
        ASK pag3bosfundefi
    ELSE
        IF subsubtem = 1.31 THEN
            ASK pag10
        IF subsubtem = 1.32 THEN
            ASK pag12
        IF subsubtem = 1.33 THEN
            ASK pag11
        IF subsubtem = 1.34 THEN
            ASK pag15

```

```

IF subsubtem = 1.35 THEN
  enabled OF exam := TRUE
IF subsubtem = 1.35 THEN
  ASK pag13
END
END

```

```

Ante OF figuras
WHEN CHANGED
BEGIN
  visible OF win1 := TRUE
  output OF win1 := hip1
END

```

```

Ante1 OF figuras
WHEN CHANGED
BEGIN
  visible OF win3 := TRUE
  output OF win3 := hip3
END

```

```

Apré OF figuras
WHEN CHANGED
BEGIN
  visible OF win9 := TRUE
  output OF win9 := hip9
END

```

```

Cambio OF tedos
WHEN CHANGED
BEGIN
  action OF dB3 GIRLS 1 IS open := TRUE
  \ FIND dB3 GIRLS 1
  WHERE variable OF dB3 GIRLS 1 < 0
  WHEN FOUND
    action OF dB3 GIRLS 1 IS recall record := TRUE
    record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
    name := nombre OF dB3 GIRLS 1
    kalificacion1 := kalif1 OF dB3 GIRLS 1
    access OF dB3 GIRLS 1 IS write := TRUE
    ultsubtema OF dB3 GIRLS 1 := subtem
    ultsub OF dB3 GIRLS 1 := subsubtem
    action OF dB3 GIRLS 1 IS pack := TRUE

```

---

```
FIND END
title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL MENU
PRINCIPAL"
CHAIN "MENU5"
END
```

**Corte OF figuras**

```
WHEN CHANGED
BEGIN
visible OF win5 := TRUE
output OF win5 := hip5
END
```

**Examen**

```
WHEN CHANGED
BEGIN
action OF dB3 GIRLS 1 IS open := TRUE
FIND dB3 GIRLS 1
WHERE variable OF dB3 GIRLS 1 <> 0
WHEN FOUND
action OF dB3 GIRLS 1 IS recall record := TRUE
record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
name := nombre OF dB3 GIRLS 1
kalificacion1 := kalif1 OF dB3 GIRLS 1
access OF dB3 GIRLS 1 IS write := TRUE
ultsubtema OF dB3 GIRLS 1 := subtem
ultsub OF dB3 GIRLS 1 := subsubtem
action OF dB3 GIRLS 1 IS pack := TRUE
FIND END
pasa OF tedos := TRUE
END
```

**Fig1 OF figuras**

```
WHEN CHANGED
BEGIN
visible OF neurona := TRUE
output OF neurona := figneu
END
```

**Fig2 OF figuras**  
WHEN CHANGED  
BEGIN  
  visible OF anatomi := TRUE  
  output OF anatomi := fiana  
END

**Fig3 OF figuras**  
WHEN CHANGED  
BEGIN  
  visible OF topolo := TRUE  
  output OF topolo := topog  
END

**Histo OF figuras**  
WHEN CHANGED  
BEGIN  
  visible OF win7 := TRUE  
  output OF win7 := hip7  
END

**Koncep**  
WHEN CHANGED  
BEGIN  
  visible OF win2 := FALSE  
  visible OF main window := TRUE  
END

**Noti OF figuras**  
WHEN CHANGED  
BEGIN  
  visible OF win2 := TRUE  
  output OF win2 := hip2  
END

**Pasa OF tedos**  
WHEN CHANGED  
BEGIN  
  IF kalificacion1 < 7 THEN  
    title OF main window := "ESPERA UN MOMENTO ESTOY BUSCANDO EL  
EXAMEN UNO"  
  IF kalificacion1 < 7 THEN

```
CHAIN "examen1"  
IF kalificacion1 >= 7 THEN  
  ASK exam1  
END
```

```
Prue OF tedos  
WHEN CHANGED  
BEGIN  
  ASK prueba  
END
```

```
Punte Of figuras  
WHEN CHANGED  
BEGIN  
  visible OF win8 := TRUE  
  output OF win8 := hip8  
END
```

```
Red OF figuras  
WHEN CHANGED  
BEGIN  
  visible OF win6 := TRUE  
  output OF win6 := hip6  
END
```

```
Sub1 OF subtemas  
WHEN CHANGED  
BEGIN  
  subtem := 1.1  
END
```

```
Sub2 OF subtemas  
WHEN CHANGED  
BEGIN  
  subtem := 1.2  
END
```

```
Sub3 OF subtemas  
WHEN CHANGED  
BEGIN  
  subtem := 1.3  
END
```

**Sub4 OF subtemas**

WHEN CHANGED

BEGIN

subtem := 1.31

END

**Sub5 OF subtemas**

WHEN CHANGED

BEGIN

subtem := 1.32

END

**Sub6 OF subtemas**

WHEN CHANGED

BEGIN

subtem := 1.33

END

**Sub7 OF subtemas**

WHEN CHANGED

BEGIN

subtem := 1.34

END

**Sub8 OF subtemas**

WHEN CHANGED

BEGIN

subtem := 1.35

enabled OF exam := TRUE

END

**Tema OF subtemas**

WHEN CHANGED

BEGIN

ultem := 1.0

END.

## TEMA2.KNB

### Actualiza

WHEN CHANGED

BEGIN

action OF dB3 GIRLS 1 IS open := TRUE

FIND dB3 GIRLS 1

WHERE variable OF dB3 GIRLS 1  $\diamond$  0

WHEN FOUND

name := nombre OF dB3 GIRLS 1

subtem := ultsubtema OF dB3 GIRLS 1

ulsubtem := ultsub OF dB3 GIRLS 1

action OF dB3 GIRLS 1 IS close := TRUE

FIND END

BEGIN

IF subtem = 2.1 THEN

ASK Npag1

IF subtem = 2.2 THEN

ASK Ppag1

IF subtem = 2.3 THEN

ASK Apag1

IF subtem = 2.4 THEN

enabled OF exam := TRUE

IF subtem = 2.4 THEN

ASK Mpag1

END

END

### Examen

WHEN CHANGED

BEGIN

action OF dB3 GIRLS 1 IS open := TRUE

FIND dB3 GIRLS 1

WHERE variable OF dB3 GIRLS 1  $\diamond$  0

WHEN FOUND

action OF dB3 GIRLS 1 IS recall record := TRUE

record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1

name := nombre OF dB3 GIRLS 1

kalificacion2 := kalif2 OF dB3 GIRLS 1

access OF dB3 GIRLS 1 IS write := TRUE

ultsubtema OF dB3 GIRLS 1 := subtem

```

    ultsub OF dB3 GIRLS 1 := subsubtem
    action OF dB3 GIRLS 1 IS pack := TRUE
    FIND END
    pasa OF subtemas := TRUE
    END

```

```

Fig1 OF decidir
    WHEN CHANGED
    BEGIN
        visible OF nfig1 := TRUE
        output OF nfig1 := npfig1
    END

```

```

Fig2 OF decidir
    WHEN CHANGED
    BEGIN
        visible OF perfig2 := TRUE
        output OF perfig2 := pfig2
    END

```

```

Fig3 OF decidir
    WHEN CHANGED
    BEGIN
        visible OF adafig3 := TRUE
        output OF adafig3 := afig3
    END

```

```

Fig4 Of decidir
    WHEN CHANGED
    BEGIN
        visible OF madaline4 := TRUE
        output OF madaline4 := mfig4
    \END

```

```

Pasa OF subtemas
    WHEN CHANGED
    BEGIN
        IF kalificacion2 < 7 THEN
            title OF main window := "ESPERA UN MOMENTO ESTOY BUSCANDO EL
    EXAMEN DOS"
            IF kalificacion2 < 7 THEN
                CHAIN "examen2"
            IF kalificacion2 >= 7 THEN

```

ASK exm2  
END

**Salir OF decidir**  
**WHEN CHANGED**  
**BEGIN**  
 action OF dB3 GIRLS 1 IS open := TRUE  
 FIND dB3 GIRLS 1  
 WHERE variable OF dB3 GIRLS 1  $\triangleleft$  0  
**WHEN FOUND**  
 action OF dB3 GIRLS 1 IS recall record := TRUE  
 record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1  
 name := nombre OF dB3 GIRLS 1  
 access OF dB3 GIRLS 1 IS write := TRUE  
 ultsubtema OF dB3 GIRLS 1 := subtem  
 ultema OF dB3 GIRLS 1 := ultem  
 action OF dB3 GIRLS 1 IS pack := TRUE  
 action OF dB3 GIRLS 1 IS close := TRUE  
**FIND END**  
**CHAIN "MENUS"**  
**END**

**Sub1 OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
 subtem := 2.1  
**END**

**Sub2OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
 subtem := 2.2  
**END**

**Sub3 OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
 subtem := 2.3  
**END**

**Sub4 OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
 subtem := 2.4  
**END**

**Tema OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
 ultem := 2.0  
**END**

**Tex1 OF textos**  
**WHEN CHANGED**  
**BEGIN**  
 visible OF lms := TRUE  
 output OF lms := lmsl  
**END**

### **TEMA3.KNB**

**Actualiza**  
**WHEN CHANGED**  
**BEGIN**  
 action OF dB3 GIRLS 1 IS open := TRUE  
 FIND dB3 GIRLS 1  
 WHERE variable OF dB3 GIRLS 1  $\diamond$  0  
 WHEN FOUND  
 name := nombre OF dB3 GIRLS 1  
 subtem := ulsubtema OF dB3 GIRLS 1  
 ulsubtem := ulsub OF dB3 GIRLS 1  
 action OF dB3 GIRLS 1 IS close := TRUE  
**FIND END**  
**BEGIN**  
 IF subtem = 3.1 THEN  
 ASK Hpagl  
 IF subtem = 3.2 THEN  
 ASK Rpagl  
 IF subtem = 3.3 THEN  
 ASK Kpagl  
 IF subtem = 3.4 THEN  
 ASK Gpagl

```

    IF subtem = 3.5 THEN
        enabled OF pictbtn 1 := TRUE
    IF subtem = 3.5 THEN
        ASK Apag1
    END
END

```

**Examen OF subtemas**

WHEN CHANGED

```

BEGIN
    action OF dB3 GIRLS 1 IS open := TRUE
    FIND dB3 GIRLS 1
    WHERE variable OF dB3 GIRLS 1 <> 0
    WHEN FOUND
        action OF dB3 GIRLS 1 IS recall record := TRUE
        record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
        name := nombre OF dB3 GIRLS 1
        kalificacion3 := kalif3 OF dB3 GIRLS 1
        access OF dB3 GIRLS 1 IS write := TRUE
        ultsubtema OF dB3 GIRLS 1 := subtem
        ultsub OF dB3 GIRLS 1 := subsubtem
        action OF dB3 GIRLS 1 IS pack := TRUE
    FIND END
    pasa OF tomar := TRUE
END

```

**Fig1 OF tomar**

WHEN CHANGED

```

BEGIN
    visible OF hopf1 := TRUE
    output OF hopf1 := hfig1
END

```

**Fig2 OF tomar**

WHEN CHANGED

```

BEGIN
    visible OF rfigu2 := TRUE
    output OF rfigu2 := rfig2
END

```

**Fig3 OF tomar**  
**WHEN CHANGED**  
**BEGIN**  
 visible OF atext1 := TRUE  
 output OF atext1 := atex1  
**END**

**Fig4 OF tomar**  
**WHEN CHANGED**  
**BEGIN**  
 visible OF atext2 := TRUE  
 output OF atext2 := atex2  
**END**

**Pasa OF tomar**  
**WHEN CHANGED**  
**BEGIN**  
 IF kalificacion3 < 7 THEN  
 title OF main window := "ESPERA UN MOMENTO ESTOY BUSCANDO EL  
**EXAMEN TRES"**  
 IF kalificacion3 < 7 THEN  
 CHAIN "examen3"  
 IF kalificacion3 >= 7 THEN  
 ASK exm3  
**END**

**Salir OF tomar**  
**WHEN CHANGED**  
**BEGIN**  
 action OF dB3 GIRLS 1 IS open := TRUE  
 FIND dB3 GIRLS 1  
 WHERE variable OF dB3 GIRLS 1 <> 0  
**WHEN FOUND**  
 action OF dB3 GIRLS 1 IS recall record := TRUE  
 record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1  
 name := nombre OF dB3 GIRLS 1  
 access OF dB3 GIRLS 1 IS write := TRUE  
 ultsubtema OF dB3 GIRLS 1 := subtem  
 ultema OF dB3 GIRLS 1 := ultem  
 action OF dB3 GIRLS 1 IS pack := TRUE  
 action OF dB3 GIRLS 1 IS close := TRUE  
**FIND END**  
 CHAIN "MENU5"

**Sub1 OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
    subtem := 3.1  
**END**

**Sub2 OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
    subtem := 3.2  
**END**

**Sub3 OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
    subtem := 3.3  
**END**

**Sub4 OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
    subtem := 3.4  
**END**

**Sub5 OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
    subtem := 3.5  
**END**

**Tema OF subtemas**  
**WHEN CHANGED**  
**BEGIN**  
    ultem := 3.0  
**END**

# APENDICE B

---

---

---

---

## APENDICE B

### Listado del PRL de SETURNA

#### TUTORIAL.PRL

\$VERSION25

\$LOCATIONS ARE PIXELS

CLASS variable

WITH dato SIMPLE

WHEN CHANGED

BEGIN

title OF main window := " UN MOMENTO POR FAVOR "

CHAIN "DATOS"

END

WITH ayuda1 SIMPLE

WHEN CHANGED

BEGIN

ASK ayuda

BEGIN

texto OF variable := text OF tx1

END

END

WITH intro SIMPLE

WHEN CHANGED

BEGIN

ASK introducción

END

WITH te SIMPLE

WHEN CHANGED

BEGIN

pagina OF variable := pagina OF variable + 1

BEGIN

IF pagina OF variable = 2 THEN

texto OF variable := text OF tx2

IF pagina OF variable = 3 THEN

texto OF variable := text OF tx3

IF pagina OF variable = 4 THEN

texto OF variable := text OF tx4

IF pagina OF variable = 5 THEN

texto OF variable := text OF tx5

IF pagina OF variable = 6 THEN

texto OF variable := text OF tx6

IF pagina OF variable = 7 THEN

```

    texto OF variable := text OF tx7
    IF pagina OF variable = 8 THEN
        texto OF variable := text OF tx8
    IF pagina OF variable = 9 THEN
        texto OF variable := text OF tx9
    IF pagina OF variable = 10 THEN
        texto OF variable := text OF tx10
    END
END

```

```

WITH texto STRING
WITH pagina NUMERIC
WITH regreso SIMPLE
WHEN CHANGED
    BEGIN
        re OF variable := TRUE
    END
WITH re SIMPLE
WHEN CHANGED
    BEGIN
        pagina OF variable := pagina OF variable - 1
    END

```

```

    BEGIN
        IF pagina OF variable = 1 THEN
            texto OF variable := text OF tx1
        IF pagina OF variable = 2 THEN
            texto OF variable := text OF tx2
        IF pagina OF variable = 3 THEN
            texto OF variable := text OF tx3
        IF pagina OF variable = 4 THEN
            texto OF variable := text OF tx4
        IF pagina OF variable = 5 THEN
            texto OF variable := text OF tx5
        IF pagina OF variable = 6 THEN
            texto OF variable := text OF tx6
        IF pagina OF variable = 7 THEN
            texto OF variable := text OF tx7
        IF pagina OF variable = 8 THEN
            texto OF variable := text OF tx8
        IF pagina OF variable = 9 THEN
            texto OF variable := text OF tx9
        IF pagina OF variable = 10 THEN
            texto OF variable := text OF tx10
        END
    END

```

```

INSTANCE variable 1 ISA variable
WITH pagina := 1

```

```

ATTRIBUTE pag SIMPLE
WHEN CHANGED
BEGIN
    te OF variable := TRUE
END
    
```

## DATOS.PRL

```

$VERSION25
$LOCATIONS ARE PIXELS
CLASS basedatos
    WITH nclave STRING
    WITH nombre STRING
    WITH appaterno STRING
    WITH apmaterno STRING
    WITH examen1 NUMERIC
    WITH examen2 NUMERIC
    WITH examen3 NUMERIC
    WITH examen4 NUMERIC
    WITH examen5 NUMERIC
    WITH examen6 NUMERIC
    WITH promedio NUMERIC
    WITH ultimo tema STRING
    WITH ultimosubtema STRING
    WITH otro STRING
    WITH otro1 STRING
    WITH otro2 STRING
CLASS dB3_GIRLS SINGLE EXTERNAL "dBASEIII "
    WITH klave STRING
        SEARCH ORDER CONTEXT
    WITH nombre STRING
        SEARCH ORDER CONTEXT
    WITH paternø STRING
        SEARCH ORDER CONTEXT
    WITH materno STRING
        SEARCH ORDER CONTEXT
    WITH kalif1 NUMERIC
        SEARCH ORDER CONTEXT
    WITH kalif2 NUMERIC
        SEARCH ORDER CONTEXT
    WITH kalif3 NUMERIC
    WITH kalif4 NUMERIC
    WITH kalif5 NUMERIC
    WITH kalif6 NUMERIC
    WITH kaltot NUMERIC
    WITH ultema NUMERIC
    WITH ultsubtema NUMERIC
    WITH variable NUMERIC
    
```

```

INSTANCE GIRLS 1 ISA dB3_GIRLS
  WITH access IS write shared
  WITH action IS open
  WITH filename := "C:\USUARIOS\RIB\GIRLS.DBF"
  WITH default error handling := TRUE

CLASS procedimientos
  WITH grabadat SIMPLE
  WHEN CHANGED
  BEGIN
    action OF dB3_GIRLS IS open := TRUE
  FIND dB3_GIRLS
    WHERE klaveprincipal = klave OF dB3_GIRLS
  WHEN FOUND
    ASK avisol
    visible OK button OF main window := TRUE
    ASK datos
  WHEN NONE FOUND
    action OF dB3_GIRLS IS append record := TRUE
    klave OF dB3_GIRLS := klaveprincipal
    nombre OF dB3_GIRLS := name
    paterno OF dB3_GIRLS := appaterno1
    materno OF dB3_GIRLS := apmaterno2
    kalif1 OF dB3_GIRLS := kalificacion1
    kalif2 OF dB3_GIRLS := kalificacion2
    kalif3 OF dB3_GIRLS := kalificacion3
    kalif4 OF dB3_GIRLS := kalificacion4
    kalif5 OF dB3_GIRLS := kalificacion5
    kalif6 OF dB3_GIRLS := kalificacion6
    kaltot OF dB3_GIRLS := total
    ultima OF dB3_GIRLS := ultem
    ultsubtema OF dB3_GIRLS := ultsubtem
    variable OF dB3_GIRLS := record OF dB3_GIRLS + 2
    action OF dB3_GIRLS IS pack := TRUE
    action OF dB3_GIRLS IS close := TRUE
  FIND END
  ASK opciones1
  END
  WITH tabla SIMPLE
  WITH encuentrame SIMPLE
  WITH cambia SIMPLE
  WHEN CHANGED
  BEGIN
    CHAIN "menu5"
  END
  WITH examenc SIMPLE
  WHEN CHANGED
  BEGIN
    CHAIN "texam1"
  END

```

```

WITH buscame SIMPLE
WHEN CHANGED
BEGIN
  action OF dB3_GIRLS IS open := TRUE
  FIND dB3_GIRLS
  WHERE klaveprincipal = klave OF dB3_GIRLS
  WHEN FOUND
  ASK aviso1
  visible OK button OF main window := TRUE
  ASK datos
  FIND END
  action OF dB3_GIRLS IS pack := TRUE
END
WITH verificar SIMPLE
WHEN CHANGED
BEGIN
  IF ultem = 1.0 THEN
    CHAIN "menu5"
  ELSE
    CHAIN "menu5"
  END
WITH boton SIMPLE
WHEN CHANGED
BEGIN
  ASK clave
END
INSTANCE variables 1 ISA variables
  WITH shances := 3
ATTRIBUTE klaveprincipal STRING
ATTRIBUTE name STRING
ATTRIBUTE appaterno1 STRING
ATTRIBUTE apmaterno2 STRING
ATTRIBUTE kalificacion1 NUMERIC
ATTRIBUTE kalificacion2 NUMERIC
ATTRIBUTE kalificacion3 NUMERIC
ATTRIBUTE kalificacion4 NUMERIC
ATTRIBUTE kalificacion5 NUMERIC
ATTRIBUTE kalificacion6 NUMERIC
ATTRIBUTE total NUMERIC
ATTRIBUTE ultem NUMERIC
ATTRIBUTE ultsubtem NUMERIC
ATTRIBUTE mensa SIMPLE
WHEN CHANGED
BEGIN
  IF shances OF variables > 0 THEN
    BEGIN
      visible OF mensaje2 := TRUE
      output OF mensaje2 := mensaje1
      FOR (num := 1 TO 5)  END
    ELSE
      FOR (num := 1 TO 5)  exit OF application := TRUE
    END
  END

```

```

WHEN CHANGED
BEGIN
  action OF GIRLS 1 IS open := TRUE
  FIND dB3_GIRLS
  WHERE key = klave OF dB3_GIRLS
  WHEN FOUND
    access OF dB3_GIRLS IS write := TRUE
    variable OF dB3_GIRLS := record OF dB3_GIRLS + 1
    klaveprincipal := klave OF dB3_GIRLS
    name := nombre OF dB3_GIRLS
    appaterno1 := paterno OF dB3_GIRLS
    apmaterno2 := materno OF dB3_GIRLS
    kalificacion1 := kalif1 OF dB3_GIRLS
    kalificacion2 := kalif2 OF dB3_GIRLS
    kalificacion3 := kalif3 OF dB3_GIRLS
    kalificacion4 := kalif4 OF dB3_GIRLS
    kalificacion5 := kalif5 OF dB3_GIRLS
    kalificacion6 := kalif6 OF dB3_GIRLS
    total := kaltot OF dB3_GIRLS
    ultem := ultema OF dB3_GIRLS
    ultsubtem := ultsubtema OF dB3_GIRLS
    manda OF variables := TRUE
    action OF GIRLS 1 IS close := TRUE
  WHEN NONE FOUND
    action OF GIRLS 1 IS close := TRUE
    shances OF variables := shances OF variables - 1
    mensa := TRUE
  FIND END
END
ATTRIBUTE oprtunidad NUMERIC
END

```

### **MENUS.PRL**

```

$VERSION25
$LOCATIONS ARE PIXELS
CLASS dB3 GIRLS 1 SINGLE EXTERNAL "dBASEIII "
  WITH klave STRING
    SEARCH ORDER CONTEXT
  WITH nombre STRING
    SEARCH ORDER CONTEXT
  WITH paterno STRING
    SEARCH ORDER CONTEXT
  WITH materno STRING
    SEARCH ORDER CONTEXT
  WITH kalif1 NUMERIC
    SEARCH ORDER CONTEXT
  WITH kalif2 NUMERIC
    SEARCH ORDER CONTEXT
  WITH kalif3 NUMERIC

```

```

SEARCH ORDER CONTEXT
WITH kalif4 NUMERIC
  SEARCH ORDER CONTEXT
WITH kalif5 NUMERIC
  SEARCH ORDER CONTEXT
WITH kalif6 NUMERIC
  SEARCH ORDER CONTEXT
WITH kaltot NUMERIC
  SEARCH ORDER CONTEXT
WITH ultema NUMERIC
  SEARCH ORDER CONTEXT
WITH ultsubtema NUMERIC
  SEARCH ORDER CONTEXT
WITH opor NUMERIC
  SEARCH ORDER CONTEXT
WITH variable NUMERIC
CLASS temas
WITH tema1 SIMPLE
  WHEN CHANGED
  BEGIN
    ultem := 1
    action OF dB3 GIRLS 1 IS open := TRUE
    FIND dB3 GIRLS 1
    WHERE variable OF dB3 GIRLS 1 <> 0
    WHEN FOUND
      action OF dB3 GIRLS 1 IS recall record := TRUE
      record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
      access OF dB3 GIRLS 1 IS write := TRUE
      ultema OF dB3 GIRLS 1 := ultem
      action OF dB3 GIRLS 1 IS pack := TRUE
      action OF dB3 GIRLS 1 IS close := TRUE
    FIND END
    title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEM
A UNO"
    CHAIN "tema1"
  END
WITH tema2 SIMPLE
  WHEN CHANGED
  BEGIN
    ultem := 2
    action OF dB3 GIRLS 1 IS open := TRUE
    FIND dB3 GIRLS 1
    WHERE variable OF dB3 GIRLS 1 <> 0
    WHEN FOUND
      action OF dB3 GIRLS 1 IS recall record := TRUE
      record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
      access OF dB3 GIRLS 1 IS write := TRUE
      ultema OF dB3 GIRLS 1 := ultem
      action OF dB3 GIRLS 1 IS pack := TRUE
      action OF dB3 GIRLS 1 IS close := TRUE
    FIND END

```

```

title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEMA DOS"
CHAIN "tema2"
END
WITH tema3 SIMPLE
WHEN CHANGED
BEGIN
ultem := 3
action OF dB3 GIRLS 1 IS open := TRUE
FIND dB3 GIRLS 1
WHERE variable OF dB3 GIRLS 1 <> 0
WHEN FOUND
action OF dB3 GIRLS 1 IS recall record := TRUE
record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
access OF dB3 GIRLS 1 IS write := TRUE
ultema OF dB3 GIRLS 1 := ultem
action OF dB3 GIRLS 1 IS pack := TRUE
action OF dB3 GIRLS 1 IS close := TRUE
FIND END
title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEM
A TRES"
CHAIN "tema3"
END
WITH tema4 SIMPLE
WHEN CHANGED
BEGIN
ultem := 4
action OF dB3 GIRLS 1 IS open := TRUE
FIND dB3 GIRLS 1
WHERE variable OF dB3 GIRLS 1 <> 0
WHEN FOUND
action OF dB3 GIRLS 1 IS recall record := TRUE
record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
access OF dB3 GIRLS 1 IS write := TRUE
ultema OF dB3 GIRLS 1 := ultem
action OF dB3 GIRLS 1 IS pack := TRUE
action OF dB3 GIRLS 1 IS close := TRUE
FIND END
title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEM
A CUATRO"
CHAIN "tema4"
END
WITH tema5 SIMPLE
WHEN CHANGED
BEGIN
ultem := 5
action OF dB3 GIRLS 1 IS open := TRUE
FIND dB3 GIRLS 1
WHERE variable OF dB3 GIRLS 1 <> 0
WHEN FOUND
action OF dB3 GIRLS 1 IS recall record := TRUE
record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1

```

```

    access OF dB3 GIRLS 1 IS write := TRUE
    ultima OF dB3 GIRLS 1 := ultem
    action OF dB3 GIRLS 1 IS pack := TRUE
    action OF dB3 GIRLS 1 IS close := TRUE
    FIND END
    title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEM
A CINCO"
    CHAIN "tema5"
    END
    WITH tema6 SIMPLE
    WHEN CHANGED
    BEGIN
        ultem := 6
        action OF dB3 GIRLS 1 IS open := TRUE
        FIND dB3 GIRLS 1
        WHERE variable OF dB3 GIRLS 1 <> 0
        WHEN FOUND
            action OF dB3 GIRLS 1 IS recall record := TRUE
            record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
            access OF dB3 GIRLS 1 IS write := TRUE
            ultima OF dB3 GIRLS 1 := ultem
            action OF dB3 GIRLS 1 IS pack := TRUE
            action OF dB3 GIRLS 1 IS close := TRUE
        FIND END
        title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL TEM
A SEIS"
        CHAIN "tema6"
        END
        WITH salida SIMPLE
        WITH tutorial SIMPLE
        WHEN CHANGED
        BEGIN
            CHAIN "TUTORIAL"
        END
        CLASS var
        WITH inicio SIMPLE
        WITH actual SIMPLE
        CLASS variables
        WITH manda SIMPLE
        WITH sal SIMPLE
        WHEN CHANGED
        BEGIN
            ASK salida1
        END
        ATTRIBUTE salir SIMPLE
        WHEN CHANGED
        BEGIN
            action OF dB3 GIRLS 1 IS open := TRUE
            FIND dB3 GIRLS 1
            WHERE variable OF dB3 GIRLS 1 <> 0
            WHEN FOUND

```

```

action OF dB3 GIRLS 1 IS recall record := TRUE
access OF dB3 GIRLS 1 IS write := TRUE
variable OF dB3 GIRLS 1 := 0
    action OF dB3 GIRLS 1 IS pack := TRUE
    action OF dB3 GIRLS 1 IS close := TRUE
FIND END
exit OF application := TRUE
END
WHEN CHANGED
BEGIN
    action OF GIRLS 1 IS open := TRUE
    FIND dB3 GIRLS 1
    WHERE variable OF dB3 GIRLS 1 <> 0
    WHEN FOUND
        klaveprincipal := klave OF dB3 GIRLS 1
        name := nombre OF dB3 GIRLS 1
        appaterno1 := paterno OF dB3 GIRLS 1
        apmaterno2 := materno OF dB3 GIRLS 1
        kalificacion1 := kalif1 OF dB3 GIRLS 1
        kalificacion2 := kalif2 OF dB3 GIRLS 1
        kalificacion3 := kalif3 OF dB3 GIRLS 1
        kalificacion4 := kalif4 OF dB3 GIRLS 1
        kalificacion5 := kalif5 OF dB3 GIRLS 1
        kalificacion6 := kalif6 OF dB3 GIRLS 1
        total := kaltot OF dB3 GIRLS 1
        ultem := ultema OF dB3 GIRLS 1
        ulsubtem := ultsubtema OF dB3 GIRLS 1
        manda OF variables := TRUE
        action OF GIRLS 1 IS close := TRUE
    FIND END
    IF kalificacion1 >= 7 THEN
        BEGIN
            enabled OF bt2 := TRUE
            IF kalificacion2 >= 7 THEN
                BEGIN
                    enabled OF bt3 := TRUE
                    IF kalificacion3 >= 7 THEN
                        BEGIN
                            enabled OF bt4 := TRUE
                            IF kalificacion4 >= 7 THEN
                                BEGIN
                                    enabled OF bt5 := TRUE
                                    IF kalificacion5 >= 7 THEN
                                        BEGIN
                                            enabled OF bt6 := TRUE
                                        END
                                    END
                                END
                            END
                        END
                    END
                END
            END
        END
    END
END
END
END
END
END

```

```
DEMON 1
IF inicio OF var
THEN buscar := TRUE
```

### TEMA1.PRL

```
$VERSION25
$LOCATIONS ARE PIXELS
```

```
CLASS dB3 GIRLS 1 SINGLE EXTERNAL "dBASEIII "
WITH klave STRING
  SEARCH ORDER CONTEXT
WITH nombre STRING
  SEARCH ORDER CONTEXT
WITH paterno STRING
  SEARCH ORDER CONTEXT
WITH materno STRING
  SEARCH ORDER CONTEXT
WITH kalif1 NUMERIC
  SEARCH ORDER CONTEXT
WITH kalif2 NUMERIC
  SEARCH ORDER CONTEXT
WITH kalif3 NUMERIC
  SEARCH ORDER CONTEXT
WITH kalif4 NUMERIC
  SEARCH ORDER CONTEXT
WITH kalif5 NUMERIC
  SEARCH ORDER CONTEXT
WITH kalif6 NUMERIC
  SEARCH ORDER CONTEXT
WITH kaltot NUMERIC
  SEARCH ORDER CONTEXT
WITH ultima NUMERIC
  SEARCH ORDER CONTEXT
WITH ultsubtema NUMERIC
  SEARCH ORDER CONTEXT
WITH opor NUMERIC
  SEARCH ORDER CONTEXT
WITH variable NUMERIC
  SEARCH ORDER CONTEXT
WITH ultsub NUMERIC
WITH actualiza SIMPLE
WITH koncep SIMPLE
WHEN CHANGED
BEGIN
  visible OF win2 := FALSE
  visible OF main window := TRUE
END
```

```

INSTANCE GIRLS 1 ISA dB3 GIRLS 1
  WITH access IS write shared
  WITH action IS open
  WITH filename := "C:\USUARIOS\RIB\GIRLS.DBF"
  WITH default error handling := TRUE

```

```

CLASS figuras
  WITH ante SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF win1 := TRUE
    output OF win1 := hip1
  END

```

```

  WITH noti SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF win2 := TRUE
    output OF win2 := hip2
  END

```

```

  WITH ante1 SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF win3 := TRUE
    output OF win3 := hip3
  END

```

```

  WITH defi SIMPLE
  WITH corte SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF win5 := TRUE
    output OF win5 := hip5
  END

```

```

  WITH red SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF win6 := TRUE
    output OF win6 := hip6
  END

```

```

  WITH histo SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF win7 := TRUE
    output OF win7 := hip7
  END

```

```

  WITH punte SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF win8 := TRUE
    output OF win8 := hip8
  END

```

```

  WITH fig1 SIMPLE

```

```

WHEN CHANGED
BEGIN
  visible OF neurona := TRUE
  output OF neurona := figneu
END
WITH fig2 SIMPLE

```

```

WHEN CHANGED
BEGIN
  visible OF anatomi := TRUE
  output OF anatomi := fiana
END

```

```

WITH apre SIMPLE
WHEN CHANGED
BEGIN
  visible OF win9 := TRUE
  output OF win9 := hip9
END

```

```

WITH fig3 SIMPLE
WHEN CHANGED
BEGIN
  visible OF topolo := TRUE
  output OF topolo := topog
END

```

```

CLASS subtemas
WITH sub1 SIMPLE
WHEN CHANGED
BEGIN
  subtem := 1.1
END

```

```

WITH sub2 SIMPLE
WHEN CHANGED
BEGIN
  subtem := 1.2
END

```

```

WITH sub3 SIMPLE
WHEN CHANGED
BEGIN
  subtem := 1.3
END

```

```

WITH sub4 SIMPLE
WHEN CHANGED
BEGIN
  subsubtem := 1.31
END

```

```

WITH sub5 SIMPLE
WHEN CHANGED
BEGIN
  subsubtem := 1.32
END

```

```

WITH tema SIMPLE

```

```

WHEN CHANGED
  BEGIN
    ultem := 1.0
  END
WITH sub6 SIMPLE

WHEN CHANGED
  BEGIN
    subsubtem := 1.33
  END
WITH sub7 SIMPLE
  WHEN CHANGED
    BEGIN
      subsubtem := 1.34
    END
WITH sub8 SIMPLE
  WHEN CHANGED
    BEGIN
      subsubtem := 1.35
      enabled OF exam := TRUE
    END
CLASS tedos
  WITH cambio SIMPLE
    WHEN CHANGED
      BEGIN
        action OF dB3 GIRLS 1 IS open := TRUE
        FIND dB3 GIRLS 1
        WHERE variable OF dB3 GIRLS 1 <> 0
        WHEN FOUND
          action OF dB3 GIRLS 1 IS recall record := TRUE
          record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
          name := nombre OF dB3 GIRLS 1
          kalificacion1 := kalif1 OF dB3 GIRLS 1
          access OF dB3 GIRLS 1 IS write := TRUE
          ulsubtema OF dB3 GIRLS 1 := subtem
          ulsub OF dB3 GIRLS 1 := subsubtem
          acción OF dB3 GIRLS 1 IS pack := TRUE
        FIND END
        title OF main window := " ESPERA UN MOMENTO ESTOY PASANDO AL MEN
U PRINCIPAL"
        CHAIN "MENUS"
      END
  WITH prue SIMPLE
    WHEN CHANGED
      BEGIN
        ASK prueba
      END

```

```
WITH pasa SIMPLE
WHEN CHANGED
BEGIN
  IF kalificacion1 < 7 THEN
    title OF main window := "ESPERA UN MOMENTO ESTOY BUSCANDO EL EXAMEN UNO"
  IF kalificacion1 < 7 THEN
    CHAIN "examen1"
  IF kalificacion1 >= 7 THEN
    ASK exam1
END
```

```
WHEN CHANGED
BEGIN
  action OF dB3 GIRLS 1 IS open := TRUE
  FIND dB3 GIRLS 1
  WHERE variable OF dB3 GIRLS 1 <> 0
  WHEN FOUND
    action OF dB3 GIRLS 1 IS recall record := TRUE
    record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
    name := nombre OF dB3 GIRLS 1
    kalificacion1 := kalif1 OF dB3 GIRLS 1
    subtem := ultsubtema OF dB3 GIRLS 1
    subsubtem := ultsub OF dB3 GIRLS 1
    action OF dB3 GIRLS 1 IS pack := TRUE
  FIND END
```

```
BEGIN
  IF subtem = 0.0 THEN
    ASK TEMA1
  IF subtem = 1.1 THEN
    ASK pag1estructura
  IF subtem = 1.2 THEN
    ASK pag2corteza cerebral
  IF subtem = 1.3 AND subsubtem = 0 THEN
    ASK pag3bosfundefi
  ELSE
    IF subsubtem = 1.31 THEN
      ASK pag10
    IF subsubtem = 1.32 THEN
      ASK pag12
    IF subsubtem = 1.33 THEN
      ASK pag11
    IF subsubtem = 1.34 THEN
      ASK pag15
    IF subsubtem = 1.35 THEN
      enabled OF exam := TRUE
    IF subsubtem = 1.35 THEN
      ASK pag13
  END
END
```

```

WHEN CHANGED
BEGIN
  action OF dB3 GIRLS 1 IS open := TRUE
  FIND dB3 GIRLS 1
  WHERE variable OF dB3 GIRLS 1 <> 0
  WHEN FOUND
    action OF dB3 GIRLS 1 IS recall record := TRUE
    record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
    name := nombre OF dB3 GIRLS 1
    kalificacion1 := kalif1 OF dB3 GIRLS 1
    access OF dB3 GIRLS 1 IS write := TRUE
    ultsubtema OF dB3 GIRLS 1 := subtem
    ultsub OF dB3 GIRLS 1 := subsubtem
    action OF dB3 GIRLS 1 IS pack := TRUE
  FIND END
  pasa OF tedos := TRUE
END
END

```

### EXAMEN1.PRL

```

$VERSION25
$LOCATIONS ARE PIXELS
CLASS dB3 GIRLS 1 SINGLE EXTERNAL "dBASEIII "
  WITH clave STRING
    SEARCH ORDER CONTEXT
  WITH nombre STRING
    SEARCH ORDER CONTEXT
  WITH paterno STRING
    SEARCH ORDER CONTEXT
  WITH materno STRING
    SEARCH ORDER CONTEXT
  WITH kalif1 NUMERIC
    SEARCH ORDER CONTEXT
  WITH kalif2 NUMERIC
    SEARCH ORDER CONTEXT
  WITH kalif3 NUMERIC
    SEARCH ORDER CONTEXT
  WITH kalif4 NUMERIC
    SEARCH ORDER CONTEXT
  WITH kalif5 NUMERIC
    SEARCH ORDER CONTEXT
  WITH kalif6 NUMERIC
    SEARCH ORDER CONTEXT
  WITH kaltot NUMERIC
    SEARCH ORDER CONTEXT
  WITH ultema NUMERIC
    SEARCH ORDER CONTEXT
  WITH ultsubtema NUMERIC
    SEARCH ORDER CONTEXT
  WITH opor NUMERIC

```

```

SEARCH ORDER CONTEXT
WITH otro1 STRING
SEARCH ORDER CONTEXT
WITH variable NUMERIC
INSTANCE GIRLS 1 ISA dBS GIRLS 1
WITH access IS write shared
WITH action IS open
WITH filename := "C:\USUARIOS\RIB\GIRLS.DBF"
WITH default error handling := TRUE
CLASS decidir
WITH kalifica exam1 SIMPLE
WHEN CHANGED
BEGIN
    kalificacion1 := 0
    IF neu OF evaluacion = TRUE THEN
        suma OF evaluacion 1 := TRUE
    IF espa OF evaluacion = FALSE THEN
        suma OF evaluacion 1 := TRUE
    IF kito OF evaluacion = TRUE THEN
        suma OF evaluacion 1 := TRUE
    IF sinap OF evaluacion = FALSE THEN
        suma OF evaluacion 1 := TRUE
    IF kere OF evaluacion = TRUE THEN
        suma OF evaluacion 1 := TRUE
    IF dona OF evaluacion = TRUE THEN
        suma OF evaluacion 1 := TRUE
    IF hebb OF evaluacion = TRUE THEN
        suma OF evaluacion 1 := TRUE
    IF rosen OF evaluacion = FALSE THEN
        suma OF evaluacion 1 := TRUE
    IF red OF evaluacion = TRUE THEN
        suma OF evaluacion 1 := TRUE
    IF ele OF evaluacion = FALSE THEN
        suma OF evaluacion 1 := TRUE
    END
WITH kalifica examen1 SIMPLE
WHEN CHANGED
BEGIN
    kalificacion1 := 0
    IF pitts OF evaluacion = FALSE THEN
        suma OF evaluacion := TRUE
    IF klima OF evaluacion = FALSE THEN
        suma OF evaluacion := TRUE
    IF nueva OF evaluacion = FALSE THEN
        suma OF evaluacion := TRUE
    IF topo OF evaluacion = TRUE THEN
        suma OF evaluacion := TRUE
    IF peso OF evaluacion = TRUE THEN
        suma OF evaluacion := TRUE
    IF error OF evaluacion = FALSE THEN
        suma OF evaluacion := TRUE

```

```
IF kon OF evaluacion = TRUE THEN
  suma OF evaluacion := TRUE
IF habi OF evaluacion = FALSE THEN
  suma OF evaluacion := TRUE
IF kereb OF evaluacion = TRUE THEN
  suma OF evaluacion := TRUE
IF natu OF evaluacion = FALSE THEN
  suma OF evaluacion := TRUE
END
WITH kalifica exm1 SIMPLE
WHEN CHANGED
BEGIN
  kalificacion1 := 0
  IF resp OF opexam 1 IS a = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 2 IS b = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 3 IS a = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 4 IS c = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 5 IS b = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 6 IS a = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 7 IS b = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 8 IS c = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 9 IS a = TRUE THEN
    suma OF evaluacion := TRUE
  IF resp OF opexam 10 IS b = TRUE THEN
    suma OF evaluacion := TRUE
  END
WITH alea NUMERIC
WITH examen NUMERIC
WHEN CHANGED
BEGIN
  IF examen OF decidir = 1 THEN
    ASK texam1
  IF examen OF decidir = 2 THEN
    ASK tdexam1
  IF examen OF decidir = 3 THEN
    ASK txamen1
  END
WITH elige SIMPLE
WHEN CHANGED
BEGIN
  alea OF decidir := RAND
  alea OF decidir := alea OF decidir * 10
  alea OF decidir := TRUNC( alea OF decidir)
```

```

IF alea OF decidir = 0 THEN
  examen OF decidir := 1
ELSE
  IF alea OF decidir = 1 THEN
    examen OF decidir := 2
  ELSE
    IF alea OF decidir = 2 THEN
      examen OF decidir := 3
    ELSE
      IF alea OF decidir = 3 THEN
        examen OF decidir := 1
      ELSE
        IF alea OF decidir = 4 THEN
          examen OF decidir := 2
        IF alea OF decidir = 5 THEN
          examen OF decidir := 3
        ELSE
          IF alea OF decidir = 6 THEN
            examen OF decidir := 1
          ELSE
            IF alea OF decidir = 7 THEN
              examen OF decidir := 2
            ELSE
              IF alea OF decidir = 8 THEN
                examen OF decidir := 3
              ELSE
                IF alea OF decidir = 9 THEN
                  examen OF decidir := 1
                END
            END
          WHEN CHANGED
            BEGIN
              kalificacion1 := kalificacion1 + 1
            END
          CLASS variables
            WITH hola SIMPLE
              WHEN CHANGED
                BEGIN
                  IF oportunidad = 3 THEN
                    exit OF application := TRUE
                  BEGIN
                    title OF main window := "MOMENTO POR FAVOR ESTOY BUSCANDO EL
MENU PRINCIPAL"
                    CHAIN "MENUS"
                  END
                END
            WITH mensaje2 STRING
            WITH mensaje STRING
            WITH borrar SIMPLE
              WHEN CHANGED
                BEGIN
                  action OF db3 GIRLS 1 IS open := TRUE

```

```

FIND dB3 GIRLS 1
  WHERE variable OF dB3 GIRLS 1 <> 0
  WHEN FOUND
    action OF dB3 GIRLS 1 IS recall record := TRUE
    record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
    action OF dB3 GIRLS 1 IS delete record := TRUE
    action OF dB3 GIRLS 1 IS pack := TRUE
    action OF dB3 GIRLS 1 IS close := TRUE
  FIND END
END
WITH grabar SIMPLE
  WHEN CHANGED
  BEGIN
    action OF dB3 GIRLS 1 IS open := TRUE
    FIND dB3 GIRLS 1
      WHERE variable OF dB3 GIRLS 1 <> 0
      WHEN FOUND
        action OF dB3 GIRLS 1 IS recall record := TRUE
        record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
        klaveprincipal := klave OF dB3 GIRLS 1
        name := nombre OF dB3 GIRLS 1
        appaterno1 := paterno OF dB3 GIRLS 1
        apmaterno2 := materno OF dB3 GIRLS 1
        ultem := ultima OF dB3 GIRLS 1
        oportunidad := opor OF dB3 GIRLS 1
        access OF dB3 GIRLS 1 IS write := TRUE
        kalif1 OF dB3 GIRLS 1 := kalificacion1
        action OF dB3 GIRLS 1 IS pack := TRUE
        action OF dB3 GIRLS 1 IS close := TRUE
      FIND END
    END
  INIT 0
  WHEN CHANGED
  BEGIN
    IF kalificacion1 >= 7 THEN
      mensaje OF variables := "
FELICIDADES CONTINUAS CON EL TEMA DOS"
    ELSE
      IF kalificacion1 < 7 THEN
        mensaje OF variables := "LO SIENTO NO APROBASTE EL EXAMEN, CON
TINUARAS EN EL TEMA UNO"
      IF kalificacion1 >= 7 THEN
        pen color OF valuebox 3 := 0,0,128
      ELSE
        pen color OF valuebox 3 := 255,0,0
      BEGIN
        grabar OF variables := TRUE
      END
    END
  END
  ATTRIBUTE mensal SIMPLE
  WHEN CHANGED

```

```

BEGIN
  oportunidad := oportunidad + 1
  IF kalificacion1 >= 7 THEN
    IF oportunidad = 1 THEN
      mensaje2 OF variables := "
CONTINUAS CON EL TEMA DOS"
    ELSE
      IF oportunidad = 2 THEN
        mensaje2 OF variables := "
CONTINUAS CON EL TEMA DOS"
      ELSE
        IF oportunidad = 3 THEN
          mensaje2 OF variables := "
CONTINUAS CON EL TEMA DOS"
        IF kalificacion1 < 7 THEN
          IF oportunidad = 1 THEN
            mensaje2 OF variables := "
RECUERDA QUE SOLO TE QUEDAN DOS OPORTUNIDADES MAS PARA PRESENTAR TU EXA\
MEN..."
          ELSE
            IF oportunidad = 2 THEN
              mensaje2 OF variables := "
RECUERDA QUE SOLO TE QUEDAN UNA OPORTUNIDAD MAS PARA PRESENTAR TU EXAM\
EN..."
            ELSE
              IF oportunidad = 3 THEN
                mensaje2 OF variables := " LO SIENTO PERO TE ESTOY BORRAND\
O DEL REGISTRO DEL TUTORIAL; SI QUIERES CONTINUAR TENDRAS QUE INICIAR CO\
MO PRINCIPIANTE"
              IF kalificacion1 < 7 THEN
                IF oportunidad = 3 THEN
                  borrar OF variables := TRUE
                END
              WHEN CHANGED
                BEGIN
                  IF kalificacion1 >= 7 AND oportunidad <> 3 THEN
                    CHAIN "TEMA2"
                  IF kalificacion1 < 7 AND oportunidad <> 3 THEN
                    CHAIN "TEMA1"
                  IF kalificacion1 < 7 AND oportunidad = 3 THEN
                    exit OF application := TRUE
                  END
                ATTRIBUTE encuentra SIMPLE
                WHEN CHANGED
                BEGIN
                  mensal := TRUE
                  IF kalificacion1 >= 7 THEN
                    oportunidad := 0
                    BEGIN
                      action OF dB3 GIRLS 1 IS open := TRUE
                      FIND dB3 GIRLS 1

```

WHERE variable OF dB3 GIRLS 1 <> 0  
WHEN FOUND  
action OF dB3 GIRLS 1 IS recall record := TRUE  
record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1  
access OF dB3 GIRLS 1 IS write := TRUE  
opor OF dB3 GIRLS 1 := oportunidad  
action OF dB3 GIRLS 1 IS pack := TRUE  
action OF dB3 GIRLS 1 IS close := TRUE

FIND END

END

END

INSTANCE the domain ISA domain

WITH kalificacion1 := 0

WITH oportunidad := 0

END

```
WHERE variable OF dB3 GIRLS 1 <> 0
WHEN FOUND
  action OF dB3 GIRLS 1 IS recall record := TRUE
  record OF dB3 GIRLS 1 := variable OF dB3 GIRLS 1 - 1
  access OF dB3 GIRLS 1 IS write := TRUE
  opor OF dB3 GIRLS 1 := oportunidad
  action OF dB3 GIRLS 1 IS pack := TRUE
  action OF dB3 GIRLS 1 IS close := TRUE
FIND END
END
END
```

```
INSTANCE the domain ISA domain
  WITH kalificacion1 := 0
  WITH oportunidad := 0
END
```

---

---

## BIBLIOGRAFIA.

Rich Knight.

Artificial Intelligence (1983)

Nils J. Nilssons.

Principios de Inteligencia Artificial.

Gustavo Gili.

Inteligencia Artificial, Mc-Graw Hill.

Cuena J. G, R. López.

Inteligencia Artificial y Sistemas Expertos, De Alianza Madrid (1986)

Luger, GF & Stubble Field, W A.

Artificial Intelligence and the Design of Expert Systems, Red wood City, C.A. (1989)

Heyes-Rot, Waterman, Lenat.

Building Expert Systems, De. Addison-Wesley Publishing Co. (1983)

Francisco J. Cantu, Msc.

Evaluación y Selección de Herramientas de Desarrollo de Sistemas Expertos

Centro de Investigación en Informática, ITESM, DGL (1989)

Lara F. y Gelman G.

Métodos y Modelos del Conocimiento para Sistemas Expertos.

Reporte Interno del Instituto de Ingeniería.

Dayhoff.

Neuronal Network Architecture and Introduction, N.Y. Van Nostrand Reinhold (1990).

Lara Felipe.

Redes Neuronales Artificiales, Instituto de Ingeniería UNAM (1990)

Obregón Sánchez Arturo.

Inteligencia Artificial y Sistemas Expertos, UNAM (1992)

Pressman Rogers.

Ingeniería del Software, De Mc-Graw Hill (1990)

UNAM, UNYSYS

Memorias: Las computadoras en las Instituciones de Educación y de Investigación.

Quinta y sexta Conferencia Internacional Computo Académico UNAM.

---

---

REVISTA

PCAI. Intelligence Solutions for Computer, Vol.8 No. 3 (1994)

REVISTA

AI EXPERT. The Magazine of Artificial Intelligence in Practice  
Patterns for building Intelligence Machines, pg 40-43

REVISTA

GEOMUNDO. Máquinas con Juicio e Inteligencia  
Año XVIII No. 4 México (1994)

Tour Juliust.

Pattern Recognition Principles

A. Bork.

El ordenador en la enseñanza: Análisis y Perspectiva del Futuro.

Ramón Galán.

Redes Neuronales.