

40  
2es



Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

Propuesta de Evaluación y Adaptación en  
un Sistema Tutor Inteligente (STIAYA)

T E S I S

Que para Obtener el Título de

INGENIERO EN COMPUTACION

P r e s e n t a n

Laureola Liliana Hernández Lagunes

Carmen del Pilar Monroy Ata

Director de Tesis:

DR. FRANCISCO CERVANTES PEREZ



México, D. f.

1993

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 está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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

## INDICE

### INTRODUCCION

1.1 Planteamiento del Problema	3
1.2 Objetivos	3
1.3 Organización	4

### CAPTULO I : ANTECEDENTES

1.1 Herramientas Básicas de la Inteligencia Artificial	6
1.2 Sistemas Tutoriales	18

### CAPTULO II: DISEÑO E IMPLANTACION DEL SISTEMA

II.1 Análisis y Diseño	32
II.2 Requerimientos	64

### CAPTULO III :PRUEBAS Y RESULTADOS

III.1 Condiciones Iniciales	67
III.2 Operación del Sistema	68

### CAPTULO IV : CONCLUSIONES

IV.1 Aportaciones	73
IV.2 Etapas Futuras	74

### BIBLIOGRAFIA

## INTRODUCCION

La Inteligencia Artificial (IA) surge fundamentalmente de la conjunción de dos corrientes del conocimiento de los años 30 y 40: la Lógica Matemática, en la cual los sistemas lógicos de Frege, Russell entre otros, mostraron que algunos aspectos del razonamiento podían ser formalizados en una estructura relativamente simple; y la Teoría de la Computabilidad en la que se establece que cualquier cosa que un humano pueda hacer mediante manipulación simbólica siguiendo un conjunto finito y explícito de reglas puede ser realizado por una máquina programada adecuadamente (Gödel, 1931; Kleene, 1936; Church, 1936; Turing, 1936; citados en Arbib, 1987). Además la Teoría de la Computabilidad dio las bases para la construcción de las primeras computadoras, en 1945, Mauchly y Eckert construyeron la primera computadora electrónica de propósito general (ENIAC).

Algunas otras disciplinas que contribuyeron al fortalecimiento de la IA fueron la Psicología, con la Teoría de Procesamiento y Almacenamiento de la Información (Miller, 1960), y la Cibernética que liga las ideas sobre el funcionamiento del sistema nervioso con la teoría de la información, la teoría de control, la lógica y la teoría de la computabilidad (Rosenblueth, Wiener y Bigelow, 1943; McCulloch y Pitts, 1943; Wiener, 1948; citados en Arbib, 1987).

Turing (1950), McCarthy (1956), Newell y Simon (1959) entre otros son considerados los fundadores de la IA, ya que trabajaron la posibilidad de que los mecanismos computacionales fueran programados para desplegar "conductas inteligentes".

La IA se define como una rama de la ciencia de la computación que se dedica al desarrollo de sistemas computacionales que exhiben algunos aspectos de las conductas que, cuando son realizadas por seres vivos, se considera que requieren de inteligencia. (Feigenbaum, 1981; Adelsberg, 1986; Arbib, 1987).

La IA se puede concebir desde dos enfoques complementarios: el Científico y el Ingenieril (Barr & Feigenbaum, 1981); el enfoque científico se preocupa principalmente por descubrir los principios generales que subyacen a la conducta inteligente, por lo que realiza investigación básica sobre la computadora y el cerebro para comprender los mecanismos involucrados en la realización de esas conductas; mientras que el enfoque ingenieril tiene como objetivo desarrollar sistemas computacionales que puedan ser utilizados en la construcción de autómatas complejos que desplieguen "conductas inteligentes".

Dentro del segundo enfoque, algunas áreas de aplicación son: la Ingeniería Neuronal, que se dedica al desarrollo de computadoras con arquitecturas no convencionales; la Robótica Perceptiva, la cual intenta desarrollar robots controlados por computadora con capacidad de procesar información de tipo sensorial; y la Ingeniería de Conocimiento, que aplica las técnicas de IA en el diseño de programas computacionales (sistemas expertos) que se han utilizado para llevar a cabo tareas específicas en un ambiente definido. Por ejemplo, en el área industrial, se han desarrollado sistemas para el diseño a gran escala de computadoras; en el de las comunicaciones, programas de traducción automática entre varios lenguajes a nivel semántico; y en el social, cuyas tareas se pueden dividir en dos sectores: el clínico, que realiza tareas de diagnóstico, y el educativo en donde se aplica la tecnología de la computación en el diseño de programas que sirvan como auxiliares en la educación (Tutores).

El interés del trabajo que presentamos en esta tesis, se centra en el sector educativo, específicamente en el desarrollo de los sistemas tutoriales.

El diseño de los tutores computacionales data de los principios de los años 60; su objetivo ha sido enseñar a los estudiantes a programar la computadora, utilizar el software, ayudarlos a descubrir estrategias en la solución de problemas, involucrarlos en juegos educativos diseñados para que el aprendizaje sea una experiencia agradable, así como implicarlos en la instrucción computerizada, en donde se utilizan tutores programados. En estos primeros sistemas el estudiante normalmente se involucra en una actividad en donde el aprendizaje juega un papel secundario [Barr y Feigenbaum, 1982].

Los primeros sistemas computacionales que explícitamente intentaron promover y controlar los procesos de aprendizaje fueron clasificados como CAI ("Computer Assisted Instruction" o Instrucción asistida por computadora). Estos sistemas imitan a los textos programados, en donde se le presenta al estudiante un texto con instrucciones, acompañado de preguntas que requieren respuestas cortas. Estas respuestas son evaluadas por el sistema según ciertos patrones que siguen determinadas rutas en relación a las respuestas del estudiante; esto es, cada alumno "programa" su propia trayectoria durante la revisión del material.

Al agregar las técnicas de la IA en el diseño de los CAI's se dio origen a un nuevo tipo de sistemas tutoriales, que fueron denominados como ICAI ("Intelligent Computer Assisted Instruction" o Instrucción Inteligente Asistida por Computadora). En el diseño de los ICAI se introduce un nuevo objetivo, que consiste en incluir material instruccional que permite analizar el desempeño del estudiante en la aplicación de estrategias tutoriales individualizadas. Sin embargo debe puntualizarse que los primeros ICAI se enfocaron principalmente en la eficiencia de la representación del material y se olvidaron de que, para realizar la tarea educativa, un buen maestro, debe poder evaluar objetivamente el desempeño del estudiante, así como al suyo propio.

Recientemente el término ICAI ha sido substituido por el de ITS ("Intelligent Tutoring System" o Sistema Tutorial Inteligente) para referir a los sistemas instruccionales por computadora que involucran el uso de técnicas de la IA. Los ITS's son una instancia para auxiliar al estudiante en la adquisición de destrezas y del conocimiento sobre un tema, a través de una actitud análoga a la de un maestro que lo atiende en forma personalizada [Wenger, 1987].

Los ITS se encuentran aún en pleno desarrollo y en esa medida pueden identificarse algunos errores metodológicos. Rosenberg [1987] menciona dos muy importantes: el primero es que los ITS no han sido bien fundamentados en un modelo de enseñanza-aprendizaje; y el segundo se refiere a que aún no hay un procedimiento adecuado para evaluar la eficacia y el desempeño del sistema en el auxilio de la instrucción.

## 1.1 PLANTEAMIENTO DEL PROBLEMA

Actualmente, en nuestro país uno de los principales problemas de la educación a nivel profesional y posgrado es la falta de homogeneidad en los conocimientos de los estudiantes.

Algunas veces la preparación y capacitación no está incluida en los planes de estudios. Esto provoca que el profesor tenga que dedicar gran parte del tiempo asignado para cubrir su programa en regularizar a los alumnos. Por tanto, muchas veces no llega a cubrir el programa o lo hace de una manera muy superficial.

Cuando esto sucede algunos de los estudiantes se vuelven autodidactas, pero la falta de asesoría en caso de dudas resulta desalentador lo que algunas veces propicia que estos estudiantes pierdan interés en la carrera.

Bajo las perspectivas mencionadas en los párrafos anteriores es necesario buscar caminos alternos que contribuyan a homogeneizar los conocimientos de los estudiantes.

Los sistemas auxiliares en el proceso enseñanza-aprendizaje (sistemas tutoriales) presentan una solución atractiva que evita los problemas que presentan las soluciones anteriores. Estos sistemas para cumplir con esta función deben fundamentarse en las teorías psicológicas del aprendizaje y la instrucción, de esta forma el estudiante podrá adquirir, almacenar y recuperar la información, haciendo conexiones directas y predecibles entre las variables instruccionales y el aprendizaje.

Es importante que los sistemas tutoriales puedan ser diseñados empleando herramientas de la IA, pero queremos puntualizar que esto debe llevarse a cabo contemplando variables y condiciones relacionadas directamente con los campos de la Psicología y la Educación [Hajovy y Christensen, 1987].

## 1.2 OBJETIVO.

El objetivo general de este trabajo se divide en dos etapas: primero, el diseño conceptual de un Sistema Tutorial Inteligente Adaptable y Autoevaluable (STIAYA); y segundo, diseñar e implementar la Interfaz-Estudiante para realizar la tarea educativa, tratando de resolver los problemas planteados por Rosenberg acerca de los ITS's.

Para conseguir este objetivo, nuestro diseño deberá incluir las siguientes características:

- a) Involucrar al educando como participante activo en el proceso enseñanza-aprendizaje.

- b) Incluir un sistema de evaluación del estudiante para medir su nivel de conocimiento adquirido así como su habilidad para manejarlo.
- c) Utilizar herramientas de psicología educativa para definir la estrategia instruccional más adecuada a la población de educandos que usarán el sistema, tanto para la forma de presentar el material (e.g., texto, animación), como para la evaluación del avance de los estudiantes (e.g., por objetivos).
- d) Construir un Modelo del Estudiante para poder fundamentar las decisiones sobre cómo instruir a un estudiante específico.
- e) Crear el Modelo de estudiante Estándar y el Modelo del Estudiante Ideal, para realizar una autoevaluación del Sistema en cuanto a la efectividad del proceso enseñanza- aprendizaje.

Para llevar a cabo lo anterior utilizaremos herramientas de tres disciplinas: Inteligencia Artificial, Psicología Educativa e Ingeniería de Software. La primera, cuenta con una serie de técnicas para representación de conocimiento y técnicas de búsqueda adecuadas para este tipo de sistemas; la segunda contiene las estrategias de enseñanza y adecuaciones necesarias para el desarrollo de nuestro objetivo; y la tercera provee la metodología para el análisis, diseño, implementación y pruebas del sistema a construir.

### 1.3 ORGANIZACION.

Nuestro trabajo esta dividido en esta introducción y 4 capítulos más.

En el primer capítulo se presenta una revisión bibliográfica sobre algunas de las técnicas de la Inteligencia Artificial; esto se refiere a las principales formas en que se puede representar la información así como los métodos de búsqueda que se han desarrollado. Además se presentan las partes principales que conforman un ITS y se describen algunos ejemplos de tutores que se han realizado en el área de la educación.

En el capítulo dos se presenta el análisis, diseño y requerimientos del sistema; se describe la arquitectura global de STIAYA y se explica detalladamente cómo se conforma la interfaz del estudiante. Asimismo se definen las herramientas de la Psicología educativa, el modelo instruccional que se propone y las estructuras de datos.

En el capítulo tres se presentan las pruebas que se llevarán a cabo con el sistema, las cuales se ilustran mediante ejemplos de los diferentes tipos de sesiones que se pueden realizar durante la interacción del estudiante con STIAYA a) sesión inicial, el estudiante interactúa por primera vez con STIAYA; b) sesión de instrucción, durante la cual se lleva a cabo el proceso de enseñanza-aprendizaje; c) sesión de evaluación, en donde se valora el desempeño del estudiante para ser promovido o no a el siguiente nivel y d) fin curso, en donde se explicará la autoevaluación del sistema.

Finalmente, en el capítulo cuatro evaluaremos nuestro sistema, se discutirán algunos puntos importantes de STIAYA comparándolo con otros tutores computacionales, y se plantean las posibles futuras etapas de nuestro sistema , y nuestras aportaciones.



# CAPITULO I

## ANTECEDENTES

En este capítulo nos proponemos revisar los fundamentos básicos de la inteligencia artificial, así como su aplicación en el desarrollo de los sistemas tutoriales. La presentación del capítulo comprende dos secciones principales: a) en la primera se presentan algunas de las herramientas básicas de la inteligencia artificial para la solución de problemas, diferentes formas de representación de información y los métodos de búsqueda más comunes; y b) en la segunda, se describen los módulos que conforman un sistema tutorial y se presentan algunos ejemplos de tutores realizados en el área educativa.

### 1.1 Herramientas básicas de la Inteligencia Artificial.

Actividades tales como entender un lenguaje natural, reconocer imágenes, analizar una estructura molecular, etc., se definen como que, de alguna manera, requieren de inteligencia. Algunas de las investigaciones en IA se dedican a construir sistemas de computación capaces de realizar este tipo de tareas, las cuales implican resolver problemas de tipo cualitativo más que cuantitativo, razonar más que calcular y utilizar una o varias organizaciones de conocimiento más que implementar algoritmos bien definidos.

Para resolver problemas como los mencionados anteriormente, debemos plantearlo con respecto a un marco de referencia, como por ejemplo, espacio de estados, reducción de problemas o teoremas que deben probarse. En nuestro caso, si consideramos que la tarea de instrucción (enseñanza) puede realizarse de diferentes formas y éstas a su vez se realizan por la interacción de varias subtarefas, nuestro problema de realizar un sistema tutorial inteligente puede plantearse como reducción de problemas. En éste método se hace un análisis del problema original para producir un conjunto de subproblemas y a partir de éstos generar sub-subproblemas, que sean más fáciles de resolver, y así sucesivamente hasta que el problema original se reduzca a un conjunto de problemas triviales [Nilsson, 1971]. La forma como sea representada la información constituirá el esqueleto sobre el cual se va a apoyar el método de solución (búsqueda).

Además otro de los aspectos importantes en el análisis y desarrollo de sistemas bajo el enfoque de la IA es definir cómo se estructura la información de un dominio particular (objetos, eventos etc..) para que pueda ser utilizada. Esto nos permite contar con una forma de representación del dominio de conocimiento, de manera que éste se pueda generalizar, modificar, y utilizarse aún cuando no sea totalmente exacto.

Debe aclararse que existen sistemas en los cuales el conocimiento sólo es necesario para optimizar la búsqueda, y otros en los que es usado para interpretar el medio ambiente del sistema.

Para decidir cual técnica o conjunto de técnicas utilizar es necesario, primero analizar las características del problema a resolver, para entonces tener bases suficientes y elegir el método o conjunto de métodos que mejor se adapten a la solución.

Las dos secciones siguientes serán dedicadas a presentar de una forma breve, las herramientas que se han utilizado en IA para construir eficientemente sistemas de tutoría: métodos de representación y búsqueda.

### 1.1.1. Métodos de Representación de Información.

Para representar información (ya sea de manera descriptiva o procedural) en una máquina, es necesario definir tipos de estructuras de datos y procedimientos.

En IA se han diseñado varias clases de estructuras de datos para almacenar información, así como procedimientos que permitan la manipulación "inteligente" de esas estructuras para realizar tareas complejas.

A continuación se describen algunos esquemas de representación de información.

#### 1.1.1.A LÓGICA DE PREDICADOS.

Es un sistema simbólico de razonamiento y deducción que representa información sobre el mundo por medio de sentencias. Los sistemas que forman las bases de la lógica formal en IA son: cálculo proposicional, que se utiliza para determinar si una proposición es falsa o verdadera (ver figura 1); cálculo de predicados, el cual especifica las relaciones entre las proposiciones y realiza generalizaciones (ver figura 2); y el cálculo de predicados de primer orden que utiliza funciones para establecer relaciones entre sentencias (ver figura 3).

<p>Si se tienen los siguientes enunciados:</p> <ul style="list-style-type: none"> <li>* Rocky es un perro</li> <li>* Si Rocky es un perro, entonces es un mamífero</li> </ul> <p>Si asumimos que ambos enunciados son verdaderos, y la regla de inferencias del cálculo proposicional llamada modus ponens indica que el siguiente en un - ciado también debe ser verdadero.</p> <ul style="list-style-type: none"> <li>* Rocky es un mamífero</li> </ul>
---

Figura 1. Ejemplo de una inferencia con el cálculo proposicional.

Este método proporciona un formato uniforme para la representación y utiliza un conjunto de reglas llamadas de inferencia, por medio de las cuales los hechos que son conocidos como verdaderos pueden usarse para derivar otros que también lo sean, es decir que cada aspecto nuevo que se infiera tendrá un 100% de seguridad (debe tenerse en cuenta que las inferencias pueden ser muy caras computacionalmente).

Sin embargo los problemas cuya naturaleza implica conocimiento incierto, requerirá otro tipo de representación de información.

\* Rocky es un perro

\* Todos los perros son más grandes que todos los ratones

(Hay que notar que en el 2º enunciado se incluyen gene -  
realizaciones como: todos los perros y todos los ratones;  
y una relación: son más grandes que)

Si estos dos enunciados son verdaderos, el cálculo de -  
predicados nos permite hacer la siguiente conclusión:

\* Rocky es más grande que todos los ratones

Figura 2. Ejemplo de una Generalización con el calculo de predicados.

Una función es una construcción lógica que "regresa" un  
valor; por ejemplo, la función PERTENECE A, aplicada a  
Rocky puede regresar Oli. Este conocimiento se puede  
representar así:

( PERTENECE A (Rocky Oli))

Si se usa el cálculo de predicados de primer orden, po-  
dríamos contestar la pregunta ¿A quién pertenece Rocky?

( Rocky pertenece a Oli)

Figura 3. Ejemplo de cómo se representa el conocimiento usando el calculo de predicados de primer orden.

#### 1.1.1.B REDES SEMANTICAS.

Las redes semánticas fueron introducidas por Ross Quillian (1968) en su modelo psicológico de la memoria asociativa humana. Este esquema permite describir objetos, eventos, conceptos, etc.. La información se representa como un conjunto de nodos que se conectan a otros (nodos) por medio de ligas (pueden ser de subclase, superclase o de instancia) que representan las relaciones existentes entre ellos. (Ver figura 4)

Una red semántica permite deducir información que no necesariamente se encuentra definida para cada elemento terminal, pero que esta asociada por el hecho de pertenecer a una clase determinada. Los factores importantes acerca de un objeto o concepto pueden ser inferidos de los nodos a los cuales están directamente ligados, sin necesidad de una búsqueda a través de una gran base de datos.

Conforme más completa sea una red semántica, la potencialidad del razonamiento del sistema aumenta.

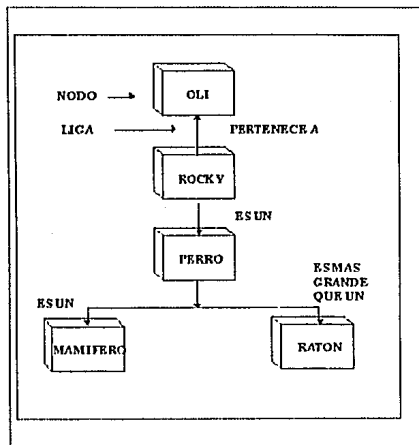


Figura 4. Red Semántica Sencilla.

De la figura podemos observar que las deducciones se pueden hacer por las propiedades de herencia. Un perro tiene las propiedades de ser y de ser más grande que un ratón, como Rocky es un perro, entonces, hereda estas propiedades.

La Lógica y las Redes Semánticas son sistemas de representación de conocimiento orientados al objeto o declarativos; porque son útiles para representar datos de los objetos así como sus relaciones con otros objetos.

Pero si se necesita realizar un sistema que pueda recomendar un curso de acción (por ejemplo un sistema experto) que contenga datos y que nos diga que hacer con esos datos, tenemos que usar otro tipo de representación de conocimiento, al que se le ha llamado sistemas de producción.

### I.I.I.C SISTEMAS DE PRODUCCION.

En IA los sistemas de producción fueron utilizados primeramente por Newell y Simon en 1972 para su modelo de cognición humana.

Un sistema de producción esta formado por tres componentes principales :

- (a) Un conjunto de reglas de producción.
- (b) Una base de conocimiento
- (c) Y Un intérprete.

Una regla de producción es un enunciado de la forma: IF condición THEN acción. La parte condicional IF o miembro izquierdo, establece las condiciones que deben presentarse para que la producción sea aplicable y la parte de la acción THEN, o miembro derecho describe la acción que debe realizarse. El lado izquierdo de cada producción representa una condición que debe estar presente en la base de conocimiento antes que la producción sea activada, ejemplo:

**Si la probabilidad de que llueva es > 50%  
Entonces lleve un paraguas**

La tarea del interprete es controlar la actividad del sistema y decidir que producción aplicar, así como operar en ciclos, los cuales tienen tres fases: "matching" (empalme), "conflict resolution" (resolución de conflictos) y "action" (acción).

En IA algunos sistemas de producción han sido utilizados para representar el conocimiento de cómo realizar tareas específicas, que impliquen el manejo de conocimiento incierto, como por ejemplo, los sistemas de diagnóstico médico.

Los métodos que hemos mencionado hasta aquí no nos permiten organizar la información en unidades complejas que representen situaciones estereotipadas u objetos, o combinar aspectos declarativos y procedurales. Para este fin se diseñaron los frames y los scripts.

### I.I.I.D FRAMES

Los frames fueron propuestos por Minsky (1975), como una base para entender la percepción visual, los diálogos de lenguaje natural y otros procesos complejos.

Estas estructuras (frames) representan la información de objetos específicos. El mecanismo de representación que hace posible esta clase de razonamiento es el de "slot" (ranura), que es el lugar donde se ubica el conocimiento dentro de un contexto creado por el frame. Cada atributo del objeto se guarda en un slot por separado.

Cada frame puede verse como una estructura de datos similar en muchos aspectos a el "registro" tradicional el cual contendrá información relevante a una entidad estereotipada.

Los slots en el frame contienen información tal como:

- a) Identificación del frame
- b) Relación del frame con otros frames. Por ejemplo el teléfono del hotel puede ser una instancia de teléfono, y esta a su vez de un dispositivo de comunicación. (Ver figura 5)
- c) Descripción de los requerimientos para igualar frames. Una silla por ejemplo tiene su asiento entre 20 y 40 cm del suelo, su respaldo de 60 cm, etc. estos requerimientos pueden usarse para determinar cuando los nuevos objetos corresponden al estereotipo definido por el frame.
- d) Información de procedimientos
- e) Información de default. Son valores que son tomados como verdaderos cuando no hay

evidencia de lo contrario.

- f) Información de nuevas instancias. Un slot se puede dejar sin especificar hasta que sea necesario definirlo para una instancia en particular.

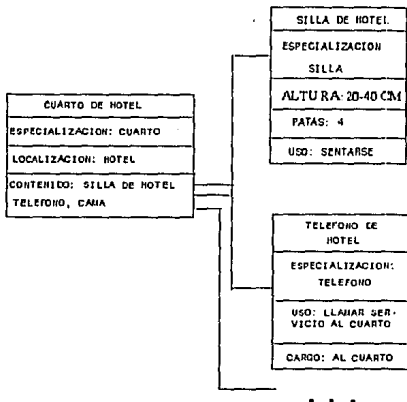


Figura 5.- Frame cuarto de hotel.

Cuando el objeto se menciona se habilitan todos los slots asociados para tener acceso a cada una de sus características.

Después de que un frame ha sido seleccionado para representar un contexto, el proceso principal en un sistema de razonamiento formado por frames, es llenar los slots con los detalles.

Cualquier nueva interpretación de la situación de los frames, puede hacer uso de los valores determinados por la experiencia, sin tener que recomputarlos. Cuando la información deba ser derivada, los procedimientos ligados proveen un medio apropiado para especificarla y se le llama heurística específica de los slots.

Los frames nos permiten acceder grandes cantidades de información a la vez, pero tienen la desventaja de ser menos modulares que otros sistemas; lo que hace que las modificaciones sean muy complicadas. De la misma forma cuando se quiere tener acceso sólo a una pieza de la información se tiene que hacer uso de todo un frame.

### 1.1.1.E SCRIPTS

Los Scripts fueron desarrollados por Schank y Abelson (1977), son estructuras compuestas por una serie de slots que describen en secuencias los eventos que se esperan en una situación familiar. Representan la información de eventos particulares.

Los componentes de un script son:

- Condiciones de entrada: condiciones que deben ser, en general, satisfechas antes de que puedan ocurrir los eventos descritos en el script.
- Resultado: Condiciones que serán, en general, verdaderas después de que hayan ocurrido los eventos descritos en el script.
- Propiedades: compartimientos que representan objetos involucrados en los eventos descritos en los scripts. La presencia de estos objetos puede inferirse aún cuando no se mencionen explícitamente.
- Roles: Compartimientos que representan personas involucradas en los eventos descritos en el script. La presencia de estas personas también puede inferirse, aún cuando no se mencionen explícitamente.
- Curso: variación específica en un patrón más general que está representado por este script particular. Rutas diferentes compartirán muchos pero no todos los componentes.
- Escenas. La secuencia real de eventos que ocurren. Los eventos están representados por la dependencia conceptual. Ejemplo script restaurant (ver Figura 6).

Existen otras estructuras importantes para la IA como la del pizarrón "blackboard", (la cual proporciona un medio para coordinar las acciones de varios expertos o fuentes de conocimiento) que fue usada en el proyecto Hearsay II (Reddy, 1976), para el entendimiento del lenguaje hablado. Estas no serán presentadas ya que se han utilizado para otro tipo de aplicaciones.

Para finalizar esta sección daremos una definición de la representación de problemas en espacio de estados, la cual es necesaria para facilitar la explicación de los métodos de búsqueda. Cabe mencionar que la diferencia con los métodos de búsqueda para reducción de problemas es que se considera que cada nodo representa un subproblema.

La representación en espacio de estados, define un estado como una configuración particular del problema. Se tiene un estado inicial, un operador que transforma un estado en otro y un estado meta. En términos de estados y operadores la solución de un problema es la secuencia de operadores que transforman el estado inicial en el estado meta; los sucesores de un nodo son calculados al aplicar los operadores asociados con el nodo  $_i(n)$ .

### 1.1.2. Métodos de Búsqueda.

Como se mencionó anteriormente, cuando tenemos un espacio de búsqueda, y se intenta encontrar una trayectoria adecuada, dentro de ese espacio, que nos lleve de las condiciones iniciales del problema a una situación de solución del mismo, podemos intentar una búsqueda ciega en donde se corre el riesgo de procesar todos los nodos del espacio de posibilidades (búsqueda exhaustiva) o emplear información inherente al problema (heuristic) que nos permita guiar el recorrido del espacio de búsqueda de manera que encontremos una solución al problema bajo estudio, si es que existe, sin visitar todos los

elementos del espacio de posibilidades.

En el caso de una búsqueda ciega existen varios métodos, de los cuales revisaremos dos métodos básicos que son muy representativos: el de "breadth-first" (anchura primero) y el de "depth-first" (profundidad primero).

SCRIPT: RESTAURANT	ESCENA: ENTRAR
CURSO: CAFETERIA	CI ENTRA AL RESTAURANT
FOKS, MESAS	CI DECIDE SENTARSE
MENU	CI SE DIRIGE A LA MESA
COMIDA	CI SE SIENTA
DINERO	
ROLES:	ESCENA 3: ORDENAR
M MESSERO	.
COCCOSINERO	.
CA CAIBERO	.
E DUBHO	
CI CLIBTE	ESCENA 3: COMER
-----	.
CONDICIONES DE ENTRADA:	.
CI ESTA HAMBRIENTO	.
CI TIENE DINERO	
RESULTADOS:	ESCENA 4: SALIR
CI TIENE MENOS DINERO	.
E TIENE MAS DINERO	.
CI NO TIENE HAMBRE	.

Figura 6. Script entrar al restaurant

En el método "depth-first", se procesan los nodos más recientemente generados (figura 7a). El árbol de búsqueda empieza a crecer en forma vertical por una de sus ramas hasta un cierto límite predefinido de profundidad. Esto puede provocar por una parte que si la solución está en los niveles más próximos del nodo raíz pero en las últimas ramas a considerar, será necesario generar demasiados nodos antes de identificar la trayectoria de solución del problema (figura 7b), o que, cuando no se conoce con exactitud el nivel de profundidad que garantiza una solución, la solución no se encuentre ya que ésta se localiza en los niveles que no fueron generados debido al límite de profundidad (figura 7c). Este método termina cuando encuentra la primera solución.



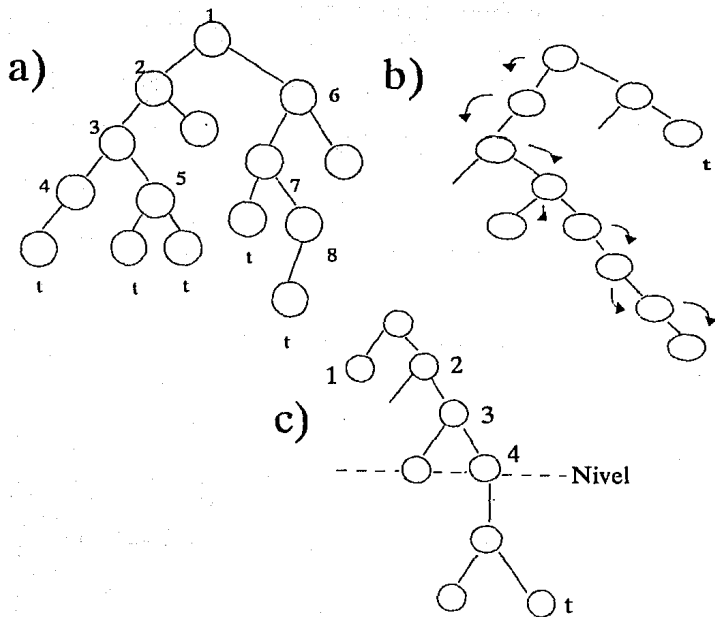


Figura 7. a) Ejemplo de cómo crece el árbol por una de sus ramas. b) La búsqueda se vuelve exhaustiva, ya que la solución se encuentra en las últimas ramas a considerar. c) La solución no se encuentra, ya que ésta se presenta en los niveles que no se generaron.

### 1.1.2.B BREADTH FIRST

En el método "breadth first", el árbol de búsqueda se va generando en forma horizontal; esto es, se analizan todos los nodos de un nivel del árbol antes que cualquier otro nodo del siguiente nivel. Con este procedimiento se garantiza que se encuentran soluciones que requieren un menor número de pasos.

La desventaja de este método es que se requiere una gran cantidad de memoria, ya que el número de nodos se incrementa exponencialmente de acuerdo al nivel y al grado de expansión de los nodos, y todos éstos deben almacenarse. Ver figura 8.

**Nota:** En las figuras 7 y 8 los números indican el orden en cual fueron generados. La línea fuerte indica la trayectoria de solución, t = nodo terminal y 0 = nodos sin solución.

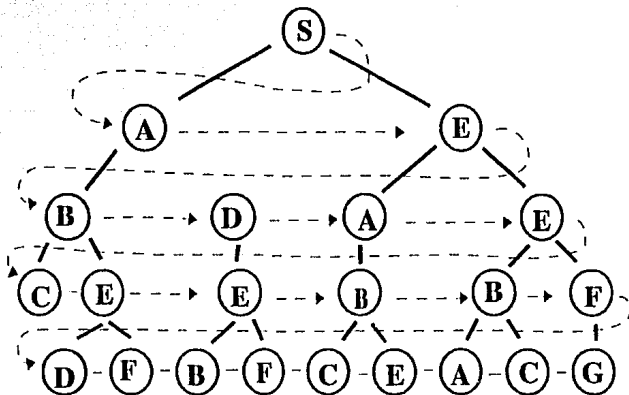


Figura 8. Ejemplo de cómo se expanden los nodos en el método "breadth-first"

Si consideramos las desventajas de los métodos anteriores, y se tiene un problema complejo donde el espacio de búsqueda es muy grande, una búsqueda ciega no es eficiente, ya que se podrían visitar demasiados nodos antes de encontrar una solución. Por ello es necesario encontrar alternativas más eficientes tomando en cuenta que en la práctica las computadoras tienen limitaciones de almacenamiento y tiempo de proceso.

Para encontrar una solución sin el riesgo de realizar una búsqueda exhaustiva, se han propuesto métodos que utilizan información particular del problema bajo estudio para guiar la búsqueda de una solución.

En ocasiones, para resolver un problema se requiere una buena solución en términos de algún costo o beneficio, lo cual puede ser representado a través de una función de evaluación que asigne un número para medir el esfuerzo necesario para alcanzar la solución.

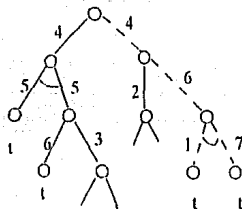
### 1.1.2.C COSTO UNIFORME

El objetivo de usar una función de evaluación, es encontrar la mejor solución, ya sea para el camino más corto, el que tenga menos movimientos, o el de "menor costo" para definir una secuencia de eventos que nos resuelva el problema. La elección de la función de evaluación determina críticamente los resultados de la búsqueda. Si una función de evaluación ignora las características reales de un nodo para su expansión puede resultar en soluciones no óptimas. De la misma manera el uso de la función de

demasiados nodos para obtener la solución mínima.

Si consideramos que  $f(n)$  es la función de evaluación y  $g(n)$  representa el costo correspondiente a la ruta desde el estado inicial hasta el nodo  $n$ , dada por ejemplo, por la suma de los costos individuales de todos los operadores empleados para tal recorrido tenemos que (figura 9):

$$f(n) = g(n)$$



SOLUCION A  
Suma Costo = 20  
Máximo Costo = 15

SOLUCION B  
Suma Costo = 18  
Máximo Costo = 17

Figura 9. Cálculo del costo máximo.

Además si en la función de evaluación incluimos, una estimación de la cercanía que tiene el nodo con el nodo meta del problema, este esquema puede servirnos para reducir el esfuerzo de búsqueda. Si la estimación es buena, puede demostrarse que un algoritmo basado en ella siempre dará la solución óptima. La función de evaluación está formada por:

$$f(n) = g(n) + h(n)$$

$g(n)$  representa el costo correspondiente a la ruta desde el estado inicial hasta el nodo  $n$ .

$h(n)$  denota la estimación (definida por criterios heurísticos) del costo que tomaría llegar desde el nodo  $n$  hasta el estado meta.

Definidas  $f, g$ , y  $h$  el criterio que se debe satisfacer para asegurar que la solución dada por el método (ya sea  $A^*$  o  $AO^*$ , ver Nilsson, 1971 caps 3 y 5) es que  $h(n)$  no sea mayor que el costo real de ir de  $n$  a un estado meta. (Ver figura 10)

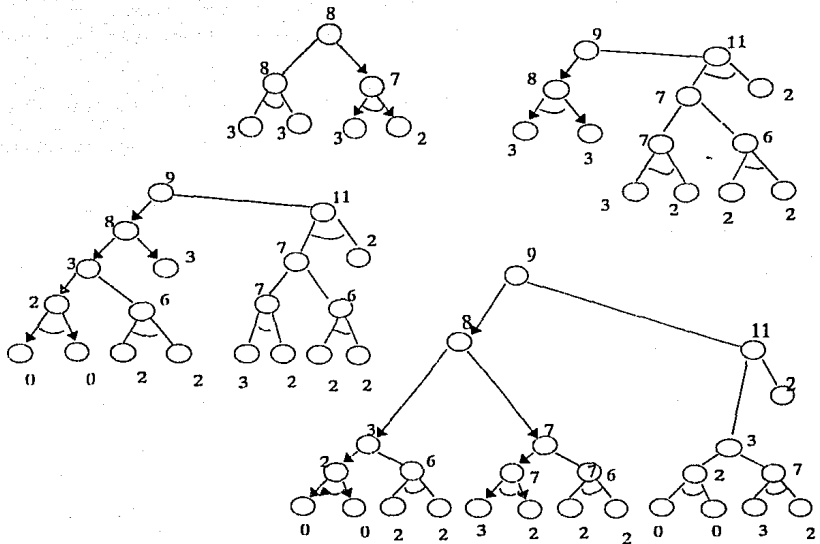


Figura 10. Expansión de los nodos usando una función de estimación  
 Nota: los números corresponden a los valores de  $h^*$  para la suma de costos

Cada uno de los métodos mencionados tienen implícito un razonamiento para guiarse. El razonamiento se produce en tres formas básicas: a) "forward" o hacia adelante, se parte desde un estado inicial para encontrar una trayectoria que nos conduzca hacia algún estado meta; b) "backward" o hacia atrás, se parte de un estado meta para encontrar una trayectoria que nos conduzca a algún estado inicial y c) se tiene una combinación de ambos y se le llama bidireccional, ya que simultáneamente las búsquedas inician del estado meta y del estado inicial, hasta que se encuentre un nodo común.

nician del estado meta y del estado inicial, hasta que se encuentre un nodo común.

Existe otro método que a veces se guía hacia adelante y otras hacia atrás según convenga y que es llamado "means ends analysis". Con esta estrategia se pueden resolver los principales problemas y "regresar" a resolver los más pequeños.

#### 1.1.2.D MEANS-ENDS-ANALYSIS

Means-Ends-Analysis es un proceso que se centra en la detección de las diferencias entre el estado actual y el estado meta. Una vez ubicada la diferencia, se debe encontrar un operador que reduzca tal diferencia. Pero el operador tal vez no se pueda aplicar al estado actual, entonces, habrá que ubicarse en un subproblema hasta llegar a un estado, en el cual se pueda aplicar el factor que reduzca la diferencia entre estado actual y estado meta (ver Figura 11). Sin embargo, el operador puede no producir los resultados esperados. Entonces el segundo subproblema es regresar al estado actual (del cual partimos). Por otra parte, si la diferencia se eligió correctamente y el operador es efectivo y reduce la diferencia entre estado actual y estado meta, entonces los dos subproblemas deben ser más fáciles de resolver que el original. El método Means-Ends-Analysis se puede aplicar recursivamente. De manera que se enfocan los grandes problemas primero y a las diferencias se les pueden aplicar prioridades. Las de mayor prioridad se pueden atender antes que las otras.

Cada uno de estos métodos cuenta con un algoritmo para su implementación, para mayor detalle puede verse Nilsson (1971): capítulos 3 y 5 y Rich (1983): capítulo 3.

## 1.2 Sistemas Tutoriales.

En esta sección mencionaremos las partes principales que forman un sistema tutorial, y presentaremos algunos ejemplos de tutores realizados dentro del área educativa. Asimismo, señalaremos algunas características que distinguen a los nuevos sistemas tutoriales, llamados ITS ("Intelligent Tutoring System").

Las partes con que cuenta un sistema tutorial son:

#### a) Módulo de Experiencia.

Es el conocimiento de un tópico específico (incluye conocimiento de manera descriptiva y procedimental), que el sistema trata de impartir al estudiante.

Por ejemplo la representación con redes semánticas y scripts jerárquicos han sido útiles para generar y responder preguntas que implican una relación causal (e.g. SCHOLAR; WHY). Otros sistemas han empleado representaciones del dominio de conocimiento por medio de reglas de producción para construir representaciones de habilidades y métodos de solución de problemas (e.g. WUMPUS; GUIDON).

**b) Modelo del Estudiante.**

Indica qué es lo que conoce y no conoce el estudiante. Trata de modelar al estudiante para poder hacer hipótesis de los errores y malos entendidos, para que se le pueda indicar porque está mal, sugerirle correcciones, etc. Se han utilizado varias técnicas de IA para modelar el conocimiento del estudiante ya como: reconocimiento de patrones, banderas en las redes semánticas o en las reglas que el estudiante ya ha dominado. Por lo general se tiene un modelo de "Overlay", el cual consiste en comparar el desempeño del estudiante con el del experto, y así poder decidir si el estudiante ha dominado o no una habilidad y/o conocimiento; se representa como un subconjunto del conocimiento del experto. O un modelo "bug" para representar los errores comunes o los procesos erróneos de los estudiantes. También se guarda otro tipo de información como es, su nivel de habilidad, cómo le gusta interactuar con el sistema, lo que olvida, etc. es decir toda aquella información que nos indique si el estudiante aprendió y qué método es el más efectiva.

**c) Módulo Tutorial.**

Especifica cómo el sistema presenta el material al estudiante (cómo se enseña). Monitorea el desempeño del alumno, le proporciona ayuda, lo evalúa, etc. Algunos métodos de enseñanza típicos son: a) por diagnóstico, cuando trata de indagar el entendimiento del estudiante, dándole tareas y evaluando su respuesta, b) el coaching, para mejorar o aprender habilidades en la solución de problemas, al indicarle al alumno la validez de sus acciones, que está haciendo mal, que no está usando, etc. c) el socrático, el cual es una serie de cuestionamientos para el estudiante con el fin de hacerlo razonar, etc.

**d) Interface.**

Una interface amigable, la cual permita la comunicación estudiante-sistema, ya sea por medio de un intérprete de lenguaje natural, menús, etc.

No todos estos componentes son completamente desarrollados en los sistemas tutoriales debido a su complejidad y tamaño. La mayoría tiende a concentrarse en una o algunas de las partes que lo integran. (Ver Figura 12)

Diferencias	Procedimientos				
	U S A A V I O N	U S A T R E N	U S A C A R R E O	U S A T A H I	C A M I N O
Más de 1000 millas	x				
Entre 100 y 1000		x	x		
Entre 1 y 100			x	x	
Menos de 1 milla					x
	E S T A R E N A V I O N	E S T A R E N T R E N	E S T A R E N C A R R E O	E S T A R E N T A H I	E S T A R E N C A M I N O

Requisitos

Figura 11. La tabla de procedimientos-diferencias es una forma de relacionar procedimientos y diferencias. Ejemplifica las alternativas de transporte que están relacionadas con las distancias posibles y los requisitos necesarios.

### 1.2.1 Ejemplos de Sistemas Tutoriales

Los sistemas tutoriales serán presentados en orden cronológico apartir de los años 70. Estos han sido agrupados por décadas para tratar de resaltar sus principales características y/o aportaciones, así como el uso que tuvieron de herramientas educativas y/o cognitivas.

SCHOLAR [Carbonell, 1970] es un sistema tutorial que enseña a los estudiantes la geografía de América del sur.

El conocimiento se representa en una red semántica cuyos nodos contienen conceptos y objetos geográficos, los cuales están organizados en una jerarquía parcial con relaciones tales como superpartes, superconceptos y superatributos, de manera que puedan realizarse inferencias utilizando la herencia de propiedades en la red semántica (Figura 13).

Inicialmente, SCHOLAR crea un modelo de estudiante ideal utilizando su red semántica completa y progresivamente transforma este modelo (eliminando y modificando algunos nodos), para reflejar el desempeño actual del estudiante. En cuanto a diagnóstico, el proceso se limita a marcar nodos de acuerdo a las declaraciones del estudiante.

Las estrategias tutoriales se ocupan principalmente de la selección de tópicos, misma que puede ser hecha por el sistema o por el estudiante. En cualquier caso se determina la información más relevante que debe presentar, por lo que se utilizan etiquetas en cada nodo y en sus diversos atributos, las cuales sirven como guía en la selección, aunque también se ayuda aleatoriamente. Para registrar los tópicos cubiertos y las estrategias empleadas, se utiliza un mecanismo de agenda.

SCHOLAR es un sistema tutorial de iniciativa mezclada; tanto el sistema como el estudiante pueden iniciar la conversación haciendo preguntas, mediante una interface de lenguaje natural, las cuales están limitadas a sentencias simples. En el caso del tutor, el texto se genera en "plantillas" de sentencias y preguntas que se llenan con información de la red. El análisis de las preguntas del estudiante sigue estos mismos principios pero a la inversa, mientras que el análisis de sus respuestas consiste en comparar palabras clave contra una lista dinámica generada por la red para cada pregunta. Esto quiere decir que SCHOLAR no entiende respuestas escritas en forma incorrecta y, por tanto, no puede diagnosticar esa información.

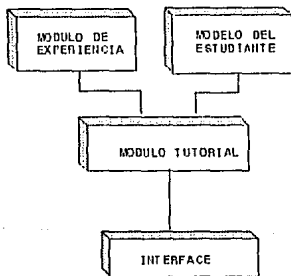


Figura 12. Arquitectura de un ITS. (Fischetti, 1990)



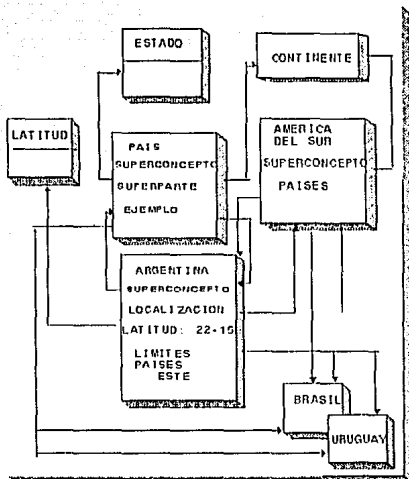


Figura 13. Arquitectura de Scholar.

SOPHIE [Brown y Burton (1976)] está formado por dos programas: SOPHIE-I y SOPHIE-II (ver Figura 14). Crea un medio de aprendizaje en donde el estudiante tiene la oportunidad de aplicar su conocimiento teórico de las leyes de la electrónica, recibir retroalimentación y adquirir estrategias en la reparación de equipo eléctrico.

SOPHIE-I, simula un laboratorio de electrónica en el cual el estudiante debe diagnosticar las posibles fallas en el equipo, llevando a cabo una serie de medidas (voltaje, corriente, etc.), las cuales le permitirán probar sus hipótesis relacionadas con la localización y naturaleza de las fallas. La tarea de instrucción se dedica a responder las preguntas hipotéticas del alumno sobre la consecuencia de una acción o evento, evaluar las nuevas medidas propuestas por el estudiante, criticar las hipótesis propuestas por el alumno tomando en cuenta toda la información que éste haya derivado de las medidas, para que en el caso necesario le sugiera hipótesis alternativas. Para responder a las preguntas del estudiante y criticar sus hipótesis, SOPHIE-I utiliza una interfase de lenguaje natural.

La principal aportación de SOPHIE es que representa el dominio de conocimiento con una simulación del modelo matemático del circuito, en el cual aleatoriamente insertará la falla de uno de los componentes. Su experiencia pedagógica es inofensiva. En este sistema tutorial no se cuenta con un modelo del estudiante, sin embargo el alumno recibe retroalimentación cuando se realiza la validación lógica de

las soluciones que ha propuesto.

SOPHIE-II es una extensión de SOPHIE-I, incluye un juego localizador de averías para 2 equipos de estudiantes y un experto articulado. El experto articulado que utiliza SOPHIE II, localiza las faltas del estudiante y articula las deducciones que conducirán a el estudiante a descubrirlas. Su falla principal es que no puede hacer un seguimiento de los errores de los estudiantes.

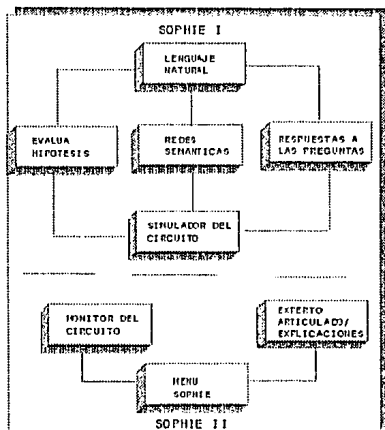


Figura 14. Arquitectura Global de Sophie

WEST [Brown y Burton, 1976]. Es un sistema basado en el juego "Cómo se conquistó el Oeste", su objetivo es ejercitar las habilidades aritméticas. En cada turno el jugador cuenta con tres números aleatorios, con los cuales debe hacer una expresión aritmética que involucre dos operadores diferentes. El resultado de esta expresión aritmética determina el número de espacios que debe moverse el jugador. Sin embargo, el juego está diseñado para que las expresiones aritméticas sean estratégicas, más que de un gran valor numérico.

Burton y Brown, proponen un paradigma general para el método de "coaching", el cual ellos llaman "resultados y ejemplos". El propósito de este paradigma es realizar las intervenciones de ("coaching") entrenamiento relevante y memorizable (ver Figura 15). El conocimiento se describe como un conjunto de resultados, los cuales se presentan al estudiante, de acuerdo a su relevancia en el

desarrollo del juego, mediante ejemplos concretos pertinentes a su actual movimiento.

WEST utiliza un modelo de estudiante de tipo diferencial, en donde el desempeño del estudiante es analizado y comparado con la actitud del experto en las mismas circunstancias de juego. La naturaleza aleatoria del juego no permite predecir el tipo de jugada que será necesaria. Por tanto, el dominio del jugador no puede determinarse simplemente por el registro de sus jugadas, pero sí mediante comparación con las decisiones del experto en las mismas jugadas.

Cuando es el turno del jugador, el experto genera una lista de todos los movimientos posibles. Si la expresión del jugador no es la óptima, se inicia el proceso de diagnóstico. Primero se analiza el movimiento del estudiante con el "identificador de resultados" para saber que movimiento fue utilizado. Después, con el mismo identificador, se analizan todos los movimientos del experto que sean mejores que el movimiento del estudiante. Finalmente los resultados se evalúan en el modelo diferencial del estudiante hasta encontrar algún resultado con el cual el estudiante se da cuenta de que está mal.

Pedagógicamente WEST adopta una actitud cautelosa e interviene solo cuando existe evidencia de deficiencia. Se pretende utilizar la relevancia cuando es necesario para motivar la memorización.

<p>Principio 1: Antes de dar , asegurarse de que realmente el resultado del alumno está mal.</p> <p>Principio 4: Si el estudiante va por mal camino, asesorarlo solo en los movimientos que lo alejen del error.</p> <p>Principio 5: No asesorarlo en dos movimientos consecutivos.</p> <p>Principio 9: El experto siempre tiene que jugar en forma óptima</p> <p>Principio 10 : Si el estudiante pide ayuda, dársela en niveles diferentes de indicación:</p> <p>(1) Aislar el problema y ubicarlo.</p> <p>(2) Delimitar el espacio de los posibles movimientos.</p> <p>(3) Seleccionar el movimiento óptimo y decir por qué.</p> <p>(4) Describir cómo hacer un movimiento óptimo.</p>
--

Figura 15. Ejemplos de principios de "coaching".

WHY [Stevens y Collin, 1977]. Fue creado para explicar el proceso pluvial. El conocimiento se almacena en scripts jerárquicos (lo cual da un panorama organizado del conocimiento) que representan una secuencia estereotípica de eventos. Esas secuencias sirven para determinar las relaciones temporales y causales entre eventos.

WHY no crea un modelo global del estudiante, su capacidad de diagnóstico se limita a la evaluación individual de respuestas. Las fallas se representan como errores, del estudiante, en las relaciones del dominio de conocimiento.

WHY emplea el método socrático (ver figura 16) de instrucción, en donde el tutor no enseña mediante una exposición directa, pero guía al estudiante con preguntas sucesivas hasta formular principios generales, basados en casos individuales, para examinar la validez de sus propias hipótesis, descubrir sus contradicciones y finalmente extraer las inferencias correctas a partir de factores que conozcan.

La generación de preguntas y la revisión de respuestas se basa en los scripts.

Se tiene un sistema de reglas de producción en donde la condición es la última respuesta dada por el estudiante y la acción es lo que el tutor propondrá a continuación. Algo importante en estas reglas es que se pueden organizar categóricamente (Taxonomía de técnicas socráticas).

WHY utiliza una interface de lenguaje natural para interactuar con el usuario.

<p><b>REGLA 6:</b> Forma una regla general para un factor insuficiente</p> <p><b>IF</b> El estudiante da una explicación de uno o más factores insuficientes</p> <p><b>THEN</b> formula una regla general asumiendo que el factor dado es suficiente Y pregunta al estudiante si la regla es correcta</p> <p><b>RAZON DE USO</b> Forzar al estudiante a poner atención en otros factores causales.</p> <p><b>EJEMPLO</b> Si el estudiante dice que el agua es la razón por la cual crece el arroz en China, preguntarle ¿Crees que en cualquier lugar donde haya agua disponible crezca el arroz?</p>
---

Figura 16. Ejemplo de una regla del método socrático.

**WUSOR-II** [Goldstein y Carr, 1977]. Es un "coach" (asesor) para el juego de computadora **WUMPUS** (Yob, 1975). El objetivo de **WUSOR-II** es ayudar a los jugadores a desarrollar las habilidades de razonamiento lógico y probabilístico al advertirles sobre la conveniencia o no de cualquier movimiento que realicen. Utiliza el método de "coaching" (asesorar al jugador) como estrategia de instrucción.

El conocimiento que utiliza **WUSOR-II** esta representado por: a) reglas que contienen el conocimiento de las habilidades que se necesitan al jugar; b) reglas de evidencia, para modelar los criterios usados al hacer hipótesis sobre que habilidades del experto el jugador posee y; c) reglas de explicación para las estrategias tutoriales, usadas cuando se seleccionan los tópicos que serán discutidos.

**WUSOR-II** está formado por cuatro módulos (ver figura 17): el experto, el psicólogo, el modelo del estudiante y el tutor.

El módulo experto usa las reglas de habilidad para sugerir y analizar movimientos. El resultado de su análisis (se lo comunica al psicólogo), es una gama de posibles movimientos con una lista que asocia las habilidades que se podrían necesitar para hacer cada movimiento.

El psicólogo (también llamado módulo de diagnóstico) formula hipótesis relacionadas con el dominio de habilidades que son conocidas por el estudiante. Construye y actualiza el modelo del estudiante.

El del estudiante indica el nivel de juego que puede esperarse de un jugador - principiante con el conocimiento incompleto de las reglas básicas del juego, - novicio, si hay un entendimiento de las habilidades lógicas, - amateur con conocimiento de lógica y las habilidades más elementales de probabilidad. Se utiliza un modelo "overlay", (el cual es su principal aportación) del estudiante en donde el estado del conocimiento del estudiante es visto como un subconjunto del conocimiento del experto.

Tutor usa el modelo del estudiante para guiar las interacciones con el jugador. Básicamente elige discutir habilidades todavía no exhibidas por el jugador en situaciones donde su uso daría mejores movimientos. También utiliza las reglas de explicación para seleccionar el tópico apropiado que se discutirá con el jugador y así podrá escoger la forma de explicación.

Algunas limitaciones que se encuentran son: a) como el conocimiento del estudiante se considera subconjunto del conocimiento del experto, el modelo de "overlay" fallará si el dominio permite múltiples soluciones al problema, ya que si el estudiante sigue una solución para la cual el experto no esta

programado, posiblemente el estudiante haga buenos movimientos que el experto puede considerar como no óptimos, b) los errores en el conocimiento de procedimientos son más difíciles de detectar, ya que el modelo de "overlay" asume que una conducta no óptima es causada sólo por el no dominio de una habilidad individual, y que las distorsiones en una habilidad correcta no ocurren.

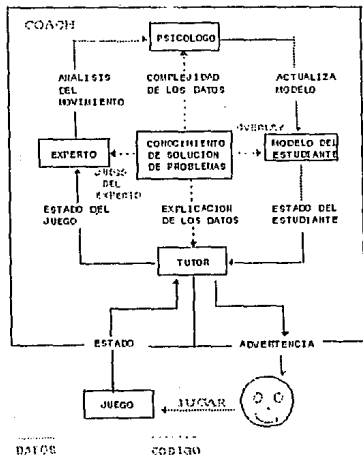


Figura 17. Arquitectura de Wusor.

**BUGGY** [Brown y Burton, 1978]. Es un programa que puede determinar los errores de los estudiantes en sus habilidades aritméticas al considerar que la mayoría de las veces tienen dificultades en la solución de problemas debido a que siguen procedimientos erróneos.

Buggy representa una habilidad como una colección de subhabilidades (por ejemplo en la suma, una subhabilidad es la de llevar un dígito a la columna siguiente).

El modelo construido por Buggy incorpora los subprocedimientos correctos e incorrectos, para simular la conducta del estudiante en un problema particular.

El modelo ha sido usado con dos fines: a) identificar los errores de los estudiantes y b) entrenar a los maestros en el diagnóstico de las causas de los errores de los estudiantes, presentándoles ejemplos

de conductas sistemáticas incorrectas.

La idea central de este trabajo es construir un modelo de diagnóstico con una red de procedimientos, en donde cada subhabilidad, ya sea correcta o incorrecta, se representa explícitamente como un subprocedimiento de la red. Este esquema de representación, facilita una descomposición apropiada de una habilidad en subhabilidades con el fin de poder hacer explícita la estructura de control, resaltar las habilidades y ejecutar directamente el modelo de diagnóstico resultante.

Más que ser un subconjunto o simplificaciones de las reglas de los expertos, (como un modelo overlay), la red de procedimientos es un modelo de la conducta del estudiante que se estructura en función de la desviación del procedimiento correcto.

**GUIDON** [Clancey, 1979]. Enseña a estudiantes de medicina la relevancia de los datos clínicos y de laboratorio, en el diagnóstico y tratamiento de infecciones bacteriológicas.

Al iniciar una sesión Guidon usa a MYCIN (sistema experto) para resolver el caso que será presentado al estudiante. Las conclusiones que MYCIN comunica a GUIDON se configuran en un árbol AND/OR de metas y reglas. Este árbol contendrá toda la información relevante al caso, lo que permitirá que el módulo tutorial tenga a la mano toda la información que el experto ha usado para tratar este caso, y cómo ha usado este conocimiento para su razonamiento.

El módulo tutorial GUIDON es un sistema basado en reglas (aprox 200) y está completamente separado de MYCIN. Para adaptar el diálogo a las necesidades específicas de esa sesión de enseñanza, el sistema utiliza un modelo de estudiante, el cual es formado como un "overlay" de las reglas del dominio.

La estrategia pedagógica adoptada por GUIDON es un diálogo de iniciativa mezclada.

Las reglas tutoriales se usan para construir el modelo del estudiante, y para seleccionar el procedimiento de discurso adecuado para responder a las preguntas de los estudiantes. Las reglas de diagnóstico de enfermedades infecciosas constituyen las habilidades que se le van a enseñar al estudiante.

Los procedimientos de discurso formalizan cómo el programa debe comportarse en términos generales, y no en términos de los datos o resultados de un caso particular. Un procedimiento de discurso es una secuencia de acciones que debemos seguir bajo determinadas condiciones debido a la complejidad del material, al entendimiento del estudiante acerca del material y las metas tutoriales para la sesión.

La importancia de GUIDON radica en la separación de la enseñanza y del dominio de conocimiento, ya que permite que la base de datos MYCIN de enfermedades infecciosas, pueda ser reemplazada por reglas de diagnóstico para problemas de otro dominio.

Una debilidad de este sistema es que intenta explicar la conducta del estudiante solamente en términos de las reglas de MYCIN.

Apartir de los años 80 se toma más en cuenta el aspecto pedagógico y cognitivo en los ITS, por ejemplo:

**INTEGRATION** [Kimball, 1981]. Es un sistema para resolver integrales. El experto del dominio está representado como una matriz que relaciona toda clase de problemas con todos los métodos de solución. Cada elemento de la matriz es un valor que indica la probabilidad de aplicar un método a un problema dado, generando un subproblema de otra clase.

Dentro de esta representación probabilística, el sistema puede tener una caracterización explícita del objetivo de enseñanza en la forma de matriz representativa de experto.

El estado de conocimiento del estudiante se representa en una matriz similar cuyos valores pueden compararse contra los del experto. En base a los resultados de la comparación, se generan nuevos problemas. De la misma forma que las estrategias, el tutor puede dar asesoría, de acuerdo a su probabilidad de solución. Sin embargo, también acepta sugerencias del estudiante.

Con respecto al diagnóstico, el sistema considera las probabilidades en la matriz del estudiante con métodos de la Teoría probabilística. Kimball, sugiere que la probabilidad permite precisar una medida de aprendizaje relevante del alumno, revelando sus discontinuidades, las cuales aparentemente están relacionadas con sus fallas.

INTEGRATION se comunica con el usuario utilizando una interface de lenguaje simple que básicamente consiste de preguntas de opción múltiple. Esto es posible porque el tutor mantiene el control completo de la interacción.

Una característica única de INTEGRATION es que el dominio experto puede improvisarse durante el curso de las sesiones. Si la aproximación del estudiante es mejor solución que la trazada en el archivo del experto, el sistema adopta la solución del estudiante como estándar.

LISP [Anderson, J. & Reiser, B., 1984-1985]. Es un tutor para enseñar a los estudiantes a programar con Lisp. LISP utiliza reglas tanto para representar (modelo ideal) el conocimiento que queremos que el estudiante adquiera (reglas correctas) así como reglas "buggy" que representan variaciones del modelo ideal.

Cuando el tutor se percata que el estudiante tiene dificultad al codificar, el tutor trabaja el algoritmo con el alumno, paso por paso usando un ejemplo. Después que el algoritmo es construido, el estudiante puede volver a codificar, presumiblemente, con una mejor idea de lo que se debe hacer para que el código trabaje adecuadamente.

El tutor sigue el código del estudiante, símbolo por símbolo y trata de ver si se puede relacionar con alguna de las reglas. Si la regla es correcta, el tutor no interviene, pero si por el contrario, se encuentra que hay un error, el tutor interrumpe para advertirlo.

El tutor le proporciona un editor para que pueda meter código. Este editor automáticamente balancea los paréntesis y pone el lugar de los argumentos de cada función. El objetivo de esto es quitarle a el estudiante la carga cognitiva requerida para chequear la sintaxis de bajo nivel y permitir así que el estudiante se enfoque en los problemas de codificación de alto nivel. Cuando un error surge el sistema da retroalimentación inmediata, es decir, cuando el alumno comete un error, el tutor responde con un mensaje de diagnóstico apropiado y construye una explicación en ingles basada en patrones asociados con cada regla. El tutor LISP tiene aproximadamente 325 reglas para leer y escribir programas en Lisp, y 475 versiones erróneas de esas reglas. Cada lección hace uso de un conjunto diferente de reglas.

El curriculum está organizado en diferentes lecciones y cada lección toma de 1 a 4 horas para realizarlas. Cada tópico consta de un pequeño folleto instruccional y de muchos problemas para practicar las habilidades que se deben enseñar en cada lección.

Después de un episodio instruccional, el tutor interrumpe a el estudiante para ver si el estudiante entiende como escribir el código. Si la respuesta del estudiante demuestra falta de conocimiento, el tutor inicia un dialogo para guiar a el estudiante a la solución correcta.

Interactúa con el usuario por medio de ventanas: a) para introducir el código, b) para desplegar la retroalimentación del tutor y c) para recordar las metas, así como diferentes tipos de letras para distinguir la entrada de código del tutor y la del estudiante (normal, bold, etc).

Este sistema a veces es lento y utiliza muchas más, se implementa en una VAX-11/725 con 3 megabytes de memoria para cada estudiante, requiriendo un poder computacional considerable. Aún en el futuro con una implementación más eficiente se estima un mínimo de 1 megabyte de memoria por alumno.

SMITHOWN [Shute and Glaser, 1986], es un sistema que enseña los principios básicos de la Economía. Combina un micromundo (el cual permite que el estudiante tenga un aprendizaje por descubrimiento), con un modulo de enseñanza. Este tutor se caracteriza porque esta organizado no en función de sus componentes funcionales como serían el modulo de experiencia, de diagnóstico, etc., más bien está organizado alrededor de sus elementos pedagógicos llamados "bites". Cada bite es un sistema en miniatura enfocado a piezas específicas de la materia en cuestión, las cuales incluyen: a) conocimiento de las relaciones conceptuales con otros "bites" (parte/clase/subclase, etc), b) relaciones curriculares con otros "bites", tales como pre o postrequisitos, c) dominio del estudiante de la materia (un modelo de estudiante en miniatura, que junto con otros "bites" forman un tipo de overlay, d) la habilidad para diagnosticar el uso o no uso del conocimiento, e) la habilidad de generar problemas que impliquen el uso de conocimiento, y f) la habilidad de generar intervenciones instruccionales relacionadas a su

conocimiento.

El sistema utiliza una simulación, la cual permite que el estudiante experimente al cambiar parámetros (como son la oferta, precios de ciertos artículos, etc.) y así estudiar las modificaciones inducidas en otros parámetros. El sistema también puede ser directivo con el estudiante cuando se da cuenta que éste tiene dificultades y le sugiere experimentos de acuerdo a el currículum. El Currículum debe descomponerse en diferentes lecciones, los cuales deben ser ligados en una red.

Ahora en los años 90 la tendencia es desarrollar herramientas (shell) para temas con características similares, así como sistemas en donde sus componentes funcionales estén distribuidos a través de la representación del currículum, utilizando la programación orientada al objeto, como ejemplo tenemos:

**TDITS** [Huang Lianjing, 1990], es una herramienta para el desarrollo de sistemas tutoriales inteligentes. Esta formada por cuatro componentes. El primero es para la adquisición de conocimiento, es el que permite al experto o a los maestros, (los cuales no necesariamente son programadores) interactuar directamente con la computadora para construir la base de conocimiento del dominio particular. La información debe tener características como: nombre de la unidad, preceptos necesarios a la unidad que se va a enseñar, descripciones y soluciones de los ejemplos, posibles errores, etc.

El segundo es el dominio de conocimiento que usa una estructura lógica o funcional para representar el conocimiento conceptual y el conocimiento de las habilidades que se van a enseñar (por el tutor).

El tercero es el módulo pedagógico, el cual se divide en dos componentes: a) el planeador, desarrolla un plan de enseñanza adaptado al estudiante que va a ser instruido. El Plan adopta una estructura de red no jerárquica, tiene un nodo inicial (condiciones iniciales), un nodo final (meta de enseñanza) y nodos TO, los cuales son operaciones de enseñanza, e incluyen precondiciones, contenidos de enseñanza, etc.; b) Ejecutor, usa el plan de enseñanza para guiar al estudiante a través del curso.

Y el cuarto es el modelo del estudiante, el cual usa un "overlay" para representar el estado del entendimiento del estudiante acerca de la materia que se va a enseñar.

El modelo del estudiante esta formado:

{ Estudiantes

(Estudiante 1 (nombre,nivel,estilo,errores,historia,(C,S))

(Estudiante n (nombre,nivel,estilo,errores,historia,(C,S)))

C representa el entendimiento del estudiante del conocimiento conceptual y S ilustra el uso de las habilidades.

**FAE** [A. Gisolfi, G. Moccaldi, 1990]. Es un tutor para la factorización de expresiones algebraicas que utiliza Smalltalk/V, el cual es un lenguaje orientado a objetos. En este caso los objetos son pantallas, en donde se muestra el conocimiento del dominio en sus diferentes formas. Para los ejercicios del dominio se emplean listas circulares cuyo puntero se almacena en una variable global que sirve para que en sesiones posteriores no se repitan los ejercicios.

El modelo del estudiante es de tipo diferencial, cuyo mecanismo emplea reglas correctas e incorrectas. Existe un mecanismo generativo que construye un árbol de solución del problema aplicando los dos tipos de reglas, el conjunto de posibles soluciones se compara contra la solución propuesta por el alumno y de esta forma se determina si el estudiante se equivoca o no. En la evaluación se considera tanto el número de reglas aplicadas como su dificultad; cada regla tiene un peso, la suma de pesos se



multiplica por el coeficiente de dificultad, este último relativo a la forma como se presenta el ejercicio.

El alumno es quien activa Smalltalk, de esta forma tiene acceso a las diferentes opciones del sistema.

ELINT [Julia Vassileva, 1990]. Es un sistema para enseñar la solución de problemas en ingeniería eléctrica, aunque se plantea libre de contenido. (Figura 18)

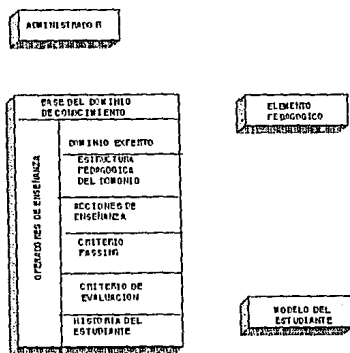


Figura 18. Componentes principales de ELint.

En este tutor la base de conocimiento incluye: el dominio experto, una estructura pedagógica concerniente al dominio para poder hacer comparaciones entre el alumno y el experto en las mismas circunstancias, acciones de enseñanza, un conjunto de operadores de enseñanza que indiquen qué acciones deben aplicarse a situaciones específicas y cómo deben transformar tal situación, criterios "passing" que determinen si los operadores de enseñanza son efectivos, un criterio de evaluación, y la historia del estudiante en términos del dominio. Con esto último se pretende hacer un modelado del alumno tipo overlay.

El modelo del estudiante como tal, debe incluir solo la representación de ciertas características individuales del estudiante independientes del dominio. Para inicializar los valores del modelo es

necesario realizar un examen previo que determine las habilidades del estudiante.

El elemento pedagógico es el encargado de planear el camino óptimo de enseñanza para cada estudiante de acuerdo a sus características y a una serie de heurísticas referentes a los operadores de enseñanza, los cuales indican: dificultad, nivel de abstracción, etc.

El administrador es el elemento que permite capturar los diferentes dominios así como adecuar las heurísticas. También coordina las funciones de los demás componentes.

Con base en esta revisión de los tutores hemos observado que de los dos problemas planteados por Rosenberg, solo uno de ellos se ha tomado en cuenta y es el que se refiere a que los sistemas estén bien fundamentados en un modelo de enseñanza-aprendizaje. Sin embargo aún falta considerar el que se refiere a la autoevaluación del sistema para conocer su efectividad en el proceso de enseñanza-aprendizaje. Este aspecto representa una característica indispensable para los ITS's, pues con la autoevaluación podrán adaptarse más efectivamente a las necesidades de los estudiante y mejorar sus estrategias instruccionales.

## CAPITULO II

### DISEÑO E IMPLANTACION DEL SISTEMA

En este capítulo utilizamos la metodología de Ingeniería de Software para realizar la etapa de análisis de STIAyA y la etapa de diseño del módulo que interactúa con los estudiantes (interfaz del Estudiante). En la etapa de análisis se contempla el objetivo del sistema así como sus alcances, funciones y restricciones. En lo que se refiere al diseño, se hace una descripción detallada de los procedimientos involucrados en el modelo de instrucción que utilizaremos y de las especificaciones de cada uno de los componentes de la Interfaz del Estudiante.

#### II.1 ANALISIS Y DISEÑO.

La finalidad de realizar el análisis de un sistema es obtener una descripción detallada de los procesos, flujos de datos y estructuras de datos involucrados para proceder a diseñar el software del sistema.

En esta tesis se pretende analizar y diseñar un sistema tutor inteligente cuyo principal objetivo es convertirse en una herramienta complementaria en la labor docente del profesor. Para ello se tienen contempladas interfaces para tres tipos de usuario: un analista, el profesor y los estudiantes. Sin embargo, dado que el problema planteado en la introducción se refiere a la tarea educativa, solo implantaremos y probaremos la interfaz del estudiante, que es donde se lleva a cabo el proceso de instrucción.

Para comenzar el análisis del sistema que proponemos describiremos las necesidades y procesos que intervienen en un ciclo de instrucción. A partir de este punto podremos hacer mención de todos los elementos que, a nuestro juicio, contribuyen a que un sistema tutor cumpla satisfactoriamente su auxilio en la labor de enseñanza.

Antes de iniciar un curso se debe realizar un examen diagnóstico para conocer el nivel de conocimiento de los alumnos, de esta forma se tendrán bases para desarrollar un plan de enseñanza específico para ese grupo.

Una vez definido el plan de enseñanza y los tópicos que formarán la base de conocimiento del curso, éstos se deben organizar en un índice temático sin olvidar que algunos temas se consideran antecedentes de otros. Además, se tienen que definir los objetivos que deben lograrse al finalizar cada uno de los temas, los cuales normalmente están formados por lecciones, mismas que deben exponerse en forma clara, sencilla y didáctica.

La enseñanza de cada lección debe ser planada y contar con elementos de apoyo que contribuyan a que el estudiante no sólo adquiera el conocimiento que se está impartiendo, sino que también aprenda cómo utilizarlo en situaciones prácticas. Los elementos referidos pueden ser: ejemplos, figuras, diagramas, dibujos, gráficas, etc. Estos elementos se pueden utilizar para que la información se presente en forma didáctica de manera que se facilite su comprensión, especialmente cuando se trate de ideas abstractas. Asimismo, también se pueden utilizar caracteres especiales, textos alargados, subrayados, color, etc., para diferenciar, resaltar y aclarar información importante, siempre y cuando estén referidos a atributos relevantes y del tópico por aprender. Además, se debe contar con herramientas de apoyo tales como un diccionario y bibliografía, con las cuales los alumnos puedan reforzar los conceptos, consultar información antecedente o bien profundizar y avanzar más rápido en algún tema en particular.

Finalmente, al concluir cada tema se debe contar con suficientes elementos de evaluación como exámenes, tareas y preguntas durante la sesión, con los cuales se pueda valorar el desempeño del

Es importante señalar que en general los ciclos de instrucción cuentan con dos tipos de información: a) la primera puede considerarse estática, como el dominio de conocimiento de un tópico específico, por el hecho de que no sufre cambios a menos que se encuentre algún error o surjan nuevos descubrimientos sobre la materia de estudio; b) la segunda cambia de manera dinámica, debido a que se modifica constantemente y solo es válida el mismo lapso de tiempo que la duración del curso, como es el caso de la información acerca del desempeño de un alumno o un grupo de alumnos. Al primer tipo de información en adelante la referiremos como "memoria a largo plazo" (MLP), mientras que a la segunda la referiremos como "memoria a corto plazo" (MCP). En estos dos tipos de memoria estará estructurada la información de nuestro tutor.

El diseño de STIAyA será planteado libre de contenido, de forma tal que cualquier materia que nazca conocimiento descriptivo y de procedimientos pueda integrarse.

A continuación se describen las herramientas de la psicología educativa incluidas en nuestra propuesta de solución para cubrir las necesidades de un ciclo de instrucción. Cabe aclarar que estas herramientas han sido seleccionadas por ser prácticas para lograr nuestro objetivo y relativamente fáciles de implantar.

Previo a la interacción de un grupo de estudiantes con STIAyA, el profesor a cargo debe realizarles una evaluación que refleje el grado de conocimiento que poseen y la habilidad que tienen para manejarlo, así se podrá definir en forma particular las condiciones iniciales del modelo de cada uno de los estudiantes. En forma general la integración de los modelos de cada estudiante nos permite definir las condiciones iniciales de un modelo estándar que represente las características y necesidades del grupo de educandos. A este tipo de evaluación la hemos denominado evaluación diagnóstica.

Además de la evaluación diagnóstica de los estudiantes, se deben articular y estructurar los elementos del índice temático, esto es, establecer dependencias y representar las relaciones existentes entre los elementos. De esta forma tendremos una organización adecuada del conocimiento a transmitir y podremos identificar propiedades que dan significado al contenido, e.g., claridad, comprensión, generalización, implicación, etc. Cuanto más solidamente esté articulado el contenido, mayores serán las posibilidades de asimilación y retención. Para lograr esto, utilizaremos un Mapa de Contenido (Figura 1), con el cual se determinarán las secuencias óptimas en la presentación del material y en la consulta de antecedentes. Este mapa será generado con la técnica de Morganov-Heredia que consiste en elaborar una gráfica y una tabla de doble entrada en donde se representa la dependencia entre el material de los distintos temas [Huerta, 1982].

Una vez establecidos la evaluación diagnóstica, el índice temático (dividido en lecciones) y el mapa de contenido, estamos en condiciones de generar un plan de enseñanza que gule adecuadamente el proceso de instrucción y satisfaga las necesidades de los educandos. En este plan de enseñanza se debe especificar el orden de presentación, elementos de instrucción y forma de presentar el material; características que nos ayudarán a lograr los objetivos de aprendizaje.

El planteamiento de los objetivos de aprendizaje permite: a) proporcionar al alumno la descripción de la meta por alcanzar; b) precisar la información que se quiere transmitir; c) planear más fácilmente las experiencias de aprendizaje; y d) facilitar la elaboración de los instrumentos de evaluación para determinar la eficacia del proceso enseñanza-aprendizaje. Estos objetivos se especifican al iniciar cada tema intentando, de esta manera involucrar activamente al estudiante en el proceso de enseñanza. En STIAyA los objetivos de aprendizaje se especificarán, clasificarán y evaluarán de acuerdo a la taxonomía de los objetivos educativos de B. S. Bloom [CISE, 1981], quien considera seis niveles del dominio cognoscitivo:

T/T	0101	0102	0103	0104	...	0n0n
0101	0	1	1	1		1
0102	0	0	1	0		1
0103	0	0	0	1		1
0104	0	0		1		1
...						
0n0n	0	0	0	0		0

Figura 1. Ejemplo de Mapa de contenido. Las columnas representan dependencia entre los temas.

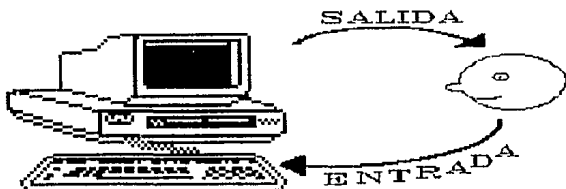
- 1) Conocimiento.- comprende aquellas conductas que consisten en aprender de memoria. El estudiante repite la información de la misma manera en que se le presenta.
- 2) Comprensión.- es el entendimiento del mensaje de una comunicación. El alumno, debe reelaborar el mensaje o identificar la misma información que se presentó enunciada de manera diferente.
- 3) Aplicación.- consiste en la transferencia del conocimiento a una situación nueva o casi nueva para el estudiante.
- 4) Análisis.- se refiere a la identificación de las partes de la comunicación o de su estructura.
- 5) Síntesis.- incluye aquellas conductas en que el alumno combina varios elementos para lograr un producto original. La expresión de ideas y experiencias propias dará como resultado un producto distinto para cada alumno.
- 6) Evaluación.- consiste en juzgar si una comunicación determinada ha satisfecho o no un criterio específico. También es la comparación de dos contenidos con un propósito determinado, mediante el proceso de razonamiento.

Los primeros cuatro niveles se refieren a la formación de conceptos y entendimiento, mientras que los dos últimos se refieren a la creatividad del educando. En esta tesis, los objetivos se definirán considerando únicamente los niveles de conocimiento, comprensión, análisis y aplicación. Los niveles de síntesis y evaluación no serán incluidos debido a que para ello

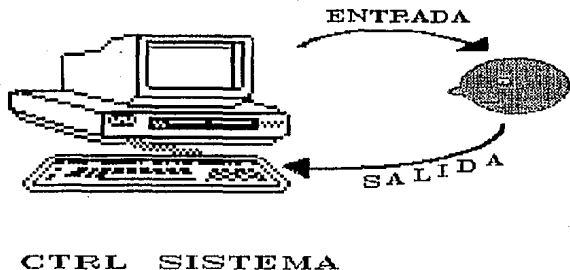
consideramos necesario un sistema completo de entendimiento de lenguaje natural. En cuanto a los cuatro niveles que consideraremos es indispensable establecer en cada lección el porcentaje deseable, para mayor entendimiento del tópico, de cada nivel cognoscitivo.

Para promover la participación de los educandos en el proceso de enseñanza-aprendizaje, además de proporcionarles los objetivos que deben alcanzar, en STIAyA se podrán distinguir dos tipos de interacción con los estudiantes: 1) cuando el sistema guíe la instrucción y 2) cuando el estudiante dirija sus actividades de aprendizaje. Debe señalarse que el sistema le cede el control del proceso al estudiante para que éste regule su acceso al material que se está consultando de acuerdo a sus necesidades y capacidad de aprendizaje, lo cual de ninguna manera implica dar al estudiante el control absoluto sobre la selección de los contenidos del dominio de conocimiento ni sobre la definición de las evaluaciones.

En el primer caso, STIAyA tendrá el control de la interacción, ya que determinará las actividades de aprendizaje al realizar el proceso de instrucción; emitirá señales a través de la pantalla a las cuales el estudiante contestará mediante el teclado. En esta forma de interacción el alumno será como una "caja negra" para STIAyA ya que éste sólo contará con respuestas cerradas a partir de las cuales deberá realizar inferencias para determinar tanto las necesidades y habilidades del estudiante como el camino más promisorio en el proceso de instrucción. En el segundo caso es el estudiante quien tendrá el control de la interacción, esto es, él mismo determinará sus actividades de aprendizaje. En esta forma de interacción el alumno verá a STIAyA como una "caja negra" a la cual le indicará sus necesidades de aprendizaje esperando una respuesta concreta a cada una de ellas, sin importarle como se lleven a cabo los procesos para que esto suceda. (ver Figura 2)



**CTRL ALUMNO**



*Figura 2. Interacción STIAyA - usuario*

No obstante, aunque el alumno tenga el control de la interacción, STIAyA supervisará las actividades de aprendizaje propuestas. El sistema le puede ceder el control al estudiante sobre la estrategia de instrucción y sobre la manipulación del contenido. En lo que se refiere a la estrategia de instrucción el control puede ser en el orden de presentación del contenido o en la secuenciación de las actividades de aprendizaje, las cuales pueden ser:

- \* Consultar ejemplos
- \* Resolver ejercicios
- \* Consultar un tema diferente al que se está revisando
- \* Consultar bibliografía
- \* Consultar el diccionario
- \* Realizar un examen

En la manipulación del contenido, el control se ejerce sobre las relaciones y procedimientos del dominio, esto implica una manipulación del contenido orientada hacia una meta u objetivo. Para este efecto son necesarias simulaciones que permitan que el educando experimente.

La instrucción se debe adaptar a las características y necesidades del estudiante para que éste pueda avanzar a su propio ritmo, contemplando en todo momento el hacer una presentación apropiada de los materiales. En general, durante el proceso de instrucción se crearán listas dinámicas, con información del mapa de contenido, de las rutas alternas a seguir en el índice temático. Para elegir el siguiente tópico, si se trata de una sola lección "siguiente" es ésta la lección que se enseñará. Si hay lecciones paralelas, se hace una diferencia entre la relación de habilidades necesarias para cada lección y las habilidades que hasta ese momento tiene el estudiante. De estas diferencias se elige la menor porque consideramos que será más fácil de asimilar para el alumno.

Una vez definida la siguiente lección, los conceptos y procedimientos que la constituyen se presentan en secuencia como lo indique el plan de enseñanza, no así sus elementos de apoyo didáctico, cuya secuencia de presentación debe adecuarse para que realmente afirme el conocimiento. Para evaluar y discernir acerca del mejor camino a seguir para un usuario en particular, se utilizarán reglas heurísticas que involucran características tanto del contenido temático como del desempeño del estudiante;

**Avance Tema** .- Es la ubicación del alumno respecto a un tema o lección.

**Tipodulmo**.- Es la clasificación del alumno de acuerdo al porcentaje promedio del dominio de sus habilidades en los cuatro niveles del dominio cognoscitivo: conocimiento, comprensión, análisis y aplicación.

**Recurso Visto** .- Es último elemento que revisó el estudiante: concepto, ejemplo, ejercicio, etc.

**H1Obj11A1** .- Es la relación entre la habilidad de conocimiento del alumno y la habilidad de conocimiento requerido para la lección

**H2Obj12A1** .- Es la relación entre la habilidad de comprensión del alumno y la habilidad de comprensión requerida para la lección.

**H3Obj13A1** .- Es la relación entre la habilidad de análisis del alumno y la habilidad de análisis requerida para la lección.

**H4Obj14A1** .- Es la relación entre la habilidad de aplicación del alumno y la habilidad de aplicación requerida para el sistema.

**RecDisp1**.- Es la relación entre los ejemplos necesarios para cada concepto o procedimiento y los ejemplos que ha revisado el alumno.

**RecDisp2**.- Es la relación entre los ejercicios necesarios para cada concepto o procedimiento y los ejercicios que ha revisado el alumno.

**RecDisp3**.- Es la relación entre las analogías necesarias para cada concepto o procedimiento y las analogías que ha revisado el alumno.

**RecDisp4**.- Es la relación entre las redes necesarias para cada concepto o procedimiento y las redes que ha revisado el alumno.

**Importancia del tema** .- Es un indicador para realizar preguntas acerca del material que se está revisando.

El sistema debe informar al usuario de su nivel inicial de conocimiento, los requerimientos de la secuencia de instrucción y sus necesidades personales. Además debe evaluar su ejecución, así como ofrecerle indicaciones del por qué sus respuestas fueron o no correctas. Tomando en cuenta todo lo anterior, son necesarias tres diferentes evaluaciones: a) la primera que represente el estado de conocimiento del estudiante, así como sus necesidades, antes de iniciar su interacción con el sistema; b) la segunda, que represente el desempeño del estudiante mediante la colección de datos durante la instrucción; y c) la tercera, la cual se realizará al final de un tema y al finalizar el curso, que certifique la competencia o no del alumno. Los exámenes son una parte importante en esta última evaluación. Para crearlos se utilizará una matriz de conductas en la que se especifica el número de reactivos necesarios para evaluar, en cada tema las cuatro habilidades del dominio cognoscitivo (Ver figura 3).



T/T	0101	0102	0103	0104	...	0n0n
0101	0	1	1	1		1
0102	0	0	1	0		1
0103	0	0	0	1		1
0104	0	0		1		1
...						
0n0n	0	0	0	0		0

*Figura 3. Matriz de conductos. Cada celda representa el número de reactivos necesarios para evaluar cada habilidad en cada tema*

El análisis de estas tres evaluaciones será la base para asesorar al estudiante; darle recomendaciones y retroalimentación, con base en la evaluación, para que mejore su desempeño.

Al finalizar la instrucción de un grupo, para realizar una autoevaluación del sistema en cuanto a la efectividad del proceso enseñanza-aprendizaje, se contempla llevar a cabo un análisis comparativo entre un modelo del desempeño ideal del estudiante y un modelo de estudiante estándar, el cual se debe elaborar con base en los modelos de estudiante de la población que utilice el sistema. La comparación de estos modelos permitirá obtener parámetros que ayuden a mejorar el plan de enseñanza del sistema.

A continuación presentamos la descripción funcional de STUyA. Se inicia con la estructura de los procesos necesarios para el funcionamiento del sistema y después con la descripción detallada de cada uno de ellos.

**ESTRUCTURA DE PROCESOS DE STIAYA [Fig.4]****1.0 Interfaz de Mantenimiento [Fig.5]**

- 1.1. Control de Mantenimiento
- 1.2. Insertar
- 1.3. Borrar
- 1.4. Modificar
- 1.5. Efectuar Transacción
- 1.6. Imprimir Transacción
- 1.7. Consultar Transacción

**2.0 Interfaz del Profesor [Fig.6]**

- 2.1. Alta de Alumnos
- 2.2. Baja de Alumnos
- 2.3. Consulta Modelo del Estudiante
- 2.4. Simulador
- 2.5. Captura Modificaciones

**3.0 Interfaz del estudiante [Fig.7]**

- 3.1. Tutor [Fig.8]
  - 3.1.1. Monitor de Control
  - 3.1.2. Ubicación del área de trabajo
  - 3.1.3. ESCAPE
  - 3.1.4. Presenta Lecciones
- 3.2. Diagnóstico [Fig.8]
  - 3.2.1. Como presentar el material
  - 3.2.2. Valida Peticiones
  - 3.2.3. Evaluación del estudiante
  - 3.2.4. Actualiza Modelo Estudar
- 3.2.5. Evaluación de STIAYA

**4.0 Utileras****DESCRIPCION DE LA FIGURA 4.****PROCESO: 1.0 Interfaz de Mantenimiento**

**DESCRIPCION:** Creación y mantenimiento del dominio de conocimiento (altas, bajas, modificaciones y transacciones). La creación se refiere a la captura de la información que compone el cuerpo del conocimiento a transmitir. El mantenimiento se debe realizar

cuando haya material sobre nuevos descubrimientos en el tema, inconsistencias o errores detectados durante el uso del sistema, ya sea por parte del propio analista, del profesor o de los estudiantes. Utiliza el correo interno del sistema para comunicarse con los demás usuarios.

**ENTRADAS:** Clave del usuario, modelo ideal del estudiante, dominio de conocimiento, mensajes, transacciones.

**SALIDAS:** Dominio de conocimiento, modelo ideal, aviso mensajes y/o transacciones, mensajes.

**RESUMEN LOGICO:** Identificación de usuario  
 Aviso de mensajes  
 Aviso de Transacciones  
 Repeat  
 If acceso correo then  
   Call utilerfas(corr.o)  
 Elige archivo de la MLP  
 If crea MLP o Modifica x mensajes then  
   Elige alta, baja o modifica  
   else (transacciones)  
 until salida

**PROCESO:** 2.0 Interfaz del profesor

**DESCRIPCION:** Supervisión de la información que se transmite al estudiante, por medio de una simulación, en la cual toma el papel de alumno; en caso de que encuentre alguna anomalía en esta información, puede realizar correcciones y/o modificaciones para el mejoramiento del sistema utilizando mensajes o especificando las transacciones para algunos archivos de la MLP. Asimismo podrá revisar el desempeño de los alumnos, hacer sugerencias y obtener reportes estadísticos del grupo.

**ENTRADAS:** Clave del usuario, mensajes, modelo del estudiante, resultados de estadística, dominio de conocimiento, modificaciones al dominio.

**SALIDAS:** Aviso mensajes, mensajes, modelo del estudiante, perición estadística, estadísticas de grupo, dominio de conocimiento, transacciones.

**RESUMEN LOGICO:** Identificación de usuario  
 Aviso de mensajes  
 Repeat  
 If acceso correo then  
   Call utilerfas(correo)

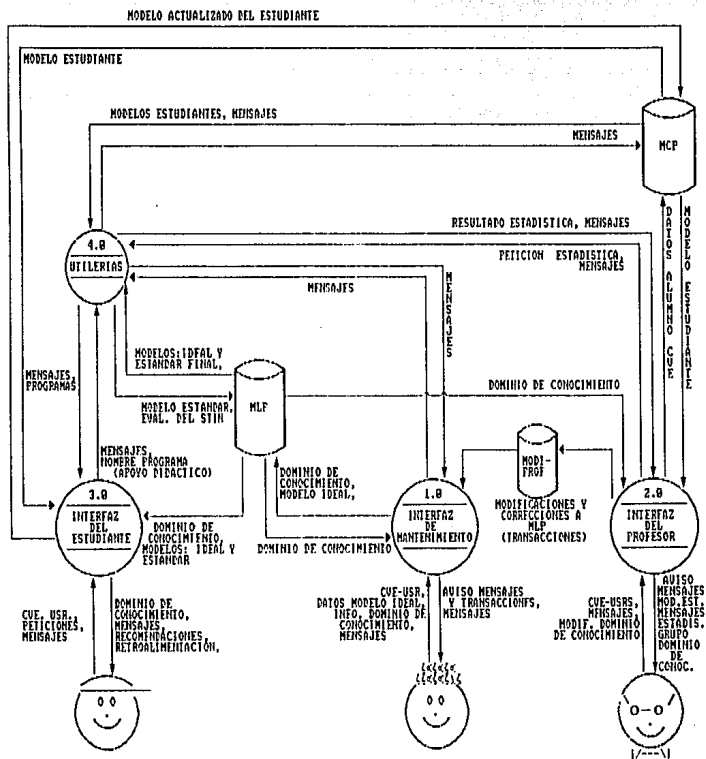


FIG. 4 D.F.D. GENERAL

```

If alumnos then
  Elige alta, baja o consulta
If simulador then
  consulta dominio de conocimiento
If modificaciones then
  elige transacciones o mensajes
If estadísticas
  Call utilerfas(estadística)
Until fin sesión

```

**PROCESO: 3.0 Interfaz del estudiante**

**DESCRIPCIÓN:** Permite al estudiante interactuar con el sistema; aquí se lleva a cabo el auxilio del proceso enseñanza-aprendizaje a través de una instrucción individualizada, en el sentido de que se va adaptando a las características y necesidades del estudiante para que éste pueda avanzar a su propio ritmo.

**ENTRADAS:** Clave del usuario, dominio de conocimiento, modelo ideal, modelo estándar, modelo del estudiante, peticiones alumno, mensajes.

**SALIDAS:** Dominio de conocimiento, modelo actualizado del estudiante, retroalimentación, recomendaciones, y modelo estándar actualizado.

**RESUMEN LÓGICO:** Identificación de usuario  
 Ubicación del ambiente de trabajo  
 Repeat Monitor de control  
 If CTRL sistema then  
 Presenta dominio de conocimiento  
 else CTRL alumno  
 Elige y valida petición  
 If fin tema  
 begin  
 Evaluación estudiante  
 Actualiza modelo estudiante  
 end  
 If todo el grupo fin tema then  
 Actualiza modelo estándar  
 Until Fin sesión

**PROCESO: 4.0 Utilerfas**

**DESCRIPCIÓN:** Programas de propósito general que contribuyen al mejor funcionamiento de STIAYA.

**ENTRADAS:** Mensajes, nombre de programa(petición de algún programa), modelos de estudiante, modelo estándar, modelo ideal.

**SALIDAS:** Mensajes, Resultados estadísticos, modelo estándar actualizado, resultado de evaluación de STIAYA.

**RESUMEN LOGICO:** Lee petición (nombre) de programa  
Ejecuta programa

## DESCRIPCION DE LA FIGURA 5.

**PROCESO:** 1.1. Control de mantenimiento

**DESCRIPCION:** Monitor de acceso y control a las funciones de mantenimiento, da aviso de la existencia de mensajes y/o transacciones.

**ENTRADAS:** Clave Usuario, mensajes, transacciones.

**SALIDAS:** Aviso transacciones, Aviso mensajes, Nombre archivo MLP, No. transacción, clave usuario, mensajes.

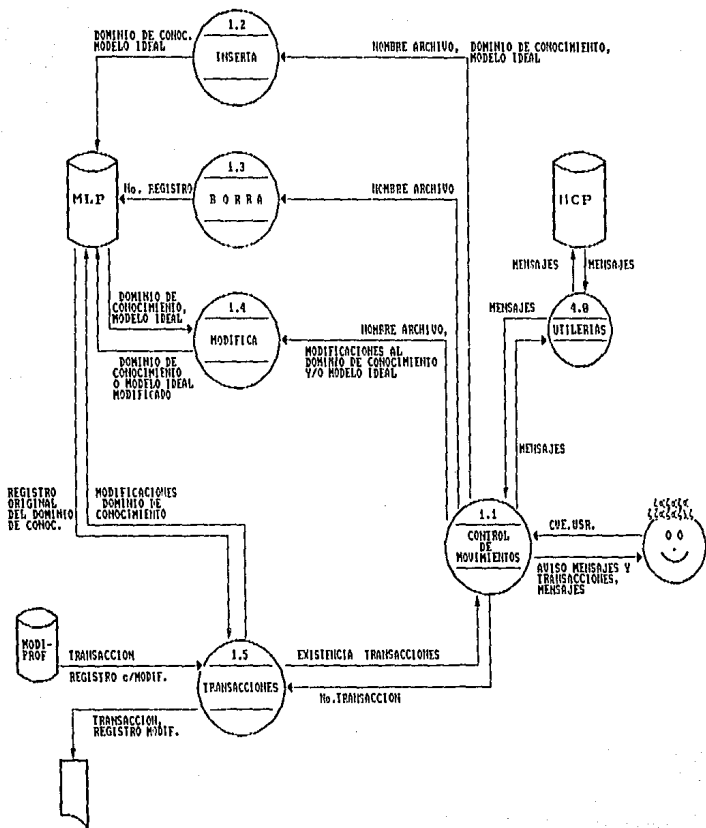
**RESUMEN LOGICO:** Identificación de usuario  
 If existe mensaje then  
   Aviso de mensajes  
 Lectura al archivo de transacciones  
 If existe transacción then  
   Aviso de transacciones  
 Repeat  
   Elige archivo de la MLP  
   if correo then  
     Call utiferlas  
   else  
     begin  
       Elige inserta, borra,  
       Modifica o transacción  
       If inserta then Call inserta  
       If borra then Call borra  
       If modifica then Call modifica  
       If transacción then Call transaccion  
     end  
   Until fin sesión.

**PROCESO:** 1.2. Inserta

**DESCRIPCION:** Captura la información de cada uno de los archivos de la MLP y los parámetros para Modelo Ideal del Estudiante.

**ENTRADAS:** Nombre Archivo MLP, dominio de conocimiento, Modelo Ideal

**SALIDAS:** Dominio de conocimiento, Modelo Ideal.



**RESUMEN LOGICO:** Lee nombre de archivo de la MLP  
 Abre archivo  
 Repeat  
 Lee datos del registro del archivo  
 If datos OK then  
 Graba registro  
 Until fin captura  
 Cierra archivo.

**PROCESO: 1.3. Borra**

**DESCRIPCION:** Muestra y elimina los registros con información no deseada de cada uno de los archivos de la MLP.

**ENTRADAS:** Nombre Archivo MLP

**SALIDAS:** Registro de MLP

**RESUMEN LOGICO:** Lee nombre de archivo de la MLP  
 Abre archivo  
 Repeat  
 Lee No. registro del archivo  
 Muestra datos del registro  
 if elimina registro OK then  
 borra registro  
 Until fin borra  
 Cierra archivo.

**PROCESO: 1.4. Modifica**

**DESCRIPCION:** Muestra y captura la información que se desea modificar en los registros de cada uno de los archivos de la MLP.

**ENTRADAS:** Nombre Archivo MLP, dominio de conocimiento, Registro modificado.

**SALIDAS:** Dominio de conocimiento modificado.

**RESUMEN LOGICO:** Lee nombre de archivo de la MLP  
 Abre archivo  
 Repeat  
 Lee No. registro del archivo  
 Muestra datos actuales del registro  
 Lee nuevos datos  
 If datos OK then  
 Graba nuevo registro  
 Until fin modifica  
 Cierra archivo.



**PROCESO: 1.5. Transacción**

**DESCRIPCION:** Permite la consulta de las modificaciones del profesor en algún registro de la MLP (transacción), facilita la impresión de las modificaciones del profesor en algún registro de la MLP, realiza el cambio de contenido en registros específicos de alguno de los archivos de la MLP por el contenido de un registro similar de la MODI-PROF, de acuerdo a las indicaciones del profesor en cada una de las transacciones.

**ENTRADAS:** No.Transacción, transacción, registro con modificación, registro original

**SALIDAS:** Modificaciones al dominio de conocimiento, Transacción, registro e/modificación.

**RESUMEN LOGICO:** Abre archivo de transacciones  
 Elige consulta, imprime o efectúa transacción  
 Repeat  
 Lee No. transacción  
 Lee transacción  
 With transacción do  
 begin  
 Identifica Archivo MODI-PROF  
 Identifica No.registro  
 Abre archivo MODI-PROF  
 Lee registro de la MODI-PROF  
 If consulta transacción then  
 Muestra el contenido del registro  
 If imprime registro then  
 Envía registro a impresora  
 If efectúa transacción then  
 begin  
 Identifica Archivo MLP  
 Identifica Movimiento al archivo  
 Abre archivo de MLP  
 Lee registro de MLP  
 Muestra ambos registros  
 If cambio de registro OK then  
 begin  
 Graba registro de archivo MODI-PROF  
 a archivo MLP  
 Borra registro de MODI-PROF  
 Borra transacción  
 Cierra archivo de MLP  
 end  
 Cierra archivo de MODI-PROF  
 end

end {with}  
 Until fin transacciones  
 Cierra archivo de transacciones.

## DESCRIPCION FIGURA 6.

### PROCESO: 2.1. Alta de alumno

DESCRIPCION: Acceso para que el profesor capture la información inicial de los estudiantes que formaran parte del grupo que interactúe con STIAYA

ENTRADAS: Clave del alumno, datos modelo estudiante.

SALIDAS: Datos personales alumnos, evaluación diagnóstica, aptitudes.

RESUMEN LOGICO: Abre archivos de MCP  
 Repeat  
 Lee datos personales alumno  
 Lee evaluación diagnóstica  
 Lee aptitudes alumno  
 If datos OK then  
 Graba registro  
 Until fin captura

### PROCESO: 2.2. Baja de alumno

DESCRIPCION: Permite al profesor eliminar todos los datos sobre un alumno específico.

ENTRADAS: Clave del alumno.

SALIDAS: No. de registros para eliminar de la MCP.

RESUMEN LOGICO: Lee clave del alumno  
 With clave do  
 Repeat  
 Abre archivos de la MCP  
 Repeat  
 elimina registros con la clave del alumno  
 Until fin registros

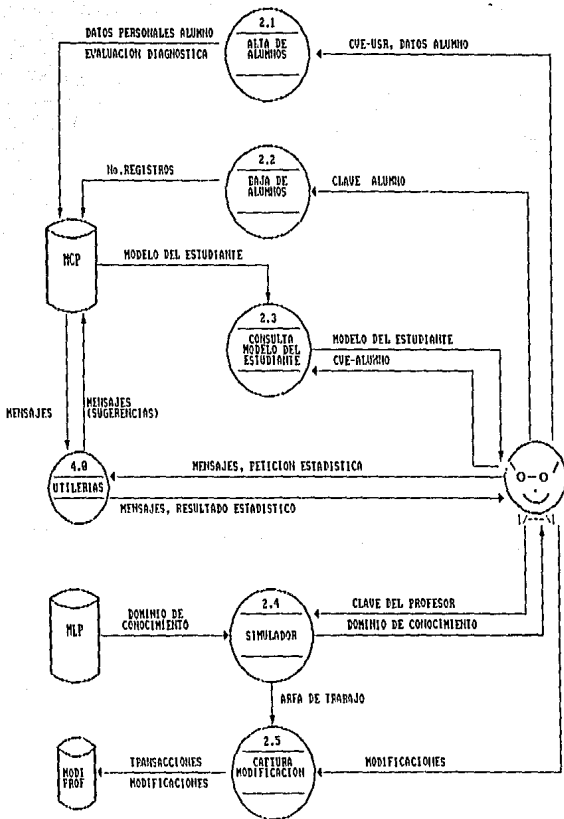


FIG.6 D.F.D. DE LA INTERFAZ DEL PROFESOR

Cierra archivo de la MLP  
 Util fin archivos

**PROCESO: 2.3. Consulta Modelo del estudiante**

**DESCRIPCION:** Acceso para que el profesor revise el desempeño del estudiante en cualquier instante del curso.

**ENTRADAS:** Clave del alumno.

**SALIDAS:** Modelo del estudiante.

**RESUMEN LOGICO:** Lee clave del alumno  
 Abre archivos de la MCP  
 Repeat  
 Presenta parámetros p/revisión  
 Elige opción  
 Presenta datos  
 Util fin consulta  
 Cierra archivos de la MCP.

**PROCESO: 2.4 Simulador**

**DESCRIPCION:** Permite la revisión del material de estudio en forma similar a la de un estudiante, pero con opción a indicar las modificaciones necesarias.

**ENTRADAS:** Clave del profesor, dominio de conocimiento.

**SALIDAS:** Dominio de conocimiento, área de trabajo.

**RESUMEN LOGICO:** Identificación de usuario  
 Repeat  
 Presenta Material  
 If modificación then  
 begin  
 Identifica archivo  
 Identifica área de trabajo  
 Call captura modificación  
 end  
 Util fin simulador

**PROCESO: 2.5 Captura Modificación**

**DESCRIPCION:** Registra en MODI-PROP, o mediante mensajes, los cambios al dominio de conocimiento, sugeridos por el profesor.

**ENTRADAS:** Modificaciones, área de trabajo.  
**SALIDAS:** Transacciones y modificaciones.

**RESUMEN LOGICO:** Abre archivo de transacciones  
 Lee archivo de MLP  
 Lee No.registro de MLP  
 If archivo en MODI-PROF then  
 begin  
 Escribe transacción  
 Abre archivo de MODI-PROF  
 Repeat  
 Escribe modificación  
 Until datos OK  
 Graba registro en MODI-PROF  
 Cierra archivo de MODI-PROF  
 end  
 else  
 begin  
 Indica archivo y No. de registro  
 Call Ufile(maestro) (correo)  
 end  
 Cierra archivo de transacciones.  
 end

## DESCRIPCION DE LA FIGURA 7.

**PROCESO:** 3.1 Tutor

**DESCRIPCION:** Controla la interacción del alumno con el sistema, así como las llamadas a diferentes procesos para dar respuesta a todas y cada una de las peticiones del estudiante; entre otras permite que el estudiante se comunique con el profesor y el analista utilizando un sistema de correo interno.

**ENTRADAS:** Clave usuario, peticiones, resultado petición, mensajes, plan de enseñanza adecuado, modelo del estudiante, dominio de conocimiento, retroalimentación, recomendaciones.

**SALIDAS:** Mensajes, retroalimentación, recomendaciones, dominio de conocimiento, ambiente, petición, nombre programa, modelo actualizado del estudiante.

**RESUMEN LOGICO:** Identificación de usuario  
 Aviso mensajes  
 Ubicación del área de trabajo  
 Repeat  
 If CTRL.sistema then  
 begin  
 Call Diagnóstico

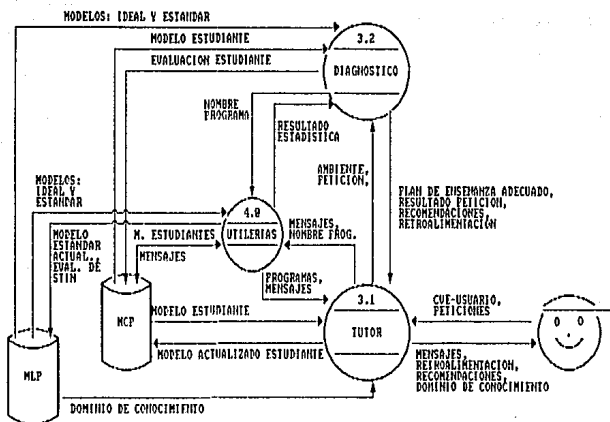


FIG.7 D.F.D. INTERFAZ DEL ESTUDIANTE

```

    presenta lección
end
If Ctrl alumno
begin
    Presenta Opciones
    Call Diagnóstico
    Notifica resultado opción
end
If fin tema then
begin
    Realiza examen
    Call Diagnóstico
    Presenta retroalimentación
    Presenta Recomendaciones
end
Until fin sesión

```

**PROCESO:** 3.2 Diagnóstico

**DESCRIPCION:** Realiza todas las evaluaciones necesarias para decidir cómo se va a presentar el material al estudiante, cómo evaluarlo, validar las peticiones del alumno, generar el plan de enseñanza adecuado, actualizar el modelo estandar y evaluar al sistema.

**ENTRADAS:** Area de trabajo, petición, resultado estadística, modelo estudiante, modelo ideal, modelo estandar.

**SALIDAS:** Plan de enseñanza adecuado, resultado petición, recomendaciones, retroalimentación, nombre programa, evaluación del estudiante.

**RESUMEN LOGICO:** If presenta material then  
     Define orden y presentación  
 If peticion then  
   begin  
     Lee petición  
     Define resultado de petición  
   end  
 If evaluación del estudiante  
   begin  
     Define retroalimentacion  
     Define recomendaciones  
   end  
 If actualiza modelo estandar  
   begin  
     Lee modelos de estudiantes  
     Escribe modelo estandar actualizado  
   end  
 If evaluación sistema  
   begin  
     compara modelos  
     Escribe evaluación del sistema  
   end.

**DESCRIPCION DE LA FIGURA 8.****PROCESO: 3.1.1. Monitor de Control**

**DESCRIPCION:** Identifica el control de la interacción, ya sea por parte del estudiante o por parte del sistema, para elegir las diferentes rutas a seguir en la comunicación con el alumno.

**ENTRADAS:** Clave Usuario, peticiones, mensajes, área de trabajo(ambiente), avance, retroalimentación, recomendaciones.

**SALIDAS:** Clave usuario, mensajes, área de trabajo, avance, retroalimentación, recomendaciones, fin de tema, fin de curso.

**RESUMEN LOGICO:** Identificación de usuario

```

If primera vez then
  Presenta bienvenida al curso
Call Ubicación del área de trabajo
While sesión do
begin
  If CTRL sistema then
begin
  Call presenta dominio de conocimiento
  Call ubicación área de trabajo
  If fin tema then
begin
  Realiza examen
  Call diagnóstico(evalua estudiante)
  Presenta retroalimentación
  Presenta recomendaciones
end
end
else CTRL alumno
begin
  Call ubicación del área de trabajo
  Call ESCAPE
  If petición no OK then
  Notifica resultado opción
end
end
end

```



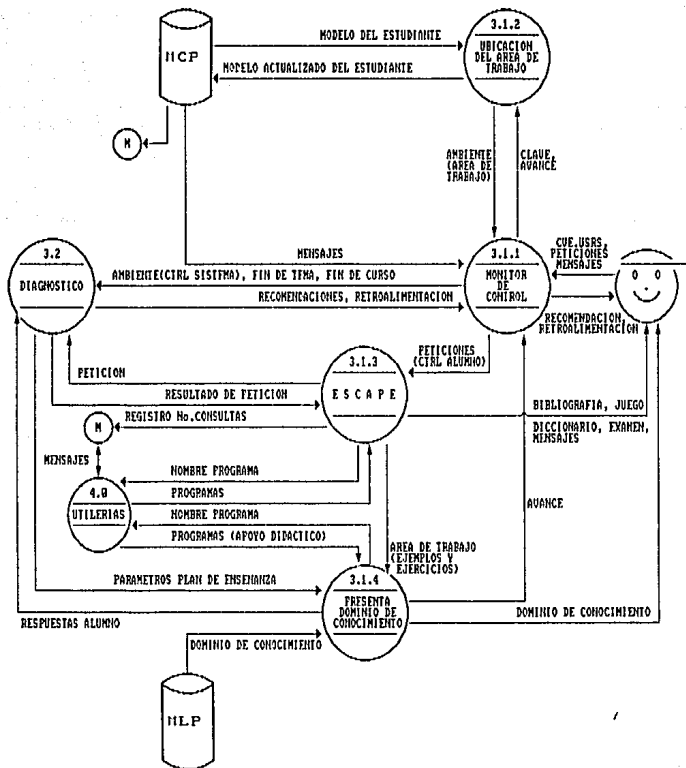


FIG. 8 D.F.D. DEL PROCESO TUTOR

**PROCESO: 3.1.2. Ubicación del Área de trabajo**

**DESCRIPCION:** Reconstruye el área de trabajo de la sesión anterior para iniciar la nueva, determina la dirección actual del estudiante en su avance por el plan de enseñanza y relocaliza un área de trabajo interrumpida.

**ENTRADAS:** Clave Usuario, modelo del estudiante.

**SALIDAS:** Área de trabajo, modelo actualizado del estudiante.

**RESUMEN LOGICO:** Lee clave del usuario  
 If entrada then  
   Localiza última área de trabajo  
 If avance then  
 begin  
   Lee avance alumno  
   Actualiza modelo del estudiante  
   área de trabajo = siguiente lección  
 end  
 If ESCAPE then  
   Guarda área de trabajo.

**PROCESO: 3.1.3. Escape <Ctrl Alumno>**

**DESCRIPCION:** Presenta las diferentes opciones que tiene el alumno para dirigir sus experiencias de aprendizaje

**ENTRADAS:** Peticiones alumno, resultado de petición, programa de propósito general.

**SALIDAS:** Petición, nombre programa: Diccionario, bibliografía, exámen, otro tema, juego.

**RESUMEN LOGICO:** Repeat  
 presenta opciones  
 elige opción  
 if continuar plan then  
 begin  
   presenta dominio de conocimiento  
 if Índice  
 begin  
   call Índice  
   registra consulta  
 end

```

If Otrotema then
begin
  Call diagnóstico
  If otro tema OK then
  Indica No.otro tema
  else
    Notifica otro tema inválido
  end
If bibliografía then
begin
  Call bibliografía
  Registra consulta
end
If diccionario then
begin
  Call diccionario
  Registra consulta
end
If examen then
begin
  Call diagnóstico
  If examen OK then
  begin
    Presenta examen
    Califica examen
    Evalúa examen
    Emite recomendación
  end
  else
    Notifica NO examen
  end
If correo then
begin
  Call utilerfas
  Registra correo
end
If juego then
begin
  Call diagnóstico
  if juego OK then
  begin
    Call utilerfas y registra juego
  end
  else
    Notifica NO al juego
  end
Until salida ESCAPE.

```

#### PROCESO: 3.1.4. Presenta Dominio de Conocimiento

**DESCRIPCION:** Presenta el dominio de conocimiento siguiendo el plan de enseñanza o por petición del alumno.

**ENTRADAS:** área de trabajo, dominio de conocimiento.

**SALIDAS:** Avance, dominio de conocimiento.

**RESUMEN LOGICO:** Lee área de trabajo

```

While (not fin curso) or (monitor <> ESC) do
begin
  Lee Modelo alumno
  If inicio tema then
    Case tipo alumno then
      bueno, regular: presenta Obj. Gral.
      malo ; Presenta Obj . Gral.
      Presenta Organizador Ant.
    If fin tema then
      Call Examen
      If Lección then
        begin
          LimpiaBuffer Avance Alumno
          Define concepto
          Repeat
            Call diagnostico orden
            Call diagnostico tipoPresentación
            Identifica elemento
            Call Area de trabajo
            Presenta Información
            Actualiza Buffer Avance Alumno
            Define elemento siguiente
            If org.ante OR resumen OR
              ejemplo OR ejercicio OR
              analogía OR red then
              begin
                Call Diagnóstico
                If existen then
                  begin
                    Presenta opciones
                    Call Diagnóstico Orden
                    Call Diagnóstico Tipo de Presentación
                    Identifica elemento
                    Call Area de trabajo
                    Presenta Información
                    Actualiza Buffer Avance Buffer Avance
                  end
                end
              Until Completa Concepto
            If Preguntar then
              begin
                lee pregunta
                presenta pregunta
                lee respuesta
                registra respuesta
              end
            end
          end
        end
      end
    end
  end
end

```

```

        responde respuesta
    end
    Actualiza Avance Alumno
end {curso}

```

## DESCRIPCION DE LA FIGURA 9.

**PROCESO:** 3.2.1 Como presentar el material

**DESCRIPCION:** Realiza todas las evaluaciones necesarias para generar el plan de enseñanza adecuado; la forma, estrategia y orden.

**ENTRADAS:** Area de trabajo, modelo estudiante, modelo ideal, modelo estandar, trayectorias, mapa de contenido,

**SALIDAS:** Plan de enseñanza adecuado.

**RESUMEN LOGICO:** If orden then  
 begin  
     Consulta trayectorias  
     compara modelos  
     evalua trayectorias  
     define siguiente lección  
 end  
 If elementos de enseñanza then  
 begin  
     Consulta Recursos existentes  
     Identifica Habilidades  
     Crea Contexto alumno  
     Aplica reglas heurísticas  
     Define siguiente elemento  
     end  
 If forma then  
 begin  
     Lee modelo del estudiante  
     define forma  
 end  
end

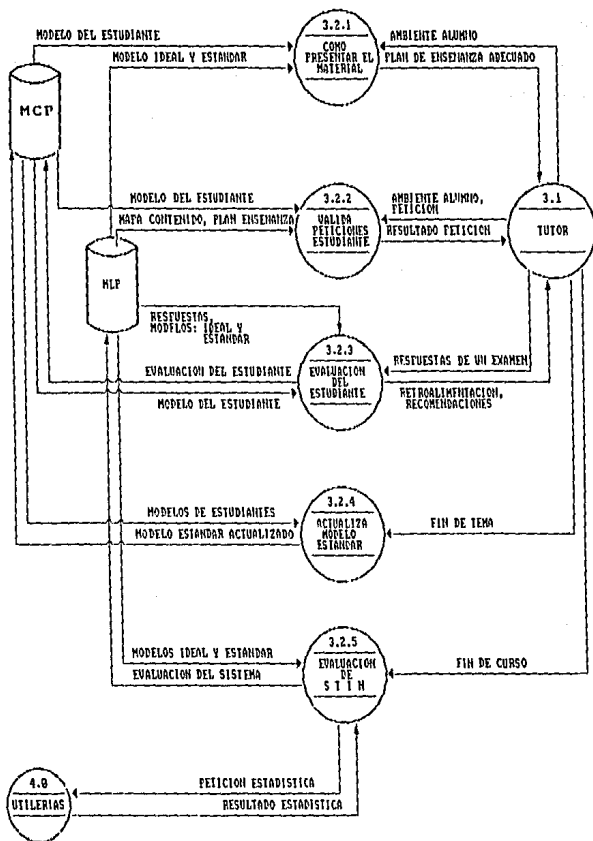


FIG.9 D.F.D. DEL PROCESO DIAGNOSTICO

**PROCESO: 3.2.2 Vnlda Peticiones**

**DESCRIPCION:** Realiza las advertencias pertinentes al alumno sobre la posibilidad, de su petición. Es un semáforo en el camino de aprendizaje del alumno.

**ENTRADAS:** Area de trabajo, petición, modelo estudiante, plan de enseñanza, trayectorias, mapa de contenido.

**SALIDAS:** Resultado de la Petición.

**RESUMEN LOGICO:** Identifica la petición

```

If org.antiop. OR resumen OR
ejemplos OR ejercicios OR
analogías OR redes then
begin
  If existe nuevos elementos then
    petición OK
  else
    notifica inexistencia
end
if examen then
If avance >= X 0% then
  petición OK
Else
  Justifica petición inválida
if otro tema then
  if área de trabajo completa then
    if nuevo tema es paralelo then
      petición OK
    else
      justifica petición inválida
    else
if Juego then
  If avance OK
    máximo 10 mins.
  else
    máximo 5 mins.

```

**PROCESO: 3.2.3 Evaluación del Estudiante**

**DESCRIPCION:** Después que se le ha aplicado un examen al alumno este proceso es el encargado de darle retroalimentación y recomendaciones. La primera se refiere a la solución correcta del examen y a la información general del desempeño del estudiante. Y la segunda se hace con respecto a la bibliografía y algún problema de antecedentes.

**ENTRADAS:** Exámen contestado, modelo del estudiante, modelo ideal, modelo estandar, mapa de contenido, libros.

**SALIDAS:** Evaluación del estudiante, retroalimentación y recomendaciones.

**RESUMEN LOGICO:** Checa resultado examen alumno  
 Lee Evaluación Formativa  
 compara modelos  
 define evaluación  
 informa sobre desempeño  
 recomienda bibliografía  
 realiza recomendaciones

**PROCESO:** 3.2.4 Actualiza Modelo Estandar.

**DESCRIPCION:** Actualiza el Modelo estandar de la población cada vez que un grupo termina un tema.

**ENTRADAS:** Modelos de estudiantes, fin tema.

**SALIDAS:** Modelo Estandar actualizado.

**RESUMEN LOGICO:** if fin tema then  
 begin  
 lee modelos de estudiantes  
 calcula el promedio  
 escribe modelo estandar actualizado  
 end

**PROCESO:** 3.2.5 Evaluación de STIAYA

**DESCRIPCION:** Evalua la eficacia del sistema al finalizar un curso.

**ENTRADAS:** Fin curso, modelo ideal modelo estandar, resultado estadístico.

**SALIDAS:** Petición estadística, Evaluación del sistema

**RESUMEN LOGICO:** if fin curso then  
 begin  
 call utilerías  
 compara modelos estandar e ideal  
 define evaluación  
 escribe evaluación del sistema  
 end



#### **ALMACENAMIENTO DE DATOS: MEMORIA A LARGO PLAZO (MLP)**

**DESCRIPCION:** Es la base de datos que contiene el dominio de conocimiento, almacenado de manera descriptiva o de procedimiento, el modelo ideal del estudiante y el modelo estándar del alumno. (Ver Figura 10)

**ARCHIVOS DE DATOS:** Temario,  
Mapa de contenido,  
Plan de enseñanza,  
Elementos de enseñanza  
Libros, Bibliografía,  
Diccionario,  
Habilidades,  
Banco de preguntas de OM,  
Banco de respuestas de OM,  
Matriz de conductas,  
Reglas

#### **ALMACENAMIENTO DE DATOS: MEMORIA A CORTO PLAZO (MCP)**

**DESCRIPCION:** Es la estructura que contiene el modelo de aprendizaje del alumno; desempeño general del estudiante e su historia académica (Ver figura 11).

PLAN ENSEÑANZA										
PLANENSESE					PLANENSEE					
TELELEM	NUM EJER	NUM EJERC	NUM ANALOG	NUM RED	PREGUNTA	MAX EJER	MAX EJERC	MAX ANALOG	MAX RED	
S6	INT	INT	INT	INT	BOOLEAN	INT	INT	INT	INT	
LLAVE										

TEMARIO			
RETEMARIO		TEMARIO	
TEMA	NUM-TEMA	OBJETIVO	% CURSO
S4	S60	S250	INT
LLAVE			

LIBROS					
RESBROS					LIBROS
NUM-LIBRO	TITULO	AUTOR	EDITORIAL	EDICION	LUGAR
S4	S80	S32	S20	INT	S20
LLAVE					

REGLAS	
LADO IZQ	LADO DER
S11	CHAR
LLAVE	

BIBLIOTECA		
TEMA	TIPO-OBIB	NUM-LIB-REV
S4	S2	S4
LLAVE		

REVISTAS						
REVREVISTA				REVISTA		
NUMREV	HOMBRE-REV	VOLUMEN	YEAR	ARTICULO	AUTOR	PAGINAS
S4	S60	S2	INT	S80	S32	S10
LLAVE						

DICCIONARIO		
TERMINO	TEMA	DEFINICION
S35	S4	S250
LLAVE		

RESMATCO		MATCON			
MATRIZ DE CONDUCTAS					
CUENSR	TEMA	H1	H2	H3	H4
S6	S4	INT	INT	INT	INT
LLAVE					

BANCO DE PREGUNTAS DE OPCION MULTIPLE				
TELELEM	NUM-PREG	TIPO-PREG	PREGUNTA	HABILIDAD
S6	S2	S2	S250	S1
LLAVE				

BANCO RESPUESTAS DE OPCION MULTIPLE					
TELELEM	NUM-PREG	TIPO-PREG	OP-RESP	VALOR	EXPLICA
S6	S2	S2	S80	BOOLEAN	S250
LLAVE					

MAPA DE CONTENIDO			
TEMA	TIPO-MODO	TEMA-REL	TIPOREL
S4	CHAR	S4	CHAR
LLAVE			

ELEMENTOS DE ENSEÑANZA			
TELELEM	RECURSO	TIPORES	OBSERVACION
S6	CHAR	CHAR	CHAR

FIGURA 10. MEMORIA A LARGO PLAZO

**ARCHIVOS DE DATOS:**

Datos personales,  
 Ambiente,  
 Mensajes,  
 Resumen de habilidades,  
 Temas vistos,  
 Exámenes,  
 Evaluación formativa,  
 Consultas,  
 Evaluación diagnóstica,  
 Avance,  
 Evaluación sumaria.

**ALMACENAMIENTO DE DATOS: MODIFICACIONES\_PROFESOR (MODI-PROF)**

**DESCRIPCION:** Es una estructura similar a la MLP que almacena el material del dominio de conocimiento, donde se colocan temporalmente todas las recomendaciones del profesor tendientes a enriquecer, corregir o modificar la información contenida en la base de datos. Además cuenta con un archivo de transacciones, donde se indican los cambios específicos.

**ARCHIVOS DE DATOS:**

Transacciones  
 Temario,  
 Mapa de contenido,  
 Plan de enseñanza,  
 Libros, Bibliografía,  
 Diccionario,  
 Habilidades,  
 Banco de preguntas Opción Mult.,  
 Banco de respuestas de OM,  
 Matriz de conductas,

**II.2 REQUERIMIENTOS DEL SISTEMA**

Este tipo de sistemas requieren trabajar en tiempo real debido a la dinámica necesaria para la interacción tutor-alumno. Además se debe contar con suficiente espacio en memoria primaria y secundaria para almacenar el dominio de conocimiento y la información referente al grupo de estudiantes, así como para el área de trabajo definida en cada sesión. Es importante poder manejar colores que contribuyan a

USUARIO DATOS PERSONALES							
CUEUSR	PASSWORD	NOMBRE	FECHAINS	SEXO	EDAD	CARREPA	NIVEL
S6	S6	S32	S10	CHAR	INT	INT	INT
LLAVE							

REGANZIENTE AMBIENTE				AMBIENTE
CUEUSR	ELE-REF	ELE-VISTO	ELE-ACTUAL	FIN-SEC
S6	S9	S9	S9	BOOLEAN
LLAVE				

RESCOPY MENSAJES			
CUEUSR	CUEUSR	Mensaje	FECHA
S6	S6	S250	S10
LLAVE			

REGAFITUD ACTITUD				
HABILIDADES				
CUEUSR	H1	H2	H3	H4
S6	INT	INT	INT	INT
LLAVE				

REGEVADIAS EVADIAS				
EVAL. DIAGNOSTICA				
CUEUSR	H1	H2	H3	H4
S6	INT	INT	INT	INT
LLAVE				

REGEXAMENAL EXAMENES							
CUEUSR	TEMA	# DE EXAMEN	RESULTADO	H1	H2	H3	H4
S6	S4	INT	BOOLEAN	INT	INT	INT	INT
LLAVE							

REGVALFORMA EVALFORMA			
EVAL. FORMATIVA			
CUEUSR	TELELEM	# FREG	TIPO RESP OR
S6	S6	S2	CHAR
LLAVE			

REGCONSULTA CONSULTA			
CUEUSR	TEMA ORIG	TEMA CONTS	NUM VECES
S6	S4	S4	INT
LLAVE			

REGEXAMEN EXAMEN TEXTORAL				
CUEUSR	TEMA	NUM-REG	HR-RESP	NIVEL
S6	S4	LONGINT	LONGINT	INT
LLAVE				

REGAVANCE AVANCE			
AVANCE			
CUEUSR	TELELEM	RECURSO	OPTI-PRES
S6	S6	CHAR	CHAR
LLAVE			

REGVALSUM EVALUACION SUMARIA																	
CUEUSR	TEMA	RESULT	H1	H2	H3	H4	# EXAM	# DICC	# BIB	# RESUM	# ORG-ANT	# EJEM	# EJER	# ANAL	# RED	# SESION	# JUEGO
S6	S4	BOOLEAN	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT
LLAVE																	

FIGURA 11. MEMORIA A CORTO PLAZO

presentar el material de forma más didáctica. Por tanto, los recursos técnicos mínimos necesarios para implantar un sistema de este tipo son:

- \* una computadora PC compatible
  - procesador 8088
  - 640K RAM
- \* disco duro 20 Mb
- \* monitor CGA

Este equipo se seleccionó por cubrir los requerimientos de velocidad y almacenamiento del sistema así como por ser muy comercial, de bajo costo y con posibilidades de integrar un sistema multiusuario. En este último caso, es recomendable cambiar a un procesador 80286, por lo menos, para que el sistema no vea minimizada su velocidad.

Por otra parte, para facilitar el mantenimiento de STIAYA, se realizará un diseño modular, el cual será programado en TURBO PASCAL V.5.5, ya que es un lenguaje que cuenta con un compilador rápido y proporciona un ambiente de trabajo que ofrece muchas facilidades al programador. Además permite acceder fácilmente utilerías, tales como TOOLBOX IV etc.

## CAPITULO III

### PRUEBAS Y RESULTADOS DE STIAYA

Este capítulo tiene como objetivo mostrar la operación del sistema tutorial que proponemos. En la primera sección establecemos las condiciones de arranque del sistema mientras que en la segunda presentamos ejemplos mediante los cuales pretendemos ilustrar cómo se utilizan los elementos propuestos en los procesos de adaptabilidad y de autoevaluación.

#### III.1 CONDICIONES INICIALES

El sistema propuesto fue diseñado para operar libre de contenido, sin embargo, para fines de prueba del funcionamiento de STIAYA consideramos una serie de datos iniciales correspondientes a la información que debe tener la Memoria a Largo Plazo (MLP) y la Memoria a Corto Plazo (MCP).

El contenido de la MLP corresponde principalmente a diversos parámetros relacionados con la materia que se va a estudiar y cómo se va a presentar; por ejemplo, deben estar listos el temario, el mapa de contenido, el plan de enseñanza, los elementos de enseñanza y la matriz de conductas, principalmente.

Los datos de los parámetros que vamos a utilizar como ejemplo se muestran en las siguientes tablas.

MAPA DE CONTENIDO			
Tema	TipoNodo	TipoRel	TemaRel
0100	I	S	0200
0200	F	A	0100
0101	I	S	0102
0101	I	S	0103
0102	F	A	0101
0202	M	A	0201
0202	M	S	0203
0203	F	A	0202

MATRIZ DE CONDUCTAS					
USR	TEMA	NP11	NP12	NP13	NP14
	0101	1	1	1	1
	0102	2	2	2	2
	0103	1	1	1	1
	00201	1	1	1	1

PLAN DE ENSEÑANZA							ME	MJ	MA	MR
TELELEM	NEP	NJP	NAP	NUP	PREGUNTA					
010000	0	0	0	0	FALSE	0	0	0	0	
010100	0	0	0	0	FALSE	0	0	0	0	
010101	1	1	1	1	TRUE	2	2	1	1	
020301	1	1	1	1	TRUE	2	2	2	2	

El contenido de la MCP se refiere a los datos personales del estudiante y al resultado de una evaluación diagnóstica que revele el dominio de sus habilidades: en cuatro niveles cognoscitivos: conocimiento, comprensión, análisis y aplicación. A partir del resultado promedio del grupo de estudiantes que utilizó STIAYA, se obtiene el diagnóstico de grupo al inicio del curso (STAND1). La MCP también incluye los parámetros considerados como ideales y los porcentajes deseables de habilidades para cada lección de la materia en estudio. Cabe mencionar que de acuerdo a la evaluación diagnóstica del alumno, éste se puede clasificar como bueno, regular o malo. A continuación se muestra un ejemplo de ésta primera evaluación:

Alumno	EVALUACION DIAGNOSTICA			
	Conocimiento	Comprensión	Análisis	Aplicación
IDEAL1	90	90	90	90
BBB100	90	90	80	80 bueno
RRR100	60	60	60	50 regular
KMM100	40	30	25	20 malo
STAND1	63	60	55	50

La coherencia en la información anterior (MLP y MCP) es de vital importancia para que STIAYA pueda realizar la función de adaptabilidad y autoevaluación.

## III.2 OPERACION DEL SISTEMA

Para trabajar con STIAYA es necesario tener una clave y "password" de acceso con objeto de que el alumno pueda identificarse ante el sistema. Como primer punto, se despliega una bienvenida personalizada, se define el plan de enseñanza y se activa el modelo del estudiante que va a trabajar.

El ciclo de instrucción desarrollado en STIAYA se lleva a cabo en tres etapas: a) objetivos, b) enseñanza y c) evaluación. Así, para mostrar la operación de nuestro tutor hemos hecho la siguiente clasificación de interacciones: a) sesión inicial, b) sesión intermedia y c) sesión de evaluación.

### A.- Sesión Inicial

La primera vez que algún estudiante utiliza el sistema se le presenta una breve introducción al curso que está por iniciar, en la cual se le indican los objetivos generales, el resultado de su evaluación diagnóstica y las recomendaciones pertinentes de acuerdo a éste resultado. La forma de interacción entre el tutor y el estudiante es por medio de menús y preguntas de respuestas cerradas (de opción múltiple).

Una vez identificado el estudiante, STIAYA construye el ambiente de trabajo que le corresponde, es decir, lo ubica dentro del plan de enseñanza. En esta sesión lo coloca al inicio y

comienza la instrucción por parte del tutor.

### ***B.-Sesión Intermedia.***

Esta sesión tiene lugar después de la bienvenida al curso y se refiere a todas las interacciones necesarias para cubrir el plan de enseñanza. Al tener la ubicación del alumno se puede inferir cuál fue su última actividad, la cual es el punto de partida para las demás lecciones y/o elementos de instrucción.

Como se mencionó en el diseño, la secuencia y forma de presentación de los elementos de instrucción se definen de acuerdo a reglas de producción, en cuyo contexto se considera como elemento más importante la clasificación del estudiante (tipo de alumno). Sin embargo, también se tomó en cuenta la relación entre las habilidades requeridas para cada tópico y las que posee el estudiante, además se contempla el número de elementos complementarios a un concepto propuestos por el profesor en el plan de enseñanza (columnas NIP,NJP,NAP,NRP del la tabla del plan de enseñanza).

De manera que, considerando un tipo de alumno regular para nuestro ejemplo, STIAYA define primero el tema y/o lección que debe presentar, para lo cual utiliza el mapa de contenidos y el avance del alumno, con lo que se evita presentar en secuencia incorrecta o duplicada las lecciones del plan de enseñanza. De hecho, el sistema primero determina el tema, presenta objetivos generales del mismo e inmediatamente después determina cuál será la primer lección en esta etapa.

Por tratarse de un alumno regular, también se le presentan los objetivos particulares por lección. Enseguida de esto viene la presentación de conceptos y procedimientos, ordenados y especificados de acuerdo al plan de enseñanza. Esto significa que las lecciones están compuestas por uno o varios conceptos y/o procedimientos, los cuales se exponen secuencialmente.

De acuerdo con el plan de enseñanza cada concepto cuenta con cierta cantidad de elementos de apoyo didáctico, cuya secuencia y forma de presentación lo definen las reglas de acuerdo al contexto del alumno en cada momento. Así, la secuencia que se le presentaría al alumno de nuestro ejemplo es:

- Alumno Regular**
- Objetivo General (muevo Tema)
  - Objetivo Especifico (lección)
  - Concepto y/o procedimiento
    - Ejemplo
    - Analogía
    - Ejercicio
    - Real
    - Pregunta
    - Resumen (del tema)

Después de la presentación de cada elemento de instrucción, éstos se van contabilizando en un buffer hasta llegar al número de elementos propuesto por el plan de enseñanza. Al mismo tiempo se actualiza el contexto del alumno en lo que se refiere a recursos disponibles, además de indicar cuál fue el elemento visto.

Al finalizar la revisión de cada concepto y/o procedimiento, si el plan de enseñanza indica que se debe realizar una pregunta, entonces se selecciona del base de datos de preguntas, ya que cada pregunta del dominio de conocimiento que está siendo estudiado se relaciona con alguna habilidad específica. Esto es con el objeto de apoyar las habilidades en las que el alumno está fallando. La respuesta de la pregunta se evalúa de inmediato y se le presenta el resultado al alumno. La pregunta pasa

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**



a formar parte de la evaluación formativa del modelo de estudiante .

Cada vez que el alumno complete la revisión de un concepto y/o procedimiento junto con sus elementos de apoyo didáctico se registrará en el archivo de Avance lo que ha visto y cómo lo ha visto. Si decide interrumpir antes de terminar dicha secuencia, no se considerará visto ese concepto y/o procedimiento, de manera que cuando regrese a este mismo punto deberá revisarlo su definición .

Hasta aquí solo nos hemos referido al control de la enseñanza por parte del sistema. Sin embargo, en cualquier momento del proceso de instrucción el alumno puede interrumpir durante la presentación de elementos de apoyo didáctico y controlar la secuencia de presentación, esto significa que puede seleccionar a su gusto cualquiera de los siguientes elementos:

#### ELEMENTOS DE APRENDIZAJE

Organizador Anticipado

Resumen

Ejemplo

Ejercicio

Analogía

Red

<FIN> para volver al tutor

Con la opción de Organizador anticipado puede ver una introducción contextual del tema, la opción de resumen le permite ver una síntesis de la información del tema y las opciones de ejemplo, ejercicio, analogía y red son apoyo didáctico para el concepto que está aprendiendo el estudiante, la única limitante es el número de elementos existentes en la base para cada concepto.

Otra forma de que el alumno controle sus actividades de aprendizaje es oprimir la tecla de ESCAPE con lo que interrumpe la enseñanza del concepto y pasa a un menú de actividades complementarias:

#### STIAYA

M E N U    A L U M N O

Continuar con el plan de enseñanza

Temario

Otros Temas

Bibliografía

Diccionario

Exámenes

Mensajería

Juego

Salida

La opción **Temario** sirve para que el alumno pueda revisar el índice de temas y lecciones, así como los objetivos de cada una de ellas. Con la opción **Otros Temas**, el alumno puede elegir algún tópico, mismo que será analizada tomando en cuenta las relaciones definidas en el mapa de contenido para darle acceso a la información requerida o justificarle por qué no. En el caso de un tema visto, STIAYA le presenta la secuencia de elementos que originalmente revisó. Si se trata de un tema paralelo el sistema revisa que el estudiante tenga haya finalizado el tema que se le estaba presentando antes de la interrupción. Por último, si el alumno elige un tema siguiente STIAYA le negará el acceso.

La opción de **Bibliografía** le muestra al alumno nombres de libros y revistas de apoyo para cada lección, mientras que la opción **Diccionario** es un glosario relacionado con los temas del curso que el estudiante puede consultar cuantas veces lo desee. Puede hacer la revisión del glosario de dos formas: a) por la letra inicial del término, en cuyo caso se le presenta una lista de términos para que seleccione el que busca o b) puede escribir el nombre de la definición que requiere.

La opción **Exámenes** le da oportunidad al alumno de avanzar más rápido en el curso, ya que si tiene un avance del 80%, por ejemplo, puede intentar realizar el examen correspondiente. Los exámenes son grabados temporalmente para que en el caso de que no hubiera terminado se continúe la próxima sesión con él. La opción de **Mensajería** permite al estudiante comunicarse con otros usuarios del sistema. O bien, leer y borrar los mensajes que algún otro usuario le envió. En la opción de **juegos** se le indica al alumno cuáles hay disponibles y se toma el tiempo que se dedica a jugar. Finalmente la opción de salida significa que termina la sesión. STIAYA almacena los datos que corresponden al avance del alumno, y contabiliza todas las actividades realizadas por el estudiante durante la sesión. Para ello emplea el archivo de evaluación sumaria porque a partir de éste se van a realizar las estadísticas del grupo (modelo estándar).

### *C. Sesión de evaluación*

Cuando finaliza una lección, de manera interna, STIAYA valora las preguntas de la evaluación formativa y el resultado obtenido, lo graba en el archivo de exámenes pues servirá para complementar la evaluación del tema. El resultado se considera como un examen parcial.

Ahora bien, cuando se llega al final de un tema se procede a realizar un examen tomando en cuenta el número de preguntas superado para cada habilidad en la matriz de conductas. La selección de preguntas se realiza en forma aleatoria para cada estudiante, a su vez éste contesta el examen y en cuanto termina se lo indica a STIAYA. La evaluación de este examen (Promovido o NO-promovido) se realiza enseguida y da lugar a una retroalimentación; si el desempeño fue bueno se felicita al alumno como estímulo, pero cuando hay errores, se indica el por qué.

La evaluación completa del tema considera los resultados de la evaluación formativa y del examen del tema. Ambos resultados se analizan en la parte de habilidades cognitivas y su promedio define la evaluación del tema. A partir de ésta y de la información del modelo del estudiante, STIAYA realiza sugerencias al alumno, las cuales pueden referirse a la consulta de temas ya vistos o recomendación sobre bibliografía relacionada con el tópico.

Al obtener la evaluación del tema, STIAYA realiza una nueva clasificación del estudiante a partir de la cual definirá la secuencia de elementos instruccionales apropiados para el nuevo tipo de estudiante que en nuestro ejemplo cambia a alumno bueno. La diferencia se reflejará en el menor número de elementos de instrucción que se le presentarán.

**Alumno Bueno**

- Objetivo General (nuevo tema)
- Concepto y/o procedimiento
  - Ejemplo
  - Ejercicio

Cuando todos los alumnos hayan terminado la misma lección, se llevará a cabo un proceso de actualización del modelo de estudiante estándar con objeto de representar el promedio de la población de estudiantes en términos de habilidades y del número de elementos instruccionales que se revisaron.

En los párrafos anteriores hemos descrito la función de adaptabilidad en la operación de STIAYA, ahora explicaremos cómo se llevará a cabo la evaluación del sistema en su función de auxiliar en el proceso de enseñanza-aprendizaje.

Los elementos que tenemos para evaluar el funcionamiento de STIAYA es un modelo de estudiante ideal y un modelo de estudiante estándar, cuyos parámetros serán comparados y la diferencia entre ellos nos dará la pauta para analizar si el plan de enseñanza propuesto es correcto o hay que modificarlo en función de los elementos que revisó el grupo de estudiantes.

Entre los parámetros de comparación consideramos las evaluaciones diagnóstica y final del estudiante estándar, la diferencia entre éstos nos indica si existe alguna mejora en el estado de conocimiento y dominio de habilidades del grupo de estudiantes que utilizó a STIAYA. Considerando que la diferencia sea positiva se procede a comparar las habilidades finales del estudiante estándar contra las habilidades del estudiante ideal. El resultado de esta comparación nos permitirá concluir si el plan de enseñanza fue el adecuado o no.

## CAPTULO IV CONCLUSIONES

De la revisión bibliográfica, presentada en el capítulo I, es claro que en el campo de la Educación el uso de sistemas computacionales para asistir el proceso instruccional han ido evolucionando paulatinamente y cada vez es mayor el interés por integrar en un mismo sistema los elementos necesarios para que esta función de auxilio en la instrucción se realice de manera más eficiente. Se asume de manera implícita que el éxito de estos sistemas computacionales dependerá, cada vez más, de la variedad de herramientas, que incluyan, para facilitar la labor instruccional del profesor y para estimular su uso por parte de los estudiantes. Debe señalarse que en la actualidad éste es un problema no resuelto, aunque ya se han identificado una serie de cuestiones cuya respuesta será el primer paso hacia la obtención de una solución; por ejemplo, Rosenberg (1987) plantea que para que los ITS's sean útiles en la instrucción deberán incluir en su diseño un método de enseñanza-aprendizaje así como herramientas que permitan al sistema evaluar su propia actuación.

Existen ya algunas propuestas para montar un proceso de enseñanza-aprendizaje en los ITS [Flouris, 1989; Shing, 1987; Gagne, 1991], pero no hay mucho sobre el problema de la autoevaluación. En la mayoría de los métodos de enseñanza-aprendizaje, las reglas para el diseño instruccional son muy vagas y subjetivas, lo que las hace difíciles de transportar al desarrollo de los ITS's. Esto ha provocado que en la construcción de algunos sistemas [Merril, 1990; Epstein, 1990] se haya cometido el error de utilizar la emulación del desempeño de un maestro cuando se para frente a un grupo como modelo instruccional.

### IV.1 APORTACIONES

En este trabajo hemos presentado en detalle el diseño de la Interfaz-Estudiante de un sistema tutorial, en el cual se ofrece una alternativa para resolver los problemas planteados por Rosenberg, éste es, se incluyen:

1. procedimientos que permiten involucrar al estudiante en el proceso de enseñanza-aprendizaje de una manera activa (e.g., se da al estudiante la posibilidad de autorregular su aprendizaje, otorgándole el control de la estrategia instruccional en algunas etapas, para que él defina su propia secuencia de acceso al material, revisión de ejemplos, realización de ejercicios, toma de exámenes, etc.);
2. otros procedimientos que permiten al sistema modificar su interacción con el estudiante de manera personalizada y en función de los resultados parciales observados (i.e., manejo combinado de resultados obtenidos por el estudiante en evaluaciones parciales y su modelo correspondiente en la definición de un proceso de adaptabilidad); y
3. algunas herramientas que nos permitan evaluar automáticamente el desempeño de nuestro sistema (e.g., manejo de modelos de estudiante, estudiante estándar y estudiante ideal para ajustar el Plan de Enseñanza de cada estudiante, durante el curso, y para definir el Plan de Enseñanza inicial del siguiente curso). En la creación y actualización de los diferentes Modelos de Estudiante, de acuerdo al modelo de enseñanza-aprendizaje considerado, para definir la el nivel de conocimiento que va adquiriendo el alumno en cada una de las etapas de evaluación y el manejo que es capaz de hacer de este conocimiento, se utilizaron variables que se refieren a las siguientes categorías del dominio cognitivo: a) conocimiento, b) comprensión, c) aplicación y d) análisis [de acuerdo a la taxonomía de Bloom, citado en CISE 1980].

Otro aspecto importante en nuestro diseño se refiere al hecho de involucrar al profesor como participante activo en la continuación del contenido de las Based de Datos (i.e., "Memoria a Largo

Plazo y Memoria a Corto plazo") y como el responsable del control y manejo instruccional, ya que consideramos que él es el más indicado para definir tanto el contenido como la estructura del curso. Asimismo, su interacción con el Ingeniero responsable del Mantenimiento, durante esta etapa, permite que el profesor no tenga necesariamente que saber de programación para participar en el desarrollo de su tutor. Con base a nuestro trabajo y a la revisión de la bibliografía que realizamos, podemos concluir que el uso de Tutores Computacionales, y en general la Ingeniería en Conocimiento, tendrán una mayor aceptación e impacto en otras disciplinas si son utilizados bajo este esquema.

Este tipo de Tutores debe pasar ciertas pruebas antes de utilizarse para impartir cursos completos con ellos; por lo que pensamos que uno de los posibles usos iniciales para nuestro sistema estaría en proporcionar a los estudiantes, unas semanas antes del inicio del nuevo semestre, uno de estos tutores con el material antecedente del curso que van a tomar. Con esto se evitaría que el profesor tenga que dedicar algunas clases para revisar y homogeneizar el nivel académico de los estudiantes antes de iniciar con el material propio del curso. En la mayoría de las escuelas éste es un problema cotidiano que consume una gran cantidad de tiempo y esfuerzo, por lo que el utilizar un Tutor Computacional como el nuestro representa una alternativa para resolver este problema. Adicionalmente, para los estudiantes éste tipo de sistemas les ofrece la oportunidad de observar y estudiar (e.g., a través de simulaciones y despliegues en animación) fenómenos cuyo cambio y movimiento sería muy difícil de apreciar si sólo se utilizarán los libros como herramienta de estudio. Asimismo, se contemplan casos en que el mismo estudiante pueda cambiar valores a una o varias variables de manera simultánea o en experimentos sucesivos y observar el efecto que producen dichos cambios en el fenómeno bajo estudio.

Cabe señalar que para verificar experimentalmente las conclusiones sobre la bondad de las propuestas presentadas en esta tesis, se debe terminar el diseño e implantación de la interfaz de Mantenimiento e Interfaz del Profesor mencionadas en el diseño conceptual de la arquitectura Global de STIAYA.

#### IV.2 ETAPAS FUTURAS

En esta tesis se presentó el diseño detallado de la Interfaz Estudiante, de la arquitectura global de STIAYA, y para probar el funcionamiento de las características más importantes tuvimos que construir manualmente las bases de datos que deberán ser conformadas por la interfaz Mantenimiento; asimismo para poder tener control y seguimiento de los estudiantes que interactúan con STIAYA se debe implantar la Interfaz del Profesor. Por lo tanto, de manera natural, las siguientes etapas deben contemplar el diseño y construcción de los demás módulos de STIAYA, así como de las interfaces definidas en la arquitectura global del sistema.

Nuestra implementación tuvo como objetivo verificar si los elementos de adaptabilidad y autoevaluación propuestos en STIAYA, por lo que en etapas futuras, para contar con un sistema más versátil y amigable, se podían utilizar las bondades de nuevas tecnologías, tales como multimedia, lenguajes de cuarta generación, manejadores de grandes bases de datos, etc.

Dentro de la taxonomía de Bloom se mencionan otros dos niveles cognoscitivos, síntesis y evaluación, los cuales no fueron incluidos en STIAYA debido a que, para poder evaluarlos el sistema debe poder manejar preguntas abiertas y esto implica contar con un sistema completo de entendimiento de lenguaje natural. Por tanto, sería interesante trabajar sobre su diseño e incorporación al STIAYA.

Consideramos que nuestra propuesta de autoevaluación y adaptabilidad en STIAYA es buena, pero es necesario buscar herramientas que nos permitan definir formalmente algunas métricas (e.g., índices) para medir el grado de avance del estudiante o del desempeño de nuestro sistema; por ejemplo, alguna función de evaluación para determinar de manera concreta las capacidades de autoevaluación y adaptación en los FIS.

## BIBLIOGRAFIA

- Anderson, J., Reiser, B. (1985): "The Lisp Tutor", BYTE, Abril, Vol 10, # 4, pp. 159-175.
- Barr A., Feigenbaum E. (1981): 'The Handbook of Artificial Intelligence' Vol. I, Ed. William Kaufmann, Inc. U.S.A.
- Barr A., Feigenbaum E. (1982): 'The Handbook of Artificial Intelligence' Vol. II, Ed. William Kaufmann, Inc. U.S.A.
- Carreño E. (1987): Enfoques y Principios Teóricos de la Evaluación, Trillas, México.
- Castañeda M. (1982): Análisis del aprendizaje de Conceptos y Procedimientos, Trillas, México.
- CISE (1980): Sistematización de la Enseñanza, UNAM, México.
- De la Vega M. (1986): Introducción a la Psicología Cognitiva, Ed. Alianza, México.
- Díaz F. Diseño de Estrategias Instruccionales Cognoscitivas, <apuntes> Facultad de Psicología, UNAM.
- Duchascl P. (1989): "ICAL Systems: Issues in Computer Tutoring", Computer Education, Vol. 13, # 1: 95-100.
- Epstein K., Hillegeist E. (1990): "Intelligent Instructional Systems: Teachers and Computer-Based Intelligent Tutoring Systems", Educational Technology, November: 13-19.
- Fischetti E. , Gisolfi A. (1990): "From Computer-Aided Instruction to Intelligent Tutoring Systems", Educational Technology, August: 7-17.
- Flouris G. (1989): "The Use of an Instructional Design Model for Increasing Computer Effectiveness", Educational Technology, November: 7 - 13.
- Gisolfi A., Mocerardi G. (1990): "An Intelligent Tutoring System for the Factorization of Algebraic Expressions in Smalltalk/V", EFTI 27,4: 406 - 413.
- Hajovy H., Christensen D. (1987): "Intelligent Computer Assisted Instruction: The Next Generation", Educational Technology, May: 9 - 14.
- Huerta J. (1982): Organización lógica de las experiencias de aprendizaje, Ed. Trillas, México.
- Lianjing H. (1990): A Tool for Developing Intelligent Tutoring Systems", Educational Technology, September: 54-55.
- Lianjing H., Taotao Hu (1991): "A Uniform Student Model for Intelligent Tutoring Systems: Declarative and Procedural Aspects", Educational Technology, November.
- Livas I. (1988): Análisis e Interpretación de los Resultados de la Evaluación Educativa, Trillas, México.
- Luger G. , Stubblefield W. (1989): Artificial Intelligence and the Design of Expert Systems, The Benjamin/Cummings Publishing Company Inc., USA.
- Merril M. (1988): "The role Of Tutorial an Experiential Models in Intelligent Tutoring Systems", Educational Technology, July: 7-13.

- Merrill M. (1990): "Second Generation Instructional Design (II2)". Educational Technology, Febrero: 7-14
- Metcalf K. (1989): "Professional Skill Development in Teacher Training: Toward a More Effective Model". Educational Technology, September: 35-39.
- Mishkoff H. (1986): Understanding Artificial Intelligence. Texas Instruments Learning Center, USA.
- Nilsson J. (1971): Problems Solving Methods in Artificial Intelligence. Mc Graw-Hill, U.S.A.
- Rich E. (1983): Artificial Intelligence. Ed Mac Graw Hill, U.S.A.
- Rosenberg R. (1987): "The Use of an Instructional Design Model for Increasing Computer Effectiveness". Educational Technology, January: 14 - 21.
- Russell J., Blake B. (1988): "Formative and Summative Evaluation of Instructional Products and Learners". Educational Technology, September: 22 - 28.
- Sera J. (1986): Análisis y Diseño de Sistemas de Información, Mc Graw Hill, U.S.A.
- Shing J. (1987): "Individualizing Instruction Through ICAI: A perspective". Educational Technology, March: 7-15.
- Sleeman D., Brown J. (1982): Intelligent Tutoring Systems. Academic Press, England.
- Sommerville I. (1988): Ingeniería de Software, Addison-Wesley Iberoamericana, México.
- Twitchell D. (1991): "Robert M. Gagne and M. David Merrill in Conversation: ICAI, Computers in the Schools, an Instructional Design Expert System". Educational Technology, January: 34-40.
- Vassileva J. (1990): "An architecture and Methodology for Creating a Domain-Independent, Plan Based Intelligent Tutoring System". ETI 27,4: 386 - 397.
- Wenger E. (1987): Artificial Intelligence and Tutoring Systems. Morgan Kaufmann Publisher, Inc., U.S.A.
- Winston P. (1979): Artificial Intelligence. Addison-Wesley Publishing Company, USA.