



**UNIVERSIDAD TECNOLÓGICA  
DE LA MIXTECA  
HUAJUAPAM DE LEÓN, OAXACA, MÉXICO**

---

---

**INGENIERÍA EN COMPUTACIÓN**

**DESARROLLO, IMPLANTACIÓN Y VALIDACIÓN  
DE UN ALGORITMO DE TOMA DE DECISIONES,  
BASADO EN UN MODELO DE ESTUDIANTE, PARA  
DEFINIR ESTRATEGIAS DE ENSEÑANZA EN  
SISTEMAS TUTORIALES INTELIGENTES**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN**

**P R E S E N T A**

**EDUARDO FLAVIO GUTIERREZ IBARRA**

PRIMAVERA 1998



*Universidad Tecnológica de la Mixteca*

Universidad Tecnológica de la Mixteca  
Huajuapam de León, Oaxaca, México

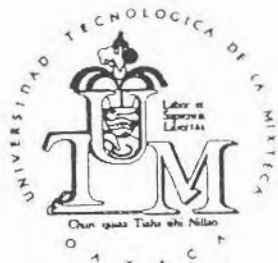
Ingeniería en Computación

**Desarrollo, Implantación y validación de un algoritmo de toma  
de Decisiones, basado en un Modelo de Estudiante, para Definir estrategias  
de enseñanza en Sistemas Tutoriales Inteligentes**

Tesis que para obtener el título de Ingeniero en Computación presenta  
Eduardo Flavio Gutiérrez Ibarra

Primavera 1998

U. T. M. 8384



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

VICE-RECTORIA ACADÉMICA

OFICIO No.0457/VAC/UTM/98.

C. EDUARDO FLAVIO GUTIERREZ IBARRA  
PRESENTE.

Después de haber revisado el trabajo que Usted presentó, y que le permita obtener el título de Ingeniero en Computación, que otorga la Universidad Tecnológica de la Mixteca.

Por este conducto, le comunico que está Usted autorizado para proceder a la impresión de la tesis intitulada "Desarrollo, implantación y validación de un algoritmo de toma de decisiones, basado en un modelo de estudiante para definir estrategias de enseñanza en sistemas tutoriales inteligentes"

Impresa, deberá entregar 10 ejemplares de la misma, a esta Vice-Rectoría el día Viernes 5 de Junio del presente a las 9:00 A.M, con el objeto de distribuirlas en las diversas áreas académicas de la Universidad.

Sin más por el momento, quedo de Usted.

Atentamente

"Labor et Sapientia, Libertas"  
Acatlima, Huajuapán de León, Oaxaca  
Mayo 28 de 1998.

EL VICE-RECTOR ACADÉMICO

ING. GERARDO GARCÍA HERNÁNDEZ



VICE-RECTORIA  
ACADÉMICA

U. T. M. 8354

C.c.p. Expediente.

Desarrollo, implantación y validación de un algoritmo de toma de decisiones, basado en un modelo de estudiante, para definir estrategias de enseñanza en sistemas tutoriales inteligentes

Aprobado:



---

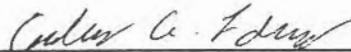
**Dr. Guillermo Romero Jiménez**  
Asesor de Tesis

Investigador del Instituto de Investigaciones Eléctricas



---

**Dr. Anatoly Koulinitch Semenovitch**  
Director de la División de posgrado  
Universidad Tecnológica de la Mixteca



---

**MC. Carlos Alberto Fernández y Fernández**  
Profesor asociado de la  
Universidad Tecnológica de la Mixteca



---

**Ing. Hugo Suárez Onofre**  
Director de la carrera de Ingeniería en Electrónica  
Universidad Tecnológica de la Mixteca

Dedicado a mis hermanos: Judith,  
Francisco y  
Patricia.

a mis padres: Graciela y Flavio  
y a mis abuelos: Candelaria, Raúl †, Clara † y Pedro.

## Agradecimientos

A:

Dr. Guillermo Romero por su amistad y la oportunidad de trabajar juntos en este proyecto.

Ing. Alfredo Báez por su importante participación en el desarrollo del prototipo y por ser un excelente amigo.

Los doctores Pablo de Buen y Francisco Múgica, por sus excelentes observaciones al proyecto.

El Instituto de Investigaciones Eléctricas por la infraestructura proporcionada durante el desarrollo de este proyecto.

Mis amigos becarios en el IIE: Agustín, Luis, Miguel Ramírez, Uc, Raúl, Adriana, Patricia, Isabel, José Manuel y Leonora por su apoyo profesional y amistad.

Mis amigos de la Unidad de Resultados de Simulación: Virginia, Yadira, Mary, Esperanza, Montoya, Jorge, Cesar, Erik, Miguel Rossano, Mario, Angeles, Cony, Marco y Arjona, por su apoyo profesional en todo momento.

La Universidad Tecnológica de la Mixteca, por las facilidades que me fueron proporcionadas para la elaboración de este proyecto y por el ímpetu profesional.

Mis amigos de carrera y generación: Atonaltzin, Moisés, Fabiola y Chema por que seguimos siendo amigos y ayudándonos siempre.

La familia Ortiz Ibarra: Miguel, Silvia, Eurídice y Amilcar por el apoyo que me brindaron al inicio de este proyecto y por ser excelentes amigos.

Las familias Ibarra Martínez: Raúl, Irma, Oscar y Alba; y Salvador Ibarra: Job, Irma, Itze y Jazel por su apoyo constante.

Karina Ibarra por su amistad y apoyo.

Dios por la vida, es hermosa.

Y de manera muy especial a Estela, por que gracias a tu amistad tengo el coraje y ánimo de enfrentar retos. Eres el resplandor que me abre al día.

## Resumen

En esta tesis se presenta un análisis de varios modelos de estudiante que están siendo utilizados en los diferentes Sistemas Tutoriales Inteligentes (STI) que se desarrollan actualmente. De estos se selecciona un Modelo de tipo superposición que sirve de retroalimentación al STI para proporcionarle información acerca del estudiante. A través de un algoritmo de toma de decisiones (Módulo Tutor), el STI determina las estrategias de enseñanza óptimas para completar el proceso de enseñanza - aprendizaje acerca de un tema particular (dominio de aplicación) enmarcado en el contexto de los Sistemas Eléctricos de Potencia.

Se describe además la implementación de un prototipo de STI basado en una plataforma Multimedia y en técnicas de Inteligencia Artificial.



Tema	Página
Introducción.....	1
Antecedentes.....	1
Planteamiento del problema.....	1
Objetivo de la Tesis.....	2
Alcance.....	2
Contenido de cada capítulo.....	2
Marco Teórico.....	3
2.1 Generalidades.....	3
2.2 Evolución.....	4
2.3 Arquitectura.....	4
2.4 Clasificación de STI.....	7
2.5 Métodos de Representación del Conocimiento de la Materia.....	9
2.5.1 Conocimiento del Área de Estudio.....	10
2.5.2 Conocimiento de los problemas a resolver.....	12
2.6 Modelo del Estudiante.....	14
2.7 Examinación y diagnóstico de conocimiento.....	15
2.8 Manejo del Proceso de Enseñanza.....	17
Modelado del Estudiante.....	26
3.1 Elementos principales en el Modelado del Estudiante.....	27
3.2 Clasificación de Modelos de Estudiante.....	28
3.2.1 Bandwidth.....	30
3.2.2 Tipo de conocimiento.....	31
3.3 Modelo del Estudiante.....	33
3.4 Algoritmo de Toma de decisiones basado en un Modelo del Estudiante.....	35
3.4.1 Estrategias de enseñanza.....	37
3.5 Inicio del Modelo del Estudiante.....	49
3.6 Ajuste del Modelo del Estudiante.....	55
3.6.1 Diferencia entre ajuste de primero en amplitud y primero en profundidad.....	57
3.7 Diseño del ME basado en la técnica orientada a objetos.....	59
3.8 Acceso a Bases de Datos.....	62
3.9 Contexto del Modelo del Estudiante.....	64
3.10 Diagrama de Flujo del Módulo Tutor en su parte administrativa.....	64
Arquitectura.....	68
4.1 Desglose de Conceptos del Dominio de Aplicación.....	69
4.2 Conformación del Modelo del estudiante.....	70
4.3 Diseño de la Agenda.....	72
4.4 Diseño Interfaz.....	74
4.5 Generación de Lecciones.....	76
4.6 Generación de Indicadores.....	77
4.7 Control del flujo en la Herramienta Multimedia.....	81
4.8 Codificación Agenda.....	84
4.9 Codificación Indicadores.....	85
4.10 Codificación del Módulo Tutor y Modelo del Estudiante.....	87
4.11 Acceso a Bases de Datos.....	90
4.12 Validación.....	100
4.12.1 Pruebas Locales a los Módulos que conforman el prototipo de STI.....	100
4.12.2 Evaluación del prototipo de STI frente a usuarios.....	102
Conclusiones y trabajos futuros.....	103
5.1 Conclusiones.....	103
5.2 Perspectivas y trabajos futuros.....	103
Referencias.....	104

# Índice

Apéndice A.....	107
Apéndice B.....	123
Apéndice C.....	132
Apéndice D.....	137

# Capítulo uno

## Introducción

### ***Antecedentes***

Las computadoras se han ido involucrando en diferentes áreas de las actividades humanas, la educación es una de ellas. El uso de las computadoras en el campo de la educación es posible debido a varios factores: el desarrollo de la tecnología ha permitido reducir costos y es muy común que muchas personas tengan acceso a una computadora personal. Ese mismo desarrollo permite también el cómputo de grandes cantidades de información en muy poco tiempo, además la evolución de las interfaces gráficas permite la fácil utilización de esos equipos.

La educación es un factor importante en el desarrollo de la sociedad. Esta ha utilizado diferentes medio para transmitir conocimiento, ( lenguaje, libros, radio, televisión, cine y más recientemente las computadoras). Desde hace algunos años se introdujeron en el mercado sistemas basados en computadora que permiten la transmisión de conocimiento. Se tiene actualmente una gran variedad de sistemas dedicados a la educación que van desde los libros electrónicos hasta complejos sistemas basados en simulación para el entrenamiento [13].

El uso de las computadoras apoya y mejora en algunos casos el proceso de enseñanza - aprendizaje, proceso fundamental en la educación, ya que permite la transmisión de conocimientos en varios modos y formatos: voz, vídeo, texto, animación, simulación. Además, los desarrollos en otras especialidades como la pedagogía, la inteligencia artificial, Multimedia, y las ciencias cognoscitivas entre otras, permiten el desarrollo de sistemas complejos capaces de mejorar el proceso de enseñanza-aprendizaje.

Dentro de los diferentes sistemas basados en computadora para la educación se encuentran los Sistemas Tutoriales Inteligentes (STI) los cuales han venido desarrollándose desde principios de los años 70's. Actualmente el desarrollo de estos sistemas está cobrando un gran auge debido básicamente al rápido desarrollo de la tecnología que permite que cada vez más personas puedan desarrollar este tipo de sistemas.

Un STI es un programa de computadora que enseña a un usuario (estudiante) de una manera inteligente. Estos sistemas conducen al estudiante a obtener información apropiada de acuerdo a las características que el mismo posee respecto a determinado tema. Por lo cual el sistema infiere el nivel de dominio del tema que tiene el estudiante que esta frente a el.

En la Unidad de Resultados de Simulación (URS) del Instituto de Investigaciones Eléctricas (IIE) se han venido desarrollando Sistemas Tutoriales (ST) enfocados a mejorar el proceso de enseñanza aprendizaje de temas teóricos relacionados con los simuladores para el adiestramiento de centrales eléctricas [2], [6]. En estos ST básicamente se explota las ventajas de los sistemas Multimedia y de la estrategias de enseñanza denominada "voltar la pagina". Así, estos ST carecen de mecanismos que controlen y guíen el proceso de enseñanza, es decir no llegan a ser STI en el sentido estricto y amplio del termino.

### ***Planteamiento del problema***

Ante la experiencia de los ST mencionados en el párrafo anterior, se plantea la necesidad de desarrollar un STI que contenga al menos los cuatro módulos indispensables en todo STI. Uno de los problemas principales en el diseño y construcción de un STI radica en la concepción de tanto el módulo del Modelo del Estudiante como en el Módulo Tutor. Los módulos restantes que en general conforman un STI son los modulos de Interfaz y dominio de aplicación.

# Introducción

Finalmente el reto planteado más importante consiste en diseñar y validar un prototipo de STI que eso cumpla con la función básica de guiar, controlar y completar el proceso de enseñanza - aprendizaje relacionado con el tema de los Sistemas Electricos de Potencia, tomando en cuenta en utilizar las herramientas computacionales de la URS.

## ***Objetivo de la Tesis***

Desarrollar, Implantar y validar un algoritmo de toma de decisiones que se basa en un Modelo estudiante, para definir estrategias de enseñanza en Sistemas Tutoriales Inteligentes

## ***Alcance***

El alcance esperado del prototipo consiste esencialmente en la verificación de la operabilidad, funcionalidad y consistencia de los algoritmos y métodos presentados en esta tesis. Dichos algoritmos se refieren básicamente al Modelo del Estudiante y Módulo Tutor

Así, el alcance contemplado para el prototipo de STI permitirá sentar la bases del desarrollo de STI con amplias perspectivas de comercialización que la Unidad de Resultados de Simulación del IIE pretende.

## ***Contenido de cada capítulo***

En el segundo capítulo se hace una revisión general de lo que es y comprende un Sistema Tutorial Inteligente (STI) como un marco teórico que define sus componentes y sus características. Asimismo, se hace una revisión histórica de su evolución, una clasificación de los Modelos de estudiante vigentes, exploración de métodos de diagnóstico y finalmente se explora con gran detalle las diferentes formas de llevar a cabo la administración de la enseñanza (Módulo Tutor).

En el tercer capítulo se propone el diseño de un STI. Aquí se hace énfasis en la construcción del Modelo del Estudiante y el Modulo Tutor con el objetivo de implantar un prototipo. Además se proponen las decisiones y estrategias de enseñanza que sigue el prototipo de STI. Finalmente, se expone el diseño de otras características inherentes en sistemas de este tipo.

En el cuarto capítulo se describe como se lleva a cabo la implementación del prototipo con una herramienta Multimedia y un lenguaje de programación orientada a objetos (OOP), la validación del algoritmo y la del prototipo.

Finalmente se presentan las conclusiones del presente trabajo y los trabajos futuros recomendados.

### 2.1 Generalidades

Los Sistemas Tutoriales Inteligentes (STI) han venido desarrollándose desde varias décadas atrás. Cada uno de los diferentes prototipos se han desarrollado con fines de investigación, es decir, no se tiene una metodología rigurosa para construir un STI. Los proyectos cubren diferentes tópicos a enseñar y van desde la resta hasta la enseñanza de lenguajes de programación. Cada uno de los proyectos en un inicio no seguían un patrón definido. Uno de los primeros intentos hacia una normalización fue la gran diferencia que existe en enseñar los diferentes tópicos que involucran el conocimiento humano. También se descubre la gran gama de formas que existen de enseñar una misma materia, es decir cada persona tiene diferentes puntos de vista acerca de como enseñar y los aplica.

Hoy en día se ha logrado distinguir cuatro componentes básicos de un STI, los cuales son: *Base de Conocimientos, Interfaz, Modelo del Estudiante, Modulo de Tutor*. Cada uno de estos roles siguen un patrón común el la enseñanza, es decir de ahí mismo se pueden generar y diseñar todos los algoritmos posibles para diseñar un STI. Con el transcurso de los años se ha descubierto la gran complejidad que existe en la construcción de un STI, debido a la gran cantidad de trabajo que debe hacerse en diferentes disciplinas y conjuntarlas. Las metodologías de programación, la pedagogía y la Inteligencia Artificial proponen algunos métodos para la conformación de un STI. La pedagogía muestra la manera en que se debe enseñar. Esta ha implementado métodos para enseñar efectivamente en lo que se refiere al tipo de enseñanza uno a uno. Para cualquier tutor (humano), es incierto en primera instancia enseñar efectivamente (inteligentemente) a un alumno que no conoce, puesto que es imposible conocer el estado cognoscitivo que tiene. Bajo este punto de vista se tiene que diseñar un método para enseñar a cualquier estudiante que se le presente. Los avances en lo que se refiere a métodos pedagógicos han ido evolucionando y mejorando. El gran reto, consiste en modelar esa experiencia en un sistema de computo. Un Sistema Tutorial Inteligente debe inferir el nivel cognoscitivo que tiene un alumno, con el fin de determinar su comportamiento.

Por otro lado, la IA en si misma ha navegado buscando métodos de representación del comportamiento humano (dado que un STI seria el modelo de un Tutor con experiencia). La IA tiene grandes deficiencias debido a la poca información que existe acerca de como los humanos se comportan inteligentemente o de que es en si la Inteligencia. La IA también ha generado excelentes resultados al tratar de modelar los procesos humanos que son "inteligentes".

Grandes avances se han generado en el ámbito de los STI, la gran mayoría de los proyectos que están en investigación y los ya terminados, han seguido abstracciones de un modelo general en aras de conseguir un comportamiento inteligente relativo a los objetivos que persiguen. Muchos investigadores han seguido y formulando diferentes metodologías para desarrollar un STI. Por lo cual es fácil encontrar una amplia gama de metodologías para resolver el problema de enseñar inteligentemente. La gran diversidad de metodologías se debe a las diferentes materias, objetivos, formas de enseñar, tiempo y herramientas computacionales usadas. Cabe decir que las diferentes metodologías adoptadas, aun se ubican dentro de un marco general, es decir, dada la complejidad se puede en determinado momento decidir unir módulos, como por ejemplo: tener un experto tanto en la materia a enseñar como en pedagogía. Muchas simplificaciones y uniones de los componentes de un STI se hacen con el fin de lograr un mejor desempeño del este.

## 2.2 Evolución

Los primeros trabajos en educación y computadoras datan de inicios de los 60. En ese entonces quedaron en evidencia las imperfecciones e inadaptabilidad del hardware y software que estaban disponibles en esa época. A finales de esa década se crearon los primeros sistemas basados en computadora, PLATO y COURSEWRITER [1]. Estos eran básicamente paquetes de software especializado con propósitos de diseño y mantenimiento de programas de enseñanza, que eran denominados Cursos de Estudio Basados en Computadora. La evolución de los Sistemas Tutoriales basados en computadora caminó a la par de del desarrollo y perfeccionamiento del hardware. Con la aparición de las computadoras personales con monitores a color y los dispositivos de almacenamiento de datos visuales, es posible ahora crear un medio ambiente psicológicamente agradable para la enseñanza. El trabajo que se ha realizado con el propósito de crear medios ambientes adecuados para la enseñanza, ha conducido al diseño de lo que se actualmente se conoce como Sistemas Tutoriales Inteligentes. Mientras que los tradicionales Cursos de Estudio Basados en Computadora se asumen como el resultado de una teoría de comportamiento de enseñanza, los Sistemas Tutoriales Inteligentes se asocian con teorías psicológicas, principalmente con las teorías de la psicología cognoscitiva. Los primeros estudios en esta área fueron hechos al principio de 1970 y están relacionados con el concepto de programas de enseñanza generativos. Los primeros intentos se enfocaron en generar problemas en base a las condiciones del estudiante y la solución automática de los problemas generados. El siguiente paso ocurrió a mediados de la década de los 70 cuando se buscaron métodos para representar el conocimiento adquirido por el alumno (Modelo del Estudiante) y también de estrategias de enseñanza. En los 80 el desarrollo de los STI fue influenciado en gran medida por la ingeniería del conocimiento y el diseño de sistemas expertos. En este periodo, fue propuesta una clasificación de los STI y se hicieron intentos para el desarrollo de una tecnología para el diseño de STI en áreas específicas [27]. Por otro lado, se desarrollaban interfaces hombre máquina gráficas. Como un todo, el proceso de generalización y extensión de la experiencia adquirida en el desarrollo de STI ocurrió junto con la transición de sistemas experimentales a comerciales, junto con el desarrollo de herramientas para construirlos.

En los apartados siguientes, se presenta una arquitectura general de Sistemas Tutoriales Inteligentes y se describen los métodos básicos de construcción de cada uno de los componentes asociados. Se hace énfasis en los diferentes métodos de manejo del proceso de enseñanza. Se incluye un repaso general sobre el Modelo del Estudiante. Asimismo, en la tesis existe un capítulo donde se habla de manera más amplia de este tema.

## 2.3 Arquitectura

Dentro de la gran variedad de estrategias para el diseño de un STI, se puede distinguir un modelo general que permita construir un STI. Pero primero se hace énfasis en los diferentes objetivos pedagógicos que puede perseguir un STI. Cada uno de ellos determina dentro del conjunto de STI un subconjunto de los mismos. Dentro del Universo de STI existen varios parámetros que determinan las herramientas que se van a usar para su construcción (técnicas Computacionales, lenguajes de programación, hardware, software de base) y las posibles abstracciones de los mismos. Uno de esos parámetros es el objetivo pedagógico, el cual comprende los siguientes [22]:

# Marco Teórico

*Demostración de Material Instructivo.* Al alumno se le presentan textos, gráficos, audio y material visual en un orden definido por el sistema. Los sistemas de este tipo se denominan sistemas demostrativos e incluyen lecturas computarizadas y libros electrónicos.

*Diagnostico y prueba.* El STI muestra su inteligencia al proporcionar medios por los cuales puede inferir los tópicos que el alumno conoce y solo presentar aquellos que no son dominados. En estos se hace gran uso de herramientas Multimedia para presentar información. Al alumno se le presentan exámenes para discernir ciertos indicadores, en particular, su dominio en la materia o el grado en que ha asimilado habilidades y hábitos que ha obtenido previamente.

*Ejercicios.* Al estudiante se le provee con conocimiento, habilidades y hábitos necesarios para realizar una actividad en particular, se requiere el medio necesario para impartir y reforzar los hábitos y habilidades requeridas. Sus características son las siguientes:

- a) la presencia de un modelo formal y audiovisual del área que se estudia.
- b) conocimiento explícito de los objetivos funcionales del sistema.
- c) un procedimiento para examinar la actividad del estudiante.
- d) la presencia de límites de tiempo en el cual el alumno debe conseguir los objetivos que se le proponen.

*Enseñanza* El estudiante obtiene conocimiento, hábitos y habilidades en algún área en particular bajo el control de un programa de enseñanza: aquí el programa de enseñanza asume todas las funciones de un maestro: organiza la presentación de material instructivo, examina el dominio del alumno en el material, y diagnostica los errores que comete el alumno. Esta clase de programas se caracterizan por:

- a) la existencia de un objetivo de enseñanza.
- b) la implementación de un método particular de enseñanza en busca de cumplir el objetivo requerido y un claro estilo de interacción con el usuario y
- c) Una solución comprensible de problemas que surgen en el proceso de enseñanza, control y diagnóstico.

Los Programas de Enseñanza han sido desarrollados por medio de técnicas tradicionales basadas en computadora de tipo demostrativo, los cuales o dan al estudiante una relativa libertad, o poseen una rígida estructura de control. En contraste, los STI proveen al estudiante con cierta libertad, haciéndoles posible seleccionar tópicos instructivos y problemas, hacer preguntas acerca de lo que esta interesado, obtener respuestas a estas preguntas y la obtención de estimaciones de su conocimiento y habilidades. Esta distinción se basa en el hecho de que en los STI el significado contenido en el área de estudio, y el conocimiento del manejo del proceso de enseñanza y el estado del conocimiento del alumno están representados de forma explícita.

El STI se define como un sistema programado que implementa objetivos pedagógicos particulares bajo las bases del conocimiento que tienen los expertos en alguna área de estudio, en el diagnóstico del conocimiento del estudiante y en control sobre el proceso de enseñanza.

Se identifican los problemas básicos que surgen en el diseño de un STI;

- Representación y desarrollo del conocimiento en el área de estudio
- El desarrollo de métodos de examinar y diagnosticar los errores del estudiante
- El desarrollo de métodos de manejo del proceso de enseñanza

La arquitectura de un STI se basa en el siguiente modelo de proceso de enseñanza. Existe un objetivo pedagógico, expresado en términos de las características actuales del estudiante.

La siguiente secuencia de actividades se repite hasta que el objetivo se cumple:

# Marco Teórico

- Se genera un problema recurrente bajo las bases del estado actual del estudiante y la técnica de enseñanza (Un problema se entiende por cualquier pieza de información que requiera acciones de respuesta por parte del estudiante).
- La respuesta del estudiante se compara con la solución estándar y, basándose en cualquier diferencia encontrada, se genera un diagnóstico del error del estudiante.
- Bajo los resultados del diagnóstico las características del estudiante se corrigen.

De acuerdo con este modelo del proceso de enseñanza un STI se puede considerar como un conjunto de tres sistemas independientes.

- Un sistema diseñado para resolver problemas en el área particular de estudio (Base de conocimientos).
- Un sistema diseñado para diagnosticar los errores del estudiante (Modelado del Estudiante).
- Un sistema diseñado para planear el manejo del proceso de enseñanza (Módulo Tutor).

Un sistema para la solución de problemas se diseña para desarrollar una solución estándar del problema. Este componente de un STI puede consistir de lo siguiente:

- a) Un sistema construido bajo las bases del conocimiento que resuelve problemas bajo la particular área de estudio.
- b) Un solucionador de problemas en una área fuertemente formalizada del conocimiento
- c) Programas para la extracción o selección de una solución de un problema en particular de un conjunto finito (posiblemente grande) de soluciones, si es que la enseñanza se conduce en base a una colección de problemas previamente preparados.

Un sistema en el diagnóstico de errores se diseña para descubrir cualquier noción incorrecta que posea el estudiante en el área de estudio, ya que su respuesta se compara con la respuesta estándar. Para construirse se deducen indicadores bajo las bases de diferencias en las respuestas del estudiante contra aquellas que el sistema da para la solución de problemas, y cambios en las características actuales del estudiante (Modelo del Estudiante) constituyen los resultados.

Un sistema para el manejo del proceso de enseñanza constituye un sistema para planear bajo las restricciones impuestas por el material instructivo disponible. Tal sistema formaliza el conocimiento de enseñanza

La interacción entre un STI y el proceso de enseñanza se da de la siguiente manera. De acuerdo con el objetivo actual, el sistema en el proceso de enseñanza genera el proyecto para el estudiante, el cual es transmitido simultáneamente al sistema en la solución de problemas. El sistema de diagnóstico de errores entonces compara la solución obtenida por el estudiante y la generada por el sistema en la solución de problemas y, basándose en las diferencias encontradas, trata de establecer que tipo de nociones incorrectas tiene el estudiante en el área de estudio que pudieron haber llevado al estudiante a tales discrepancias. Como resultado del diagnóstico la representación del estudiante formada en el STI se refleja en un cambio en el Modelo del Estudiante y el proceso de enseñanza asume nuevamente el control, refinando el objetivo actual. La interacción con el usuario se lleva a cabo a través de una interfaz que contiene texto, gráficos, herramientas de entrada y salida, procesadores lingüísticos, etc.

Las siguientes bases de conocimientos resaltan como parte de la estructura de un STI: Una base de conocimiento instructivo para una área en particular, un modelo de estudiante, una base de conocimientos para posibles errores y una base de conocimientos del proceso de enseñanza.



# Marco Teórico

La base de conocimientos instructivos no solo describe los conceptos básicos y métodos básicos del proceso de solución de problemas en el área en particular, también contiene la definición de conceptos, la descripción de métodos, ejemplos, ejercicios y problemas. Al contrario de la base de conocimientos en la solución de problemas, la base de conocimientos instructivos debe explícitamente reflejar la estructura del dominio del conocimiento y del conocimiento estratégico que conlleva a métodos para la solución de problemas. Por otro lado, la base de conocimientos instructivos se puede considerar como una representación de las restricciones dentro de la estructura bajo la cual el proceso de enseñanza planea precisamente ese proceso.

El Modelo del Estudiante contiene información acerca del estado cognoscitivo del estudiante, tanto características generales y aquellas que reflejan el dominio del estudiante en el material instructivo.

La base de conocimientos de errores contiene un catálogo de errores potenciales del estudiante y reglas que proponen y verifican las hipótesis que confirman las nociones incorrectas que tiene el estudiante que lo llevaron a determinado error, bajo las bases de las diferencias entre las soluciones propuestas por el estudiante y aquellas propuestas por la base de conocimientos en la solución de problemas, así como también el estado actual del Modelo del Estudiante.

La base de conocimiento del proceso de enseñanza contiene conocimiento acerca de la planeación y organización del proceso de enseñanza y técnicas generales y especiales de enseñanza.

## 2.4 Clasificación de STI

Una clasificación de STI se establece de acuerdo con sus objetivos de operación. Se irá describiendo sucesivamente STI de consulta, diagnóstico, manejo y de apoyo, ya que este orden refleja la complejidad promedio del desarrollo del sistema.

El STI de *consulta* se diseña para proveer asistencia al estudiante por medio de la presentación de información en respuesta a peticiones por parte del estudiante o a través de la solución de problemas propuestos por el estudiante seguidas de una explicación (si es que el estudiante lo pide) una vez que la información o solución se obtiene. Un STI de consulta consiste de una referencia de información o un medio ambiente de solución y sistema de explicación, también si es posible un modelo de estudiante. El Modelo del Estudiante se usa para seleccionar el modo de interacción con el estudiante y determinar su actividad (por ejemplo, que tipos de problemas puede resolver por medio del sistema o que tipo de conceptos que pertenecen al área de estudio en la cual el solicita información). Si el sistema se basa en una referencia de información, el sistema se denomina un sistema de preguntas y repuestas, y que tan "inteligente" es se determina por la amplitud de su lenguaje de preguntas.

Un STI de *diagnóstico* se enfoca a determinar las nociones incorrectas acerca del área de estudio que han conducido al estudiante a cometer errores en el curso de la solución de un problema en particular. Un STI de diagnóstico consiste de una interfaz, un sistema en la solución de problemas, un sistema en el diagnóstico de errores, y un Modelo del Estudiante. Desde el punto de vista de su origen un STI de diagnóstico se puede considerar como un complemento de un STI de consulta ya que se considera:

- 1) una extensión de la interfaz con facilidades por las cuales el estudiante puede hacer uso de la computadora para resolver problemas por si mismo (un rastreo del sistema del uso de estas facilidades hacen posible que se determine una "ruta" o plan del estudiante y comparar su solución con el plan de solución del sistema)
- 2) el desarrollo de un Modelo del Estudiante que establece el conocimiento del estudiante y habilidades en la solución de problemas de un determinado tipo, y

# Marco Teórico

- 3) el desarrollo e implementación de métodos de diagnóstico de errores del estudiante en la forma de un sistema. Frecuentemente los sistemas de este tipo se conocen como Ejercitadores Inteligentes o Sistema Expertos en Ejercicios, ya que se usan para proveer al estudiante práctica en la solución de problemas, ya sea que una secuencia de problemas se genere por el sistema o por el mismo estudiante.

El STI de *manejo* se enfoca a guiar la actividad cognoscitiva del alumno. Un STI de manejo constituye una extensión de un STI de diagnóstico, en base a que contiene además conocimiento acerca de los objetivos de operación del sistema presentes en el material instructivo y las estrategias de enseñanza. Hay que hacer énfasis en la enseñanza de conceptos y habilidades (hábitos), así, los sistemas son conocidos como Programas de Enseñanza y Programas de Ejercicios. La diferencia principal entre Programas de Ejercicios y Programas de enseñanza es que en los últimos existe la suposición de proveer al estudiante con material nuevo.

Un STI de *apoyo* se enfoca a rastrear la actividad del usuario en determinados sistemas (herramientas) y proveer al usuario con asistencia ante acciones incorrectas o irracionales que son detectadas. Un STI de apoyo en efecto "ve sobre el hombro" del usuario en la pantalla y trata de comprender lo que el usuario hace, evalúa como está desarrollando actividades en particular, decide cuando ayudarlo, y si es posible, como debe ayudarlo. Un STI de apoyo contiene todos los componentes de un STI, pero, al contrario de los STI de manejo:

- 1) no conoce el objetivo de la actividad del usuario y debe predecirla, por lo tanto aumentan las funciones diagnóstico del sistema encargado del diagnóstico de errores, y
- 2) es menos interactivo ya que evita distraer al usuario del problema que está resolviendo.

Desde el punto de vista que se viene desarrollando, un STI de apoyo se considera una extensión de un STI de manejo en base a:

- 1) una extensión de las funciones del sistema en el diagnóstico de errores ya que predice los objetivos en base a la actividad del usuario
- 2) la correlación del material instructivo con las situaciones de error, y
- 3) el desarrollo de un sistema para el proceso de enseñanza que contiene estrategias para asistir a los usuarios cada vez que ocurran acciones incorrectas o irracionales.

La conclusión a la que se llega es que los dos últimos STI son los más complicados de construir.

A continuación se presenta una tabla en donde se resumen los tipos de STI. Así como sus características en base a su estructura y propósito.

Tipo de STI	Estructura	Propósito
Consulta	Explicaciones en un medio ambiente de aprendizaje	Consulta de soluciones de problemas y en la recuperación de información técnica.
Diagnóstico	Solucionador de Problemas Diagnóstico Modelo del Estudiante	Diagnóstico de errores en la solución de problemas
Manejo	Solucionador de Problemas. Diagnóstico. Administración del Proceso de enseñanza.	Conceptos de enseñanza y habilidades en base a la simulación del conocimiento del alumno.
Apoyo	Herramientas del Sistema. Diagnóstico. Administración del proceso de enseñanza. Modelo del Estudiante	Rastreo del comportamiento del alumno e intervención del tutor en respuesta a acciones irracionales o incorrectas

**Tabla 1**

Clasificación de Sistemas Tutoriales Inteligentes de acuerdo a su arquitectura y propósito.

## 2.5 Métodos de Representación del Conocimiento de la Materia

En un STI el conocimiento acerca de un área en particular esta contenido en la base de conocimiento instructivo y en el sistema experto en la solución de problemas. Desde el punto de vista de los objetivos de operación de un STI, las principales restricciones impuestas por este tipo de conocimiento esta en la forma en que el sistema explota sus características, ya que:

- 1) debe proveer información instructiva y de la estructura del área de estudio
- 2) debe dar información acerca de los método de enseñanza, y
- 3) datos acerca de las propiedades de la información instructiva en si misma, necesaria para planear el proceso de enseñanza.

Se distinguen dos tendencias en el proceso de enseñanza: la enseñanza de conceptos y el desarrollo de habilidades (hábitos).

En la enseñanza de conceptos el énfasis esta en la presentación y explicación de material nuevo tanto conceptual y correcto, incluyendo: la definición de conceptos y la descripción de sus propiedades y relaciones, descripción de procesos tanto general y ampliado que ajustan datos cualitativos y cuantitativos en las características de objetos y procesos, evaluación de las propiedades de los conceptos.

En el desarrollo de habilidades, el énfasis principal esta en la creación de problemas repetitivos y gradualmente situaciones más complicadas que requieran la ejecución por parte del estudiante de (una o más ) acción definitiva. Aquí se asume que el estudiante ya ha dominado cierto volumen de conocimiento y habilidades necesarias para identificar las acciones y determinar su orden.

Ambas partes del proceso de enseñanza están presente en un proceso particular de enseñanza o en un programa de instrucción, pero en proporciones diferentes. Por lo tanto, conocimiento acerca de conceptos y el conocimiento acerca del desarrollo de habilidades estarán presentes en el programa de enseñanza. Se hará referencia al primero como conocimiento del área de estudio y al ultimo como conocimiento de los problemas a resolver. Nótese que la utilidad de tal división del conocimiento se deriva del hecho que el conocimiento del área de estudio refleja representaciones científicas contemporáneas del sistema de conceptos, en la mayoría de las disciplinas académicas es una componente constante y conservadora, mientras que el conocimiento de los problemas a resolver constituye una componente más variable cuyo contenido puede cambiar como una función de los objetivos instructivos actuales. Por otro lado, el conocimiento del área de estudio describe su estructura y contenido (semántico), mientras que el conocimiento de los problemas a resolver, junto con una descripción de semántica (algoritmos de solución de problemas), contienen información pragmática (situaciones de problemas y heurísticas).

## 2.5.1 Conocimiento del Área de Estudio

El Conocimiento del Area de Estudio contiene una descripción de los conceptos del área en cuestión y las relaciones entre estos. Se hace distinción entre el esquema conceptual de la base de conocimientos, la cual contiene conocimiento de la estructura de la base (conocimiento intencional), y el contenido de la base de conocimientos. El primer tipo de conocimiento se usa para acceder al ultimo, y este ultimo es a la vez procesado por parte del estudiante y/o del programa de enseñanza. Sin mencionar los métodos formales para describir modelos conceptuales de conocimiento, el conocimiento intencional puede ser representado por una jerarquía de clases y/o instancias de entidades (conceptos) relacionados "uno a uno", "uno a varios" etc.

Por ejemplo, para una área de estudios se distinguen los siguientes componentes de la base de conocimientos.

- Clases con el propósito de abstraer descripciones de conceptos y procesos
- Relaciones entre clases que definen la estructura del modelo del área de estudio
- Clases escalares que representan los tipos de atributos de las entidades (por ejemplo, número, línea, texto, imagen, imagen audible, programa).
- Atributos escalares de una entidad que asociada con las características de los objetos y procesos distinguen clases escalares (por ejemplo, con el atributo "lado" del concepto "triángulo" el tipo escalar "número").
- Instancias de entidades que representan objetos en particular o procesos, (por ejemplo, objetos o procesos con atributos con valores específicas).
- Relaciones entre entidades que imponen restricciones adicionales sobre los objetos o procesos en la descripción particular de una situación.

De manera general, el conocimiento de una área en particular se puede representar en la forma de una red (grafo) de conceptos cuyos nodos constituyen clases y/o instancias, y cuyos arcos constituyen las relaciones entre de estos. En adelante, el termino "concepto" se usa como sinónimo de "entidad" y "clases de entidades".

Se da una mejor descripción de un concepto en la forma de una estructura genérica si el concepto representa una clase de entidades, o en la forma de una instancia si el concepto representa una entidad.

La estructura de un concepto debe contener los siguientes componentes:

- Representación formal del concepto, en la forma de un par que consiste de "nombre del atributo - tipo " o "nombre de atributo - valor"

# Marco Teórico

- Representación formal de las propiedades de un concepto en la forma de reglas de producción o procedimientos adjuntos.
- Representación de las relaciones entre el concepto y otros conceptos.
- Correlación con el concepto de material instructivo que consiste en la definición del concepto y su explicación o descripción, ilustración, etc., junto con conocimiento acerca de métodos de visualización de la información instructiva.
- Descripción de las características del concepto, por ejemplo, "la importancia" o "frecuencia de ocurrencia" del concepto.

La flexibilidad de trabajo con una base de conocimientos se determina por la abundancia de relaciones existentes entre los conceptos. Se distingue entre relaciones estructurales y semánticas. Las relaciones estructurales incluyen "es parte de", "posee a", "es una subespecie de", "consiste de", etc. Las relaciones semánticas que son de significado pedagógico incluyen relaciones como "generalización", "especialización", "analogía", "refinamiento", "simplificación", "desviación" y "corrección". A continuación se presentan definiciones para estas relaciones semánticas [22].

$C'$  es una *generalización* de  $C$  si la definición de  $C'$  se obtiene de la definición de  $C$  por medio del remplazo de una constante por una variable en cualquier término gobernado por un cuantificador universal. *Especialización* es el inverso de la relación *generalización*.

$C'$  es un *refinamiento* de  $C$  si  $C'$  contiene propiedades adicionales aplicables al subconjunto de datos usados en  $C$ . *Simplificación* es el inverso de la relación *refinamiento*.

$C'$  es una *analogía* de  $C$  si un mapeo de las constantes de la definición de  $C'$  en las constantes de la definición de  $C$  existe.

$C'$  es una *desviación* de  $C$  si  $C'$  tiene el mismo objetivo que  $C$ , pero no conduce al éxito. *Corrección* es el inverso de la relación *desviación*.

Considérese dos relaciones más generales, "descendiente" y "es elemental para conocer". Una relación semántica como "especialización" o "refinamiento" puede ser seleccionada como la relación "descendiente". El significado de la relación "descendiente" es que  $C'$  es descendiente de  $C$ , entonces  $C'$  posee una representación por medio de las propiedades de  $C$ . La relación inversa se conoce como "antecesor". Relaciones estructurales como "es parte de" o relaciones semánticas como "simplificación" pueden ser seleccionadas como la relación "es elemental para conocer". En concreto, un concepto  $C'$  es elemental para conocer  $C$  si la definición de  $C$  depende substancialmente de la definición de  $C'$ . Por ejemplo, los conceptos "segmento de línea" y "ángulo" son elementales para conocer el concepto triángulo. La relación inversa es "es vital para conocer".

En el diseño de una red de conceptos una de las relaciones se escoge como la "relación principal", la red de conceptos constituye una clasificación en cuyos niveles superiores existen conceptos más generales. Ejemplos de tales redes incluyen las clasificaciones biológicas de plantas y animales, la clasificación jerárquica de los elementos químicos, etc. Los niveles más altos de tales redes de conceptos corresponden (estructural o ideológicamente) a objetos más complejos. Los ejemplos incluyen una descripción jerárquica de un complejo dispositivo hecho por el hombre (un barco, una máquina automotriz, o una computadora) o una jerarquía de objetivos (acciones, habilidades) en la solución de problemas. La selección de la relación principal está gobernada por los objetivos funcionales del programa de estudios.

Desde el punto de vista de los usuarios (los desarrolladores del programa de enseñanza y los estudiantes) la base de conocimientos del área de estudio constituye una fuente de información. Que tan difícil es acceder a este recurso depende de su organización lógica y física, a la vez de su lenguaje de petición. Si el desarrollador puede directamente acceder tanto la representación formal de los conceptos y la información instructiva, y si se usa un lenguaje de

programación para llevar a cabo este acceso, el estudiante estará en disposición de acceder directamente solo a la información instructiva y requerirá un lenguaje de petición apropiado. En el desarrollo de un STI, todos los esfuerzos son encaminados hacia proveer al estudiante con toda la libertad posible, en particular, la libertad de acceder a información instructiva acerca de los conceptos de la base de conocimientos. Ya sea por medio de pulsar una tecla, el estudiante accede a información instructiva. El lenguaje de petición puede ser en la forma de una jerarquía de menús, lenguaje natural, hipertexto, o un lenguaje de petición gráfico. Se considera la organización de la información instructiva usando la noción de hipertexto.

Por hipertexto se considera una red de textos. Un texto, que puede ocupar el monitor por completo, contiene palabras o grupos de palabras identificadas por un color específico o fuente. Las palabras se seleccionan usando un dispositivo adecuado (el cursor o un ratón), y el usuario se mueve a otro texto. Se provee más flexibilidad por el hecho que el mismo texto puede correlacionarse con diferentes palabras y combinaciones de palabras. El conjunto de textos y relaciones forman un grafo asociativo (red). La noción de hipertexto es un componente esencial de una noción más general en la organización de sistemas de información, como Multimedia. De acuerdo con esta noción, un sistema de información debe hacer uso óptimo de la capacidad de todas las formas diferentes en las cuales la información se puede expresar, ya sea gráficos, texto, sonido, o información visual, utilización de modernas técnicas en el área de información y procesamiento (por ejemplo, discos vídeo láser). Un sistema Multimedia basado en hipertexto puede relacionarse a otros tipos de información y a programas ejecutables. Sin embargo, siguiendo la salida de información o después de ejecutar un programa, el sistema regresa al hipertexto que invocó la información o programa.

## 2.5.2 Conocimiento de los problemas a resolver

Aunque el conocimiento del área de estudio especifica su modelo general, el conocimiento de los problemas a resolver se expresa en la forma de series de modelos de situación con objetivos claros y un mecanismo de búsqueda de solución, el cual, como fue recalado anteriormente, puede basarse en respuestas preestablecidas, un solucionador de problemas o un sistema experto.

Por lo tanto el conocimiento de los problemas a resolver contiene un conjunto de problemas (preguntas) que pueden ser dadas en avance o generadas en el curso de operación de un STI por medio del uso de herramienta generadoras de problemas y mecanismos de solución de problemas que tiene el objetivo de una solución prototipo.

La descripción del problema se formula mejor en la forma de una estructura prototipo si el problema es paramétrico, en la forma de una instancia si el problema es constante (con la condición invariable), o en la forma de una estructura genérica si las condiciones del problema son generadas aleatoriamente.

La estructura de un problema debe contener los siguientes componentes:

- La descripción formal de los objetos que participan en la formulación de la situación del problema (en la condición del problema y sus objetivos) o en la forma del par "nombre del objeto - tipo" donde tipo es el nombre de una clase escalar o clase de entidades (de un concepto)
- La representación formal de las condición del problema y de sus objetivos tanto como relaciones adjuntas entre los objetos que están en la descripción de la situación del problema (modelo de la situación).
- Representación de las relaciones entre el problema dado y otros problemas, conceptos y habilidades.
- Descripción del plan de solución del problema (algoritmo).
- Análisis de la respuesta del estudiante (preliminar), que consiste en la verificación de la respuesta en términos explícitos, comparación con un prototipo y el desarrollo de indicadores para su análisis posterior.

# Marco Teórico

- Correlación con el problema de información instructiva que consiste de la condición del problema y el resultado requerido, una explicación del método de solución del problema, ilustraciones, así como también conocimiento de métodos de visualización de problemas.
- Descripción de las características del problema, así como la complejidad del problema o de su tipo (constante, paramétrico, genérico).

Por lo general se distinguen tres tipos de problemas:

- *Constante*, el problema contiene los valores de todos los objetos y atributos conocidos que están presentes en la descripción de la situación del problema.
- *paramétrico*, ciertos atributos de los objetos del problema se expresan por medio de parámetros y un problema constante se obtienen al evaluar los parámetros.
- *Genérico*, constituye un problema paramétrico junto con un mecanismo para generar los valores de los parámetros.

Aun más, los problemas genéricos - paramétricos son posibles, en los cuales algunos de los valores de los atributos se especifican en términos de parámetros y los otros son generados.

Los problemas paramétricos se usan cuando un conjunto permisible y restringido de valores de los parámetros existe, mientras que los problemas genéricos permite la extensión del número posible de ejercicios. Por lo tanto, los problemas paramétricos son mejor implementados en programas ilustrativos y de enseñanza, los problemas genéricos son mejor implementados en programas de ejercicio y pruebas.

En el caso más general, la generación de las condiciones del problema puede considerarse como el problema de diseñar un objeto que satisfaga ciertas restricciones y objetivos. Se distingue entre la generación de valores de parámetros y la generación de la estructura (condición) del problema, ya sea condicional (depende de ciertas restricciones) o incondicional. Los valores de los parámetros se generan bajo las bases de una distribución de probabilidad (uniforme, normal, binomial, etc.) usando generadores de números aleatorios. Si los valores no satisfacen las restricciones impuestas, el proceso de generación se repite hasta que las restricciones se satisfagan.

El método más general de generación de estructuras consiste en la aplicación de gramáticas generativas formales (determinísticas, no determinísticas, o probabilísticas) que especifican la descripción de todas las posibles estructuras de las condiciones del problema, el cual puede ser infinito o finito en número.

Por lo tanto el conjunto de problemas forma una base de conocimientos estructurada, en la cual el acceso a sus elementos es posible bajo las bases del nombre del problema, el valor numérico de la complejidad, el espectro, así como también bajo las bases de una combinación de espectro y complejidad.

Un resolvidor de problemas y un STI no solo obtendrán una solución correcta, si no también explicaran al estudiante como fue que se obtuvo esta respuesta (por ejemplo, presentaran un plan de solución del problema - algoritmo - junto con una validación de cada uno de los pasos que contiene el plan). Un plan para resolver el problema debe corresponder al nivel de dominio del estudiante. Más aun, un resolvidor de problemas debe ser capaz de ejecutar planes de solución de problemas (algoritmos) desarrollados por el estudiante. Un resolvidor de problemas que satisface todos estos requerimientos se dice que es transparente o articulado.

Los costos que incurren en el desarrollo de un sistema resolvidor de problemas transparente se reducen si se observan las siguientes condiciones:

# Marco Teórico

- 1) los objetivos del problema y las relaciones del problema a sub-objetivos (lo cual ya está expresado explícitamente en los lenguajes usados para representar el conocimiento, el cual envuelve una estrategia de inferencia de arriba hacia abajo)
- 2) conocimiento del área de estudio (modelo general) y conocimiento del problema (modelo de situación) se conservan separados del mecanismo de inferencia, haciendo posible explicar por separado la estrategia de solución del problema y conceptos del área de estudio al mismo tiempo que sirve de fundamento para construir el Modelo del Estudiante
- 3) conocimiento explicativo de los conceptos del área de estudio y conocimiento efectivo y proposiciones acerca de las propiedades de los datos, consumo de memoria, etc. de una solución propuesta (procedimiento de inferencia), la cual hace posible proveer una explicación más profunda y un modelo de las capacidades del usuario.

La construcción de un resolutor de problemas está entre los problemas centrales de la inteligencia Artificial.

## 2.6 Modelo del Estudiante

Por un Modelo del Estudiante se entiende por el conocimiento que posee un maestro (sistema instructivo) que concierne al estudiante y que se usa para organizar el proceso de enseñanza.

Esta definición permite dos interpretaciones. En la primera interpretación, un Modelo del Estudiante es un modelo del estado actual de conocimiento y habilidades de un estudiante en particular. En la segunda interpretación, un Modelo del Estudiante es un modelo "idealizado" del conocimiento del estudiante, que incluye conocimiento de una área de estudio, errores típicos, y mecanismos cognoscitivos.

Se usará primera interpretación como el Modelo del Estudiante a adoptar en el siguiente trabajo.

Un Modelo del Estudiante contiene conocimiento de los siguientes elementos:

- Características generales físicas y sociales del estudiante independientes del área de estudio (por ejemplo, datos psicológicos como sexo, edad, promedio de respuesta, habilidad de abstraer conocimiento, introversión contra extroversión, etc. Datos sociales, como nivel educativo, área de concentración de estudio, etc.).
- La relación entre el estudiante y el material instructivo (por ejemplo, que tan profunda y amplia es la comprensión del estudiante en el tema y/o que tan profunda y amplias son sus habilidades en el área de estudio en particular.
- Historia de la interacción entre el maestro (programa de enseñanza) y el estudiante.

El conocimiento preliminar del estudiante y el objetivo de enseñanza se formulan como restricciones en el estado inicial y final del Modelo del Estudiante.

En el caso más general el proceso de modelar al estudiante se realiza de este modo. Un Modelo del Estudiante inicial contiene las restricciones sobre el conocimiento inicial y habilidades del estudiante, y el maestro puede redefinir el Modelo del Estudiante mediante un diagnóstico preliminar del conocimiento que el alumno posee. La formación del Modelo del estudiante se lleva a cabo mediante una serie de pasos, empezando con la creación de un problema bajo las bases de un modelo general del área de estudio (un modelo de conocimiento y habilidades de un especialista en la materia), el cual, junto con la solución que el alumno provee y un análisis de la solución propuesta, genera un modelo situacional. Una forma especial (parcial) de modelo de estudiante la generalización inductiva de este modelo situacional, donde, más aun, el conocimiento de los errores que el estudiante hace cuando emplea una solución de un problema en particular. Finalmente, este modelo se combina con el modelo actual para formar uno nuevo .



Dentro de esta tesis, en el siguiente capítulo se habla extensamente acerca del tema, ya que la parte central del desarrollo de este trabajo.

## 2.7 Examinación y diagnóstico de conocimiento

Se consideran métodos de examinar y diagnosticar el conocimiento del estudiante asumiendo que el Modelo de Estudiante escogido es el modelo de superposición.

En el trayecto de dominar una disciplina académica, el estudiante adquiere o mejora su conocimiento, hábitos y habilidades. Con cada parte de conocimiento y con cada hábito existe asociado un componente del modelo de superposición que caracteriza el nivel de dominio del estudiante de un elemento particular de conocimiento. El valor de cada componente del modelo representa en general una estimación o una magnitud difusa basada en el grado de asociación con un conjunto difuso, un factor dependiente, o una probabilidad subjetiva. Por lo tanto el conocimiento del estudiante en una disciplina académica se representa por un conjunto, o en el caso más general, una red de componentes estimados. En el transcurso de examinar a un estudiante, surgen preguntas acerca de las siguientes áreas

- 1) Evaluación integral del nivel de conocimiento del estudiante (examinar o corroborar el nivel de preparación del estudiante)
- 2) Identificar aquellos subdominios de una disciplina académica, conocimiento, y/o habilidades que no son satisfactorias (diagnóstico).

El conocimiento del estudiante es examinado y diagnosticado por medio de preguntas. Cada pregunta envuelve uno o más elementos de conocimientos de la disciplina en particular y poseen un peso definitivo ("importancia" o "complejidad"). Si la selección de una respuesta con éxito depende de la respuesta dada por el estudiante con anterioridad, la examinación (diagnóstico) del conocimiento del estudiante se dice que constituye un proceso adaptivo de otro modo es no adaptivo o no condicional. En el primer caso el proceso concluye en el momento en que las preguntas que fueron planeadas se acaban, mientras que el último caso concluye una vez que el sistema ha llegado a un estado terminal. La ventaja de la examinación adoptiva (diagnóstico) en un conjunto condicional es, por un lado, pocas preguntas son hechas y consecuentemente el tiempo de examinación (diagnóstico) se reduce, y por otro lado, las preguntas se hacen de un modo propositivo y como consecuencia el sistema de examinación (diagnóstico) parece ser más racional, lo cual da al estudiante confianza en el proceso.

Antes de presentar descripciones formales de los problemas de examinación y/o diagnóstico se considera su rango de aplicación.

La examinación se usa básicamente para monitorear lo siguiente:

- 1- Conocimiento preliminar y habilidades necesarias para el éxito con el trabajo que se llevara a cabo con el sistema de instrucción (pre - examinación )
- 2- Conocimiento y habilidades obtenidas como resultado de la instrucción asistida por computadora, la enseñanza que encierra un maestro humano, aprendizaje independiente, y muchos más (examinación final o de resumen),

Los métodos de diagnóstico se usan para:

- 1- Determinar las características psicológicas del estudiante.
- 2- Generar el estado inicial del estudiante
- 3- Generar hipótesis acerca del estado actual del conocimiento del estudiante con el propósito de manejar el proceso de enseñanza.

Se considera un grafo para describir el área de estudio, el cual consiste de vértices de conceptos (habilidades) conectados por arcos del tipo "es elemental para conocer". Se especifica una partición arbitraria del conjunto de vértices del grafo en subconjuntos que permite estrictamente la intersección y la inclusión de ciertos subconjuntos en otros subconjuntos. Con cada subconjunto se asocia una o más hipótesis acerca del estado del estudiante y habilidades. Por ejemplo: "El estudiante no conoce los conceptos C1 o C2 y la relación entre ellos", "el estudiante no está en disposición de aplicar el teorema de Pitágoras" y "el dominio del estudiante en la materia es confiablemente de un grado de excelente". Usualmente, con cada hipótesis se asocia un valor que caracteriza a la hipótesis, el cual es interpretado en el diagnóstico como la probabilidad de que el estudiante no conozca un concepto en particular, y, en el procedimiento de examinación, como la probabilidad de que el estudiante haya asimilado el área de estudio con el grado especificado por la hipótesis. Como ejemplo se usa dos particiones que están en los límites de la descripción del área de estudio en el grafo. Supóngase que la partición consiste de un solo conjunto que constituye al grafo entero. Se asocian las siguientes dos hipótesis de acuerdo con esta partición:

- 1) el estudiante ha dominado el área de estudio (crédito)
- 2) el estudiante no ha dominado el área de estudio (sin crédito).

La probabilidad de la hipótesis constituyen entonces las estadísticas que gobiernan la asimilación de un área de estudio en particular por algún grupo de estudiantes y puede usarse como datos iniciales con el propósito de determinar el nivel de conocimiento que posee un estudiante. El otro límite de la partición del grafo se obtiene si las clases o conceptos están relacionados uno a uno. Entonces, por medio de correlacionar con cada clase de hipótesis, "el alumno ha asimilado el concepto particular", el cual se caracteriza también por alguna probabilidad, se obtiene un modelo de superposición del Modelo del Estudiante.

A cada hipótesis le corresponde un conjunto de indicadores. Por indicador se entiende algún evento cuya presencia o ausencia tiene influencia para que la hipótesis sea confirmada. El indicador contribuye con cada hipótesis con un peso que caracteriza la contribución del indicador a la hipótesis. Los indicadores se generan como el resultado de un análisis del proceso de solución de problemas, donde la solución de un problema puede apuntar a varios indicadores.

Por lo tanto, el dominio de conocimiento de examinación o diagnóstico se representa por una triple  $(H, S, P)$  donde:

- $H = \{H_i\} (i=1, n)$  es un conjunto de hipótesis,  
 $S = \{S_j\} (j=1, m)$  es un conjunto de indicadores, y  
 $P = \{P_k\} (k=1, l)$  es un conjunto de problemas o lecciones.

Cada hipótesis  $H_i$  de  $H$  consiste en una estimación  $v_i$  y un vector  $w_i = \{w_{ij}\}$ , donde  $w_{ij}$  es el peso del indicador  $s_j$  en la hipótesis dada. En el caso más simple, el peso de un indicador puede tomar el valor de 1 si el indicador afecta a la hipótesis, y un valor de 0 en caso contrario. En el caso general el peso de un indicador puede ser un número en el rango de  $-M$  a  $+M$ ; más aun, el valor absoluto del peso, determina la importancia de la contribución del indicador a la hipótesis, y su signo determina ya sea que la evaluación de la hipótesis incremente o decremente una vez que el indicador es evaluado.

Se usan dos métodos básicos para examinar y diagnosticar el conocimiento del alumno, el primero de los cuales consiste en la aplicación de métodos de optimización tomados de la teoría de cuestionarios que sintetizan árboles de decisión y un segundo el cual envuelve el uso de mecanismos de decisión empleados en el campo de los sistemas expertos. Se consideran ambos métodos.

El principal problema con la teoría de cuestionarios es el desarrollo de algoritmos para hacer cuestionarios óptimos, en los cuales la reducción del precio de un cuestionario usualmente sirve como un criterio de optimización. La teoría de cuestionarios encuentra aplicación en el diagnóstico de dispositivos técnicos, en la prueba de programas, y en la examinación y diagnóstico de conocimiento, métodos de búsqueda directa, programación dinámica, y en el método de límites y subdivisión.

Cuando se usan métodos de inferencia en sistemas expertos, como reglas de producción, métodos de decisión estadística (El método de Bayes, el método Dempster, etc.), etc., el énfasis principal está en preguntas procedurales que envuelven la representación de conocimiento de diagnóstico dentro de la estructura del mecanismo en particular. Más aún, considerable atención es dedicada a los métodos de respuesta a incertidumbre (factores de confianza y probabilidad).

## 2.8 Manejo del Proceso de Enseñanza

Se consideran los conceptos principales que se usan cuando se describen los métodos de manejo del proceso de enseñanza. Todos los métodos de manejo del proceso de enseñanza se derivan del hecho de que existe un objetivo del proceso de enseñanza que está especificado explícita o implícitamente y que describe restricciones impuestas en el conocimiento y habilidades que se espera que el estudiante adquiera como resultado de su interacción con el sistema. Un objetivo de enseñanza está usualmente especificado en la forma de una condición lógica expresada en términos del Modelo del Estudiante y describe el conjunto de estados terminales del conocimiento y/o habilidades del estudiante.

Más aún, existe un Modelo del Estudiante construido inicialmente que ya fue sea sugerido o puesto como resultado de una examinación previa. Existe también un conjunto de operaciones, o entradas instructivas (por ejemplo, salida de información instructiva, refinamiento del objeto de diálogo, salida de las condiciones del problema y la entrada de la solución del problema por parte del estudiante, etc.), que controlan la actividad cognoscitiva del alumno y modifican el modelo actual del estudiante en particular. El material instructivo acerca de los conceptos de la disciplina en particular bajo estudio constituyen los parámetros de las entradas instructivas.

Por lo tanto, el manejo del proceso de enseñanza constituye un proceso dinámico dirigido hacia el cumplimiento de un objetivo de enseñanza bajo las bases del actual estado de conocimiento y/o habilidades del estudiante. Se construye un plan de enseñanza que constituye pasos para conseguir el objetivo. Cada paso constituye, en el caso más general, una secuencia de entradas instructivas. La existencia de material instructivo, junto con el tiempo de enseñanza, constituyen las posibles restricciones en los planes posibles. El plan entonces se implementa paso a paso. Si en el trayecto de presentación de un plan surgen situaciones en las cuales es imposible o no práctico presentar el siguiente paso del plan, entonces surge la necesidad de modificar el plan dado o diseñar un nuevo plan. Por lo tanto, se distinguen dos componentes en el sistema que maneja el proceso de enseñanza: el planeador y el ejecutor del plan.

Los métodos en el proceso de enseñanza se distinguen por el nivel explícito en el cual el objetivo y el plan de enseñanza se presentan en el método en particular, y también ya sea que los planes sean generados dinámicamente o compilados previamente. A continuación se presenta una clasificación de métodos para manejar el proceso de enseñanza y se discuten sus ventajas y desventajas [22]. Primero se distingue entre métodos basados en el concepto de planes de enseñanza (ya sea que este dado implícita o explícitamente) y aquellos métodos que no hacen uso de este concepto en general. Los últimos incluyen lo que es conocido como métodos reactivos y métodos basados en un mecanismo de agenda.

Los *sistemas reactivos*, los cuales son usados para analizar las respuestas del estudiante y/o el estado actual del Modelo del Estudiante, son útiles en la producción de material instructivo,

pero no se usan para planear la secuencia de entrada instructiva. Este método se usa en el diseño de sistemas de apoyo. Cuando se construye un sistema reactivo se debe identificar un conjunto de situaciones "incorrectas"; con cada una de estas situaciones se correlaciona información instructiva necesaria para la explicación y corrección del error. El esquema de implementación de un sistema reactivo se presenta en el siguiente pseudocódigo:

```
EjecutorSistemaReactivo(ME):  
  MIENTRAS ( NO fin_de_la_tarea)  
    HAZ  
      EstimacionDelTrabajoEnElSistemaBase (SM, fin_de_la_tarea):  
      SI situacion_1 (SM) ENTONCES Instrucción_informativa_1(ME):  
      SI situacion_2 (SM) ENTONCES Instrucción_informativa_2(ME):  
      ...  
      SI situacion_n (SM) ENTONCES Instrucción_informativa_n(ME):  
  FIN:
```

Aquí el control lo lleva a cabo un simple ciclo y la condición la forma en que el ciclo es exitado es independiente del estado del Modelo del Estudiante (ME) y en su lugar es determinado por el estado de operación del usuario que esta en la base de datos del sistema de proceso. Después de que el usuario realiza ciertas operaciones en el sistema base, donde el volumen de trabajo puede ir desde la entrada de un simple símbolo hasta la solución de un problema complejo, su trabajo se evalúa, y si el Modelo del Estudiante satisface la situación esperada, se provee de información instructiva apropiada. La ventaja de un método reactivo radica en la simplicidad de su implementación, sin embargo el inconveniente es de que carece de un objetivo de enseñanza, el cual restringe su aplicación a sistemas de apoyo, ya que responde a acciones del usuario, pero no maneja la actividad cognoscitiva del estudiante.

Un *mecanismo de agenda* es un método de administración de acciones o, la aceptación de determinada terminología, administración de situaciones basadas en una lista ordenada dinámicamente de problemas (agenda). El orden de problemas de la lista se determina por medio de reglas heurísticas que establecen un grado de preferencia de ejecución de cada problema con el fin de alcanzar los objetivos actuales. Por medio de un mecanismo de agenda los problemas pueden ser seleccionados de tal modo que satisfagan varios objetivos simultáneamente, o, si existen objetivos diferentes, permite la selección de problemas que produzcan la mejor solución. Un mecanismo de agenda se uso en el diseño del STI SCHOLAR, con el propósito de ordenar tópicos de discusión. En la agenda se incluyen tópicos de discusión junto con un tiempo permitido, el cual es proporcional a la importancia del tópico, el sistema introduce nuevos tópicos en la lista que están relacionados al tópico actual o que son referidos en las preguntas o respuestas del estudiante. Una vez que el tópico se agota, o el tiempo permitido para su exploración se termina, el sistema selecciona un nuevo tópico para su discusión. Otro método se basa en el modelo de estudiante de superposición de tipo vector donde cada uno de sus elementos supone acerca del grado de dominio de una habilidad en particular que corresponde al elemento asociado. El calculo de las propiedades de las hipótesis se basa en el teorema de Bayes. Más aun, existe un modelo del proceso de olvido de habilidades previamente adquiridas. El objetivo del sistema es cumplir niveles específicos de cobertura (probabilidades iniciales) para todas las habilidades (o ciertas habilidades seleccionadas) empleadas en problemas de un grado de dificultad en un tiempo mínimo. En cada paso se propone una agenda de todos los problemas que requieren el uso de habilidades específicas para resolverlos. Se selecciona un problema que posee la dificultad optima para el estudiante.

Se muestra a continuación un pseudocódigo del mecanismo de agenda. Aquí los "objetivos" se refieren a una lista de objetivos de enseñanza, ME es el Modelo del Estudiante y "agenda" es la lista de problemas (entradas instructivas). Nótese que la agenda es muy similar a un plan, a diferencia de este ultimo en que constituye una secuencia de acciones ordenadas en

# Marco Teórico

términos de prioridad, cada uno de los cuales está orientado hacia la óptima satisfacción de los objetivos locales (tácticos). En resumen, la ejecución de una simple acción puede conducir a la eliminación de otras operaciones de la agenda. En contraste, un plan constituye una secuencia lógica consistente y ordenada de acciones en el cual su ejecución satisface objetivos estratégicos globales y tácticos. Un mecanismo de agenda, sin embargo, es un método extremadamente flexible de manejo y frecuentemente es empleado como un componente de un método de manejo altamente desarrollado.

MecanismoDeAgenda(objetivos, ME):

Inicialización(objetivos, ME, agenda):

**MIENTRAS** (NO Satisfacción(objetivos))

**HAS**

FormulaAgenda(objetivos, ME, agenda):

EjecutaProblema(agenda[1], ME):

**FIN:**

Se consideran ahora métodos basados en un plan explícito o implícito. Se distingue entre sistemas con un plan definido y sistemas con un plan incorporado en el momento de ejecución. El primer grupo incluye los sistemas tradicionales de enseñanza o, como se dice usualmente, Sistemas de Instrucción asistida por computadora (IAC). La mayoría de los sistemas IAC hacen uso de simplemente un plan de enseñanza definido, generalmente implícito en el código del programa. El objetivo de enseñanza en un sistema IAC no está expresado explícitamente, pero solo sugerido por el programador. Aunque el éxito de los IAC en la enseñanza es evidente, estos poseen un número de carencias. La flexibilidad que exhibe los sistemas de instrucción IAC con el estudiante es limitada, ya sea que permiten al usuario un número reducido de preguntas o en general no permitan ningún tipo de pregunta. Los IAC no contienen un Modelo del Estudiante y manejan el proceso de enseñanza basándose en un estudiante promedio en vez de basarse en el usuario que está frente a la computadora. Más allá, una estructura de este tipo para representar conocimiento en un sistema IAC requiere recursos considerables para poner en marcha el sistema y su mantenimiento. Estas carencias estimulan el desarrollo de sistemas con planes construidos en el momento, de entre los cuales se distinguen entre sistemas que generan un plan y sistemas que seleccionan un plan de una librería. Se conocen los siguientes métodos de diseño de planes:

- Generación de planes controlados por objetivos.
- Selección y especialización de planes de estructura propuestos en el sistema MENTOR.
- Selección de planes de estructura basados en el mecanismo de pizarrón

A continuación se consideran los primeros dos métodos de construcción de planes.

La creación de una currícula controlada por objetivos se basa en el método STRIPS desarrollado para planear las acciones de un robot, en un medio ambiente dinámico. El sistema para el manejo del proceso de enseñanza presupone la presencia de dos componentes fundamentales: el planeador y el ejecutor del plan. El planeador genera el plan que satisface el último objetivo de enseñanza y el estado actual del Modelo del Estudiante, y el ejecutor del plan implementa el plan, y, si el comportamiento del estudiante no corresponde a lo que se esperaba, una ruta alternativa del plan se implementa o se induce al planeador a que cree un nuevo plan. A diferencia del manejo del comportamiento de un robot en el manejo del proceso de enseñanza no hay relación directa entre los efectos anticipados y reales inducidos en la ejecución de la entrada instructiva. Las acciones que lleva a cabo un robot se monitorean visualmente; sin embargo, el efecto de una entrada instructiva en el estado más profundo del mundo del estudiante es inaccesible a observación directa. Dependiendo de esto último, solo se hacen sugerencias bajo las bases del Modelo del Estudiante. En el manejo del proceso de enseñanza, por lo tanto, surgen situaciones inesperadas y es muy probable el fracaso de la ejecución de un plan, un resultado

# Marco Teórico

que impone demanda adicional al mecanismo que pueda ayudar a encontrar una forma de solución en una situación inesperada en un sistema de manejo del proceso de enseñanza.

Se considera el proceso de enseñanza de conceptos que pertenecen a algún área de estudio. Se consideran representaciones incorrectas de los conceptos del el área particular de estudio, en suma, a representaciones correctas. Puede haber un muchas representaciones incorrectas de un simple concepto. Como Modelo de Estudiante se usa el modelo de superposición (donde con cada concepto se asocia un elemento del vector que toma el valor de 1 si el alumno conoce el concepto en particular, y un valor de 0 de otro modo), o el conocimiento del alumno se representa por una formula lógica, o una conjunción de predicados CONOCE( $X$ ), donde  $X$  es el nombre del concepto. Un modelo de estudiante es inconsistente si presenta información de un concepto y al menos una representación incorrecta de este concepto. Note que un modelo de estudiante consistente puede presentar información de varias interpretaciones incorrectas de del mismo concepto simultáneamente. Un objetivo de enseñanza se representa en la forma de expresiones lógicas que incluyan conectores lógicos (AND – &, OR – |, NOT – ~) y ocurrencias del predicado CONOCE (por ejemplo, la expresión CONOCE (seno) & CONOCE (coseno) que representa un objetivo de enseñanza que expresa el requerimiento que el ultimo modelo de estudiante especifica conocimiento de los conceptos "seno" y "coseno".

Una entrada instructiva se representa en la forma de un cuádruple ( $N, P, E, A$ ), donde  $N$  es el nombre de la entrada instructiva,  $P$  la especificación de prefacio de la entrada instructiva (por ejemplo, la condición bajo la cual, si es verdadera, hace posible aplicar la entrada en particular),  $E=(al, dl)$  es la especificación de los efectos anticipados de la ejecución de la entrada instructiva, la cual consiste de dos listas:  $al$  es la lista de suplementos que contienen expresiones lógicas que se suman al Modelo del Estudiante ( o la asignación de los elementos del vector a 1) y  $dl$  es una lista de expresiones eliminadas, que contienen expresiones lógicas que fueron eliminadas del Modelo del Estudiante ( o la asignación a elementos del vector al valor 0 ),  $A$  es una acción que involucra la presentación de material instructivo al estudiante con el fin de inducir la salida deseada.

Las primeras tres partes del la entrada instructiva son usadas por el planeador para generar el plan y por el ejecutor del plan durante el tiempo que el plan esta en marcha. El cuarto componente lo usa solamente el ejecutor del plan para interactuar con el estudiante. Para simplificar la presentación se asume que el cuarto componente tiene la forma ENSEÑA( $X$ ), donde  $X$  es el nombre de un concepto, CORRIGE( $Y$ ), donde  $Y$  es el nombre de un concepto asociado con una representación incorrecta de un concepto. Una entrada instructiva en la cual la acción tiene la forma CORRIGE( $Y$ ) se conoce como entrada instructiva correctiva. Nótese que, respecto a las diferentes formas en que el material instructivo puede presentarse, existen varias formas en que las entradas instructivas se presentan, pueden haber varias entradas instructivas para cualquiera de los conceptos o para las representaciones incorrectas de los conceptos. Para distinguir las entradas instructivas para un concepto en particular, se presenta una enumeración de la forma CONOCE- $K(X)$  Y CORRIGE- $K(Y)$ , donde  $K= 1,2,\dots$ .

Un plan de enseñanza se presenta en la forma de un grafo dirigido bipartita acíclico. Los dos tipos de vértices en un grafo (pasos y objetivos) corresponden a entradas instructivas y objetivos de enseñanza respectivamente. Los arcos especifican el orden de ejecución de las entradas instructivas. Note que en el caso general el grafo del plan de enseñanza puede no estar conectado (puede consistir en varios subgrafos sin vértices en común). Un segundo elemento importante en este tipo de plan de enseñanza es la oportunidad que permite representar varias rutas alternativas para el cumplimiento de ciertos objetivos.

A continuación se selecciona un objetivo del los datos básicos y se presenta un algoritmo para ejecutar el plan. También se amplia la presentación del algoritmo de planeación basado en el método de síntesis estructural que inicialmente fue desarrollado para generar planes para la solución de problemas computacionales.

Se sigue la siguiente representación de datos.

*ME*, el Modelo del Estudiante representado por el conjunto de nombres de los conceptos que el estudiante conoce.

*OBJETIVO*, el objetivo de enseñanza, presentado como el conjunto de nombres de conceptos que el estudiante va a aprender.

*II*, entradas instructivas, representadas como un conjunto elemental de entradas instructivas (pasos).

*PASO*, una entrada instructiva elemental, que consiste en una lista de tres elementos (*P*, *A*, *E*), donde *P* es el prefacio, o conjunto de nombres de conceptos. *A* es una acción, o nombre de una subrutina que define interacción con el estudiante, y *E* son los resultados anticipados, representados en la forma de una lista de dos elementos (*al*, *dl*), donde *al* es una lista de adiciones (el conjunto de elementos sumados al *ME*), y *dl* es una lista de eliminaciones (el conjunto de elementos eliminados del *ME*)

*PLAN*, el plan de enseñanza, representado en la forma de una lista (secuencia) de pasos, los cuales constituyen una entrada instructiva elemental.

A continuación se presenta un pseudocódigo. Aquí *LISTO* constituye un conjunto de pasos del plan que están listos para su ejecución. El algoritmo contiene dos ciclos embebidos. En el ciclo exterior el planeador es invocado con el propósito de generar un nuevo plan (*PLAN1*) basado en el estado actual del *ME*, los objetivos, y el plan anterior (*PLAN*), la ejecución de cual termina en fallo. Más aun, aquí el conjunto de pasos en el plan que están listos para ser ejecutados se determina. En el ciclo interior el plan actual se ejecuta mientras existan pasos razonables en el plan y los objetivos de enseñanza se satisfagan. Del conjunto de pasos que están listos para su ejecución un paso razonable y no redundante se selecciona para que el *ME* se modifique. El paso se elimina del plan y se presenta una entrada instructiva, en el momento en el que el *ME* puede cambiar, y entonces se determinan un conjunto de acciones razonables del plan actual. Si el conjunto de todos los pasos posibles está vacío y el objetivo de enseñanza no ha sido satisfecho, al ejecución del plan termina sin éxito, y el ejecutor del plan invoca al planeador con el propósito de conseguir un nuevo plan.

```
EjecutordePlan( OBJETIVO, ME )
PLAN:= ( ) ;
  mientras ( ME no contenga OBJETIVO )
has
Planeador ( ME, PLAN, OBJETIVOS, PLAN1 ):
LISTO:= ( ) ;
  para cada PASO=( P, A, E ) en PLAN1:
  has
    si (ME contiene P)
    entonces LISTO:=LISTO + STEP1:
    fin:
  mientras no (LISTO:=[ ])
  has
    si ( ME contiene OBJETIVO ) entonces termina todo:
    para cada PASO=( P, A, E ) en LISTO
    has
      si no(( ME contiene E.al ) Y
        ( ME no contiene E.dl ) ) Y
        ( ME contiene P )
      entonces
        ME:=(ME + E.al) - E.dl:
```

```
PLAN1:=PLAN1 - PASO:
Ejecuta( A, ME )
para cada STEP1= (P, A, E) en LISTO
has
    si ( ME no contiene P )
        entonces READY:=READY - STEP1:
    todos
fin:
todos
fin:
PLAN:=PLAN1
fin:
```

Se considera ahora el algoritmo planeador, donde son identificadas tres secciones principales: formación de objetivos: generación de un plan parcial: y el acto de combinar el plan parcial y el plan anterior. Al formar el objetivo se hace una búsqueda del llamado frente del plan anterior (por ejemplo, el estado del estudiante (ME) después de que el plan previo fue presentado). La parte frontal del plan previo se toma como el objetivo final (*OBJETIVO1*) del plan parcial (*PLAN1*) el cual, al combinarse con el plan previo (*PLAN*), produce un nuevo plan (*PLAN1*) para la satisfacción de los objetivos de enseñanza finales.

```
Planeador(ME, PLAN, OBJETIVO, PLAN1)
si PLAN = ( )
    entonces OBJETIVO1:=OBJETIVO
    de otro modo OBJETIVO1:=[]:
    para cada PASO = (P, A, E) en PLAN
        has OBJETIVO1:=OBJETIVO1 + P
    fin:
    para cada PASO = (P, A, E) en PLAN
        has OBJETIVO1:=OBJETIVO1 - E.al
    fin:
todo
GeneraPlan (ME, OBJETIVO1, PLAN1):
PLAN1:=PLAN1 !! PLAN:
```

Se presenta a continuación el algoritmo de generación del plan. Consiste de dos secciones principales: la generación de un plan para satisfacer los objetivos y la eliminación de pasos innecesarios en el plan. Al generar el plan todas las entradas instructivas se eliminan de II. Si el prefacio de una entrada instructiva satisface al Modelo del Estudiante, la entrada instructiva se incorpora al plan y el Modelo del Estudiante se modifica. Entonces el plan contendrá todas las posibles entradas instructivas posibles para el Modelo del Estudiante dado, incluyendo aquellas que no son necesarias (por ejemplo, aquellas que no satisfacen el cumplimiento del objetivo final). Si la generación del plan proviene del Modelo del Estudiante actual hacia el objetivo (ruta directa), el procedimiento de liberación de pasos innecesarios del plan se lleva a cabo de manera inversa (la ruta inversa). Se usan dos conjuntos en la limpieza, *M1*, el conjunto de objetivos intermedios que se llevan a cabo a través de las rutas que conducen al objetivo final, y, *M2*, el conjunto de objetos intermedios que se llevan a cabo a través de las rutas que conducen a objetivos adicionales. Se calculan a través de cada ruta los conjuntos *M*, o predecesores de objetivos adicionales, y *N*, o predecesores de objetivos de enseñanza y se complementan los conjuntos *M1* y *M2*.

```
GeneraPlan(ME, OBJETIVO, PLAN)
ME1:=ME:
```



```
mientras (ME no contenga OBJETIVO)
  has
    PLAN:= [ ]:
    para cada PASO = ( P, A, E ) en II
      has
        si (ME contiene P)
          entonces PLAN := PLAN + STEP
        todos
      fin:
    fin
M1 := OBJETIVO:
M2 := ME - OBJETIVO:
mientras no M1 = ME1)
  has
    M:= [ ]: ME1:=ME1-M1:
    para cada INTER en M2
      has
        si ( esta PASO = ( P, A, E ) )
          y ( INTER en E.al)
          y ( todos INTER en M1 ) implican ( INTER1 no en M1 )
        entonces
          PLAN := PLAN - PASO
          M := M + P:
        todos
      fin:
    N := [ ]:
    para cada INTER en M1
      has
        si ( es PASO = ( P, A, E ) ) y ( INTER en E.al)
          entonces N:= N + P
        todos
      fin:
    M1 := N : M2:= M - N:
  fin:
```

En el sistema MENO-TUTOR el conocimiento de la técnica de enseñanza se representa en la forma de una red de manejo de dialogo preasignada (COURT). COURT se considera como un conjunto de unidades de decisión ordenadas jerárquicamente. Se distinguen tres niveles en la estructura de COURT: pedagógico, estructural y táctico. Una transición de un nivel alto a un nivel bajo conlleva a la especialización de objetivos y acciones del sistema. En el nivel más alto, el nivel pedagógico, el sistema establece un numero de restricciones de la forma en que el dialogo tiene que ser conducido (por ejemplo, que tan frecuente debe el estudiante ser interrumpido o que tan frecuentemente se debe verificar su trabajo para buscar nociones incorrectas del área de estudio) y selecciona un método para trabajar con el estudiante, ya sea que presente un nuevo tópico, enseñe el tópico actual, corrija una noción falsa, o complemente el tópico actual. En el segundo nivel la decisión pedagógica se especializa en la forma de una estrategia de enseñanza. Por ejemplo, al enseñar un tópico actual el perfecto entendimiento del estudiante en el tópico se investiga por medio de hacerle una serie de preguntas. En el nivel más bajo o táctico, los niveles de acción se seleccionan con el propósito de implementar una estrategia de enseñanza. Por ejemplo, con el fin de describir un tópico el sistema puede escoger proveer al estudiante con conocimiento específico o general, proponer analogías o proveer un ejemplo del concepto, etc.

En suma a las relaciones jerárquicas, todas la unidades de la red (bloques) se conectan en una secuencia que especifica el orden de ejecución para un estudiante ideal. Se hace

# Marco Teórico

referencia a estas relaciones por default. Más aun, con cada unidad se asocian reglas las cuales, si el comportamiento del estudiante difiere de lo que se espera, se habilitan para moverse a otra unidad de la red. En MENO - TUTOR existen 20 reglas que funcionan a niveles tácticos y estratégicos. Como ejemplo se consideran las siguientes reglas:

*RE* - INVESTIGA- Una regla estratégica.

*I*Z: Datos de enseñanza.

*V*: Investigar - comprensión.

*DESCRIPCIÓN*: Transfiere al sistema al modo que genera una serie de preguntas generales en tópicos diferentes,

*INVOCACIÓN*: El tópico actual se complementa y el sistema no esta seguro de que el estudiante tiene un dominio sobre el material instructivo que mostró.

*COMPORTAMIENTO*: Una explicación de la transición de una examinación detallada de un tópico a una examinación superficial de varios tópicos diferentes se genera dentro de la estructura del conocimiento del estudiante que se asume que posee.

Cada unidad de COURT se implementa como una estructura cuyos elementos determinan aquellos detalles como la especificación de la salida del texto, el estado de éxito de la transición, actualización del Modelo del Estudiante y del modelo de interacción, y una lista de las meta - reglas asociadas con el nodo dado.

Antes de considerar el algoritmo ejecutor del plan, se formalizan los conceptos básicos usados en el método presente, tales como la unidad que pertenece a COURT, meta - regla y COURT.

Una unidad que representa a COURT se representa en la forma de una triple ( A, M, G ), donde A es una secuencia de acciones, M una lista de meta - reglas, y G el nombre de un nodo default que tuvo éxito, COURT se representa por medio de un mapeo del conjunto de nombres de unidades en unidades.

Una meta - regla se especifica por el triple  $R=( C, NT, A )$ , donde C es una condición que invoca una meta - regla, NT es el nombre de la unidad cuyo control se transfiere, y A son las acciones ejecutadas previamente a la transferencia de la unidad de control NT.

Se considera el algoritmo del ejecutor del plan, el cual constituye un interprete de COURT. Aquí COURT es una red de control de dialogo, puesta en la forma de una librería de diálogos de enseñanza, representada en la forma de un archivo de unidades. INIT es el nombre de la unidad inicial en COURT; *UNIT* es una variable que contiene el nombre de la unidad activa actual de COURT. El algoritmo contiene un ciclo: la satisfacción del objetivo de enseñanza constituye la condición de salida para el ciclo. En el ciclo para la unidad actual de COURT, las acciones asociadas con esta unidad se ejecutan, en la cual el Modelo del Estudiante puede cambiar. Después se selecciona la unidad activa con éxito de COURT, la cual puede ser la unidad default o la unidad de recibimiento de la regla recién ejecutada

EjecutorDelPlanCOURT ( OBJETIVO, ME, COURT, INIT)

UNIT:=INIT;

**mientras** (ME no contenga OBJETIVO)

**has**

**deja** COURT [ UNIT ] = ( A, M, G ) **en**

**comienza**

Ejecuta ( A, SM):

UNIT := G;

**para cada** R **en** M

**has**

# Marco Teórico

```
deja R= ( C, NT, A ) en
  comienza
    si C
      entonces Ejecuta ( A, SM )
      UNIT := NT;
    fuera:
      todos
    fin
  fin
fin
fin
```

# Capítulo tres

## Modelado del Estudiante

Durante el proceso de enseñanza del estudiante, los Sistemas Tutoriales Inteligentes (STI) deben continuamente mantener un apoyo a las diferentes actividades de este proceso. Esta tarea se debe hacer y ajustar de acuerdo a cada estudiante ó a cada individuo. No existe una definición exacta de lo que es enseñar "inteligentemente". Sin embargo los STI comparten la característica de inferir el estado cognoscitivo del estudiante y adaptan la enseñanza de acuerdo a este estado, el cual es modelado dentro del sistema [31].

El termino "Modelo del Estudiante" se refiere a toda aquella información que el sistema tiene del usuario en particular. Las técnicas de Planeación permiten formalizar el conocimiento instructivo para desarrollar una sesión de enseñanza interactiva. Mediante un módulo de enseñanza se planea el siguiente paso instructivo, tomando en cuenta el objetivo pedagógico y el estado en que se encuentra el estudiante (que está representado dentro del sistema por el Modelo del Estudiante). [24]

El inferir un modelo de estudiante (actualizarlo o modificarlo) se denomina diagnóstico. El Modelo del Estudiante y el modulo de diagnostico están muy relacionados. El Modelo del Estudiante es una estructura de datos, y el diagnóstico es un proceso que la manipula. Los dos componentes se deben diseñar juntos. A este diseño se le conoce habitualmente como "Modelado del Estudiante".

Las características del modelado del estudiante revelan los fundamentos necesarios en la investigación sobre psicología cognoscitiva. Los STI se encuentran en una área de investigación relativamente nueva, la cual desde un principio (junto con los métodos de Inteligencia Artificial), ha ido de la mano con los métodos cognoscitivos. En un inicio, la investigación se concentraba en la representación del conocimiento. El modelado del estudiante, por otro lado del modelado cognoscitivo, ha avanzado. En este contexto, la investigación se enfoca en la representación del conocimiento que corresponde a los procedimientos mentales que el estudiante tiene respecto al dominio de aplicación.

Básicamente para el modelado del estudiante se requiere de un trabajo psicológico, ya que en cualquier momento se hacen aserciones sobre el conocimiento que posee el estudiante. Sin embargo, en la estructura del desarrollo de un STI, el trabajo psicológico está ligado íntimamente con el esfuerzo técnico requerido para llegar al grado deseado de modelado (granularidad). Existen varias técnicas de modelado del estudiante las cuales poseen diferentes características.

[31]: Se proponen las siguientes diferentes funciones que tiene el Modelo del Estudiante [24],

*Avance*, para el apoyo a la selección del siguiente tópico a enseñar.

*Intervención*, para el apoyo a la intervención del sistema cuando el alumno requiere consejo (ayuda).

*Generación de problemas*, para el apoyo a la generación de problemas que el estudiante requiere de acuerdo a su condición.

*Explicación apropiada*, para el apoyo a la explicación de acuerdo a los conceptos que el alumno solo conoce.

*Diagnóstico*, para la identificación de errores (malos procedimientos ).

*Estratégico*, para dar información para la selección de una estrategia de enseñanza

*Predictivo*, para anticiparse a lo que el estudiante va a hacer.

*Evaluativo*, evaluación del sistema y del estudiante.

# Modelado del Estudiante

## 3.1 Elementos principales en el Modelado del Estudiante

Una vez descrito el panorama que involucra al Modelo del Estudiante y al modelado del estudiante también se definen los requerimientos básicos para construir un Modelo del Estudiante basado en Multimedia.

Cuando un estudiante usa un STI algunas decisiones instructivas se deben hacer, Quien toma estas decisiones ? Existen tres agentes involucrados; el estudiante, los diseñadores y el sistema. Algunos entusiastas justificaran los beneficios que tiene el que el usuario tome las decisiones; un diseñador de STI tratara de enriquecer el sistema para que tome las decisiones.

Estas cuestiones presentan un tricotomía falsa. Se debe buscar un balance apropiado de derechos de toma de decisiones. Este balance depende de la naturaleza del estudiante, la tecnología disponible, las actividades de enseñanza y el dominio de conocimiento.

Por instructivo se entiende el proceso de toma de decisiones que ayudan al aprendizaje, Una teoría de Instrucción se deriva de una teoría de aprendizaje, por lo tanto no debe prescribir actividades que esta considere improductivas. Self [24] comenta que "una teoría de aprendizaje provee las bases para el desarrollo (deductivo) de una teoría de instrucción, pero una teoría de instrucción solo puede desarrollarse (inductivamente) mediante prueba y error".

La fuente de decisiones instructivas o principios generales de instrucción requiere la especificación de los siguientes componentes:

1. Una teoría de Aprendizaje
2. Una teoría de conocimiento, si la teoría de Aprendizaje hace referencia a las relaciones entre piezas de conocimiento.
3. Un conjunto de acciones instructivas ( en términos de posibles interacciones con el estudiante y los convenios con la practica educacional)
4. Las características relevantes del estudiante ( en términos de conocimiento previo y diferencias individuales)
5. Las metas instructivas (en términos del conocimiento y actitudes que deben ser adquiridas por el estudiante).

Estos componentes son estrechos. Es de hecho posible que una acción instructiva sea la de "no hacer nada" (en el caso de un medio ambiente de enseñanza por descubrimiento - Discovery Learning Enviroment).

Una *teoría de Aprendizaje* es un conjunto de cláusulas que describen como el estado cognoscitivo de alumno cambia como una consecuencia de una acción instructiva. En general se definen por un conjunto de axiomas de la forma:

Estado(AlumnoX,ME0,Situación)&)AcciónInstructiva(AI,x)ENTONCES  
Estado(AlumnoX,ME1,AI(x,Situación))

Una *teoría de Conocimiento* define las relaciones entre elementos de conocimiento a los cuales la teoría de Aprendizaje hace referencia. Por ejemplo, si la teoría de Aprendizaje tiene un axioma que enuncia que un concepto análogo debe ser presentado antes de un conocimiento más complejo entonces la teoría de conocimiento debe definir que se entiende por "análogo". En general, esto envuelve formalizaciones usadas en representación del conocimiento usadas en Inteligencia Artificial (IA).

# Modelado del Estudiante

El tercer componente envuelve la definición de un conjunto de *acciones Instructivas*. Una teoría de enseñanza define los efectos generales de acciones instructivas, pero no dice que acciones están disponibles.

A continuación se necesita una descripción del estudiante en particular (ver el apartado anterior)

Finalmente, se deben definir los *objetivos instructivos*. En general, se expresan en términos de las propiedades de cualquiera de los componentes de las descripciones realizadas. En general, el objetivo debe minimizar el número o costo de acciones instructivas para que el estudiante alcance algún estado cognoscitivo con propiedades específicas.

Una vez que los cinco componentes se definen, se debe probar que el objetivo se alcanza. La prueba y las conclusiones pueden presentarse como una "guía instructiva". El producto es un proceso de prueba de teoremas y puede, en principio, llevarse a cabo por medios automáticos. El propósito general es expresar la teoría de enseñanza, teoría de conocimiento, acciones instructivas, atributos del estudiante y objetivos instructivos formalmente con el fin de que una teoría instructiva pueda representarse como un conjunto de conclusiones derivadas rigurosamente de suposiciones, no como un conjunto de guías expresadas vagamente. Las premisas de la teoría de aprendizaje, deben, por supuesto, ser verificadas independientemente (si es posible) por experimentación psicológica.

## 3.2 Clasificación de Modelos de Estudiante

A continuación en la figura 1 se presenta una clasificación de Modelos de estudiante. Se hace distinción entre modelos de variables y simulación, donde los primeros tipos constituyen conjuntos de variables que caracterizan el estado del conocimiento del estudiante y habilidades, los últimos sirven para reconstruir las nociones del estudiante acerca del área de estudio y el mecanismo de solución de problemas que emplea.

La versión más simple de un modelo de estudiante variable es el *modelo de vector*. Un modelo de vector asocia cada concepto y/o habilidad que se estudia a un elemento que toma dos valores "conoce/ no conoce" o "puede / no puede". El estado del conocimiento del alumno se determina por el conjunto de valores de los elementos del vector, y el número de posibles valores es  $2^n$  donde  $n$  es la dimensión del vector (por ejemplo, el número total de conceptos y habilidades que se estudian). En el caso más general, números (por ejemplo, en el rango de 0 a 1) que caracterizan el grado de asimilación del estudiante en un concepto o habilidad en particular pueden servir como valores de los elementos del vector; más aun, cada elemento del vector puede tener asignado su rango propio o conjunto de valores. Una ventaja del modelo de vector es su simplicidad, por otro lado sufre la carencia de que no refleja el proceso cognoscitivo y los métodos de solución de problemas que el estudiante tiene, así como también no permite la relación entre conceptos y habilidades.

Un modelo de estudiante más flexible es el *modelo de red*, el cual constituye un grafo cuyos nodos corresponde a conceptos y/o habilidades, y cuyos arcos corresponden a las relaciones entre estos. Con cada nodo o arco se asocia un valor o conjunto de valores que caracterizan el grado de dominio que el estudiante posee en un concepto o habilidad en particular; la herencia de valores se permite.

Los modelos de estudiante hasta ahora descritos son conocidos como modelos de superposición, ya que cada uno de estos modelos es en cierto sentido un subconjunto (cobertura) de un modelo del especialista. Los modelos de Superposición se recomiendan altamente en la enseñanza de conceptos, pero no son suficientes en la enseñanza de habilidades. La principal restricción que existe en los modelos de superposición es que la solución del estudiante se evalúa desde el punto de vista de las habilidades del experto, por el contrario puede haber diferencias

# Modelado del Estudiante

considerables entre las habilidades de un novicio y las de un experto. Más aun, un estudiante creativo puede aplicar una técnica efectiva que no esté en el arsenal del experto.

Un modelo conocido con el nombre de *red semántica* es una extensión del modelo de superposición el cual produce un incremento en la efectividad del proceso de diagnóstico en el caso de la enseñanza de conceptos. Los nodos de un *red semántica*, representan hechos, reglas, o procedimientos que describen las habilidades potenciales conectadas por arcos de relación del tipo: "generalización", "especialización", "analogía", "refinamiento", "desviación" y "corrección". Al contrario de un modelo de superposición de red, un *red semántica* contiene no solo las habilidades del experto, si no también diferentes generalizaciones, especializaciones, y desviaciones de las reglas empleadas por el experto. Más aun, por medio del uso de las relaciones de "refinamiento", un *red semántica* describe rutas potenciales para el desarrollo de habilidades del estudiante. Una ruta en un *red semántica* corresponde a un estudiante en

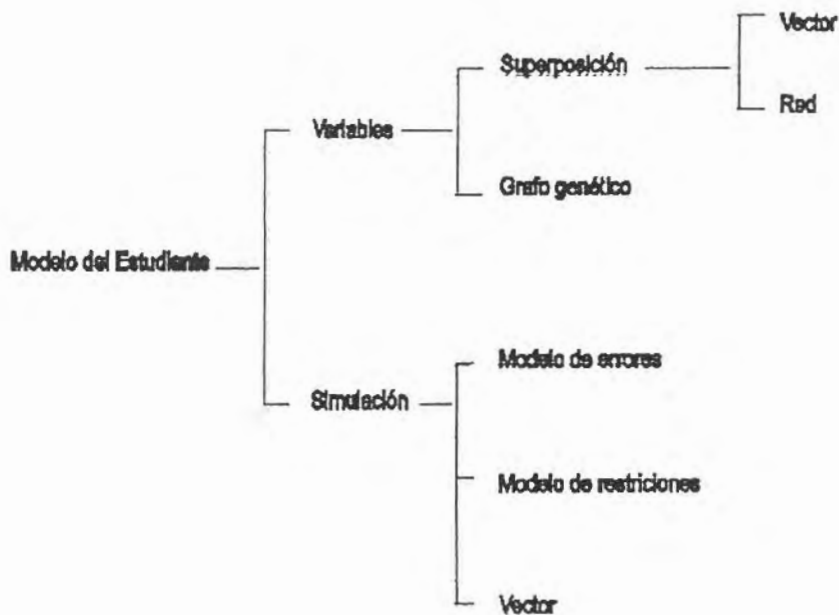


figura 1

Clasificación de Modelos de Estudiante

particular. Este método ha sido empleado con éxito para diagnosticar errores en el proceso de toma de decisiones en un ambiente de juegos teóricos, en la implementación de operaciones aritméticas, y en la representación de poses y movimientos en el ballet clásico. Una *red semántica* es el más desarrollado de todos los modelos de variables. Sus principales restricciones es que aunque es altamente elaborado, sus datos representan una estructura pasiva que refleja el estado actual de conocimientos y habilidades del estudiante, no es un modelo del mecanismo cognoscitivo que emplea el estudiante.

En los modelos de simulación el conocimiento del estudiante se representa en la forma de estructuras de datos y las habilidades del estudiante se representan en forma de procedimientos y mecanismos de interpretación de datos. Se supone que  $O$  es un conjunto de operaciones elementales (hábitos) que son necesarios para resolver un tipo particular de problemas. Entonces, una habilidad constituye un algoritmo que comprende una secuencia de operaciones elementales. Si el algoritmo se representa mediante un grafo cuyos nodos corresponden a los estados del problema y cuyos arcos corresponden a operaciones elementales, la solución de un problema en

# Modelado del Estudiante

particular corresponderá a un arco en particular. en el grafo. Si una simple operación en el algoritmo se reemplaza por otra operación, todas las rutas que incluya este arco conducirán a soluciones falsas que modelan un comportamiento no adecuado del estudiante.

Supóngase que  $M$  es un conjunto incorrecto de operaciones elementales (llamadas Malas - Reglas) que se han descubierto en el proceso de experimentos con estudiantes. Entonces por medio de reemplazar en el algoritmo ciertas operaciones del conjunto  $O$  por operaciones del conjunto  $M$ , Se obtienen modelos del estudiante que representan mal comportamiento, conocidos como malas - reglas. Cuando se construyen *modelos de error* o malas - reglas se imponen restricciones rígidas bajo la forma de en que se representan las habilidades y los mecanismos bajo los cuales las habilidades son interpretadas. Más aun, en estos modelos puede ser difícil de diagnosticar el comportamiento del alumno, o el diagnóstico puede ser incorrecto, si el conjunto  $M$  esta incompleto. La construcción del conjunto  $M$  para un conjunto de problemas envuelve un esfuerzo considerable por parte de especialista psicólogos. Los algoritmos de búsqueda para este tipo de modelos son, en general, de complejidad exponencial, el cual los hace difíciles de usar en sistemas interactivos.

Por medio de los *modelos de restricciones* es posible construir un Modelo del Estudiante en términos familiares. El modelo en sí mismo constituye un modelo de restricciones bajas y altas impuestas por los conceptos y/o habilidades que se estudian. Una restricción baja es la más especializada, y una restricción alta es de la forma más general. Una restricción se obtiene como el resultado de razonamiento inductivo por parte del estudiante cuando resuelve un problema en particular. Siguiendo con la solución del siguiente problema, el modelo se redefine de la siguiente manera: Si como resultado de razonamiento inductivo un par de nuevas restricciones se obtiene, estas restricciones son comparadas con las restricciones previas: más aún, si una nueva restricción baja (o alta) es más general (más especializada) que la anterior, la primera restricción es reemplazada por la ultima. De otro modo el nuevo par se añade al conjunto de restricciones. Este método es prometedor, ya que, por un lado, se basa en nociones objetivas al hecho de que el maestro no conoce el estado de conocimiento y/o habilidades del estudiante, pero trata de dar una solución en base en aproximaciones de este estado, y, por otro lado, se aplica conocimiento y usa métodos de enseñanza que involucran ejemplos desarrollados para enseñanza a maquinas (machine learning).

Como un todo, el desarrollo de Modelos de simulación está en una etapa temprana de investigación en la cual el número de problemas excede el numero de resultados obtenidos. Tal estado se justifica por la complejidad de los problemas y el bajo nivel de desarrollo de las técnicas para resolverlos. Más aun, la confirmación de construcciones teóricas conlleva experimentos intensivos en estudiantes.

Los modelos de estudiante descritos pudene tener aun dos enfoques más. La division principal que depende de la cantidad de información que el STI reciba del estudiante y del tipo de conocimiento que se enseña [31]

## 3.2.1 Bandwidth

La entrada a la unidad de diagnóstico consiste de varios tipos de información acerca de lo que el estudiante hace o dice. En base a esto, el diagnóstico debe inferir lo que el estudiante piensa y conoce. Mientras menos información entre, más difícil es la tarea de diagnóstico. La clasificación en base a Bandwidth es una categoría aproximada de la cantidad y calidad de la entrada al diagnóstico.

Tres niveles de información son suficientes para mostrar las diferencias que existen entre los diferentes STI. Con el fin de explicar, se asume que los estudiantes están resolviendo problemas ya sea propuestos por ellos mismos o por el sistema. Si en la solución del problema se



# Modelado del Estudiante

lleva en muy poco tiempo, entonces se puede asegurar que el estudiante lo hace en base a una serie de estados mentales. La más grande Bandwidth que un STI puede tener es una lista de todos los estados mentales por los cuales el alumno pasa al momento de resolver un problema. Los estados mentales de los humanos no son fácilmente accesibles por medio de máquinas, por lo tanto ningún STI puede realmente acceder a este bandwidth. Sin embargo, si el STI hace bastantes preguntas, entonces puede obtener suficiente información indirecta que se aproxima a los estados mentales. Por lo tanto la más grande bandwidth es la de estados mentales aproximados.

En formas de solución de problemas más complicados, como en la solución de ecuaciones algebraicas o el juego de ajedrez, el estudiante hace cambios que son observables, que llevan al problema desde su estado inicial sin resolver hasta su estado final con solución. Como resultado se tiene una serie de estados intermedios, como por ejemplo la configuración de la mitad de un juego de ajedrez o las ecuaciones que se escriben en el transcurso de la solución de una ecuación algebraica. Algunas veces un STI tiene acceso a estos estados intermedios, y alguna veces solo puede ver el estado final - una respuesta. Las otras dos categorías de bandwidth son estados finales y estados intermedios.

Para resumir, las tres categorías son, de la más alta a la más baja, *estados mentales*, *estados intermedios* y *estados finales*. Cada categoría incluye la información de la más baja. Así los Estados mentales incluyen estados intermedios y finales. Los estados intermedios incluyen los estados finales.

La programación da ejemplos acerca de bandwidth, ya que en este tema existe un STI para cada bandwidth. Un tutor de programación en lenguajes declarativos [5], contiene un modelo detallado del proceso cognoscitivo fundamental en la habilidad de programar. El tutor usa una interfaz en base a un menú que ofrece al estudiante alternativas acerca de los objetivos que quiere realizar enseguida. Que estrategias usar, que fragmentos de código usar, y varias opciones más. El modelo provee tantas alternativas que cualquier problema que el estudiante quiera tomar está disponible. Una premisa supone que el menú no interfiere con los estados mentales que sigue el estudiante y al contrario permite al STI monitorear el proceso cognoscitivo del estudiante. La entrada al diagnóstico es una aproximación a una secuencia de estados mentales. Este tutor ilustra el bandwidth de estados mentales.

La bandwidth de estados intermedios queda ilustrada con un tutor como el anterior, solo que en este caso el STI no sabe acerca de los objetivos que el alumno quiere alcanzar. Es decir, como en los sistemas reactivos, el tutor solo respondería cuando un estado intermedio es válido o falso. Un buen ejemplo son los editores de lenguajes donde las palabras reservadas tiene una fuente o color especial. De este modo el usuario puede detectar errores (aunque un editor no es un tutor, la comparación es válida)

En contraste, el tutor PROUST [1] recibe solo el programa terminado que el alumno introduce a un compilador de PASCAL, PROUST no tiene acceso al programa incompleto ni lo puede acceder en el momento de su edición.

La dimensión Bandwidth es quizá la más importante de las tres dimensiones. Más que las otras, esta dimensión determina el algoritmo de diagnóstico.

## 3.2.2 Tipo de conocimiento

Los modelos de estudiante pueden resolver problemas que los estudiantes también resuelven por lo tanto pueden ser usados para predecir las respuestas de un estudiante. Esta es una característica que distingue los modelos de estudiante usados en un STI. Los modelos de

# Modelado del Estudiante

estudiante implantados en los primeros STI no podían generar la solución de un problema, aunque tenían la posibilidad de generar una solución correcta.

En la solución de problemas se requiere un proceso de interpretación que se aplica al conocimiento del Modelo del Estudiante y del problema. Existen dos tipos comunes de interpretación, una para conocimiento procedural y otra para el conocimiento declarativo. La interpretación para el conocimiento procedural es simple. No busca pero toma decisiones en base a conocimiento local. Es como un pequeño hombrecillo con una lámpara que solo puede ver una pequeña rama desde la posición de conocimiento en que se encuentra; basándose en el conocimiento local que tiene y en las condiciones del problema decide por que rama dirigirse. Un interprete declarativo constantemente busca sobre toda su base de conocimientos. Es como un bibliotecario que busca la petición de un usuario por medio de la búsqueda de referencias, ensamblando hechos, y deduciendo la respuesta de esto.

La diferencia entre conocimiento procedural y declarativo es notorio en Inteligencia Artificial. Ya que se basa en la cantidad de trabajo que el interprete hace y por que la calidad del trabajo es esencial, los limites entre estos no son claros ni están definidos con precisión.

La diferencia entre conocimiento procedural y declarativo es importante en el contexto de STI ya que la complejidad del diagnóstico es directamente proporcional a la complejidad de la interpretación. De hecho el diagnóstico es lo inverso a la interpretación. La interpretación toma una base de conocimientos y un problema y produce una solución. El diagnóstico toma un problema y una solución y produce una base de conocimientos. Cuando se interpreta lenguaje declarativo, muchos elementos se accesan con el fin de encontrar una solución. Cuando se diagnostica conocimiento declarativo, la posibilidad de una respuesta equivocada depende de cualquiera de los elementos que pudieron ser accesados al producir una respuesta. En general, mientras más complicada sea la interpretación, más complicado será el diagnóstico.

Estas observaciones llevan a una nueva dimensión de los modelos de estudiante, el tipo de conocimiento que tendrá el Modelo de Estudiante. La distinción principal ya ha sido mencionada - procedural , declarativa. Es útil dividir al conocimiento *procedural* en *plano* y *jerárquico*. Las representaciones jerárquicas permiten tener subobjetivos, mientras que las planas no.

La distinción entre plano y jerárquico afecta el diagnóstico. Un sistema de diagnóstico plano necesita inferir que condiciones del estado del problema cargan cada operador. Esto es fácil, ya que el sistema puede ver tanto los estados del problema y la aplicabilidad de los operadores. Un sistema de diagnóstico para representaciones jerárquicas necesita inferir condiciones tanto del problema como de los subobjetivos. Pero no puede acceder a los subobjetivos activos en el momento, por lo tanto el problema de inferencia es más difícil.

En resumen, existen tres tipo de representación del conocimiento: procedural plano, el cual hace al modelado del estudiante más fácil; procedural jerárquico, el cual incrementa la dificultad del modelado del estudiante; y declarativo, el cual hace que el modelado del estudiante sea más difícil.

# Modelado del Estudiante

## 3.3 Modelo del Estudiante

La representación más adecuada de un Modelo del Estudiante es una red semántica según [22]. Cada nodo representa un concepto conectados por la relación "es elemental para conocer" y "es similar a".

La selección de la red semántica y las relaciones favorecen a la estrategia de enseñanza en el sentido de que el tutor decide la ruta a seguir en la red (currícula) y la posible selección de concepto similares, en base a conocimiento procedural plano [31]. En la figura 2 la relación "es elemental para conocer" permite la enseñanza en base a currícula (conocimiento jerárquico procedural). La relación "es similar a" permite la enseñanza en base a múltiples opciones (conocimiento procedural plano). Este modelo se toma de una abstracción de [9] y los anteriores mencionados en este párrafo.

Aquí se escoge como relación principal "es elemental para conocer" [22]

Por otro lado se insiste en que las herramientas Multimedia no permiten una

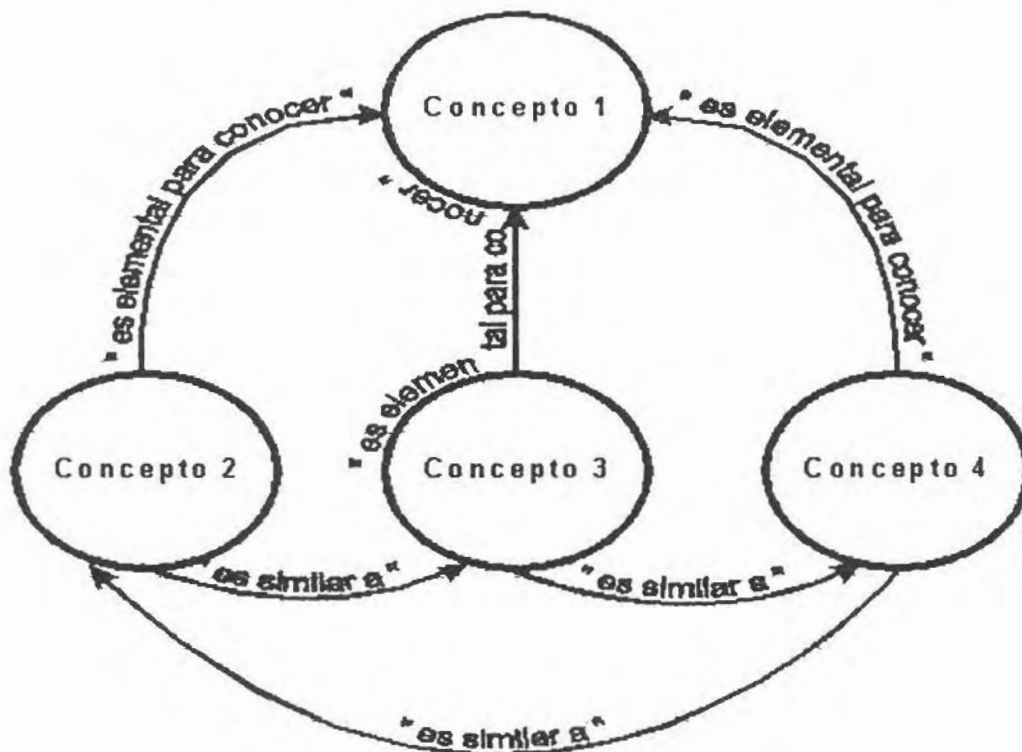


figura 2

Red semántica con las relaciones "es elemental para conocer" y "es similar a"

representación del conocimiento en base a IA. Pero estas herramienta permiten (en su mayoría) comunicación con otros programas que pueden crearse con otras herramientas o lenguajes [22]. Una red semántica puede, por ejemplo, explotarse mediante un shell para sistemas expertos basado en reglas. La mayoría de los shells para sistemas expertos tienen una interfaz (técnicamente) ya definida por lo cual los programas creados con herramientas Multimedia pueden comunicarse con estos, pero la retroalimentación no es posible. Una solución es crear programas con herramientas y/o lenguajes de bajo nivel utilizando programación procedural y/o un diseño orientado a objetos. La desventaja de esta opción es que se tienen que crear mecanismos de

# Modelado del Estudiante

inferencia (búsqueda hacia delante, búsqueda hacia atrás, en amplitud, etc.). La ventaja es que no se pretende crear todos los métodos de inferencia, sino solo aquellos que faciliten los métodos de enseñanza.

Muchos lenguajes (C/C++, BASIC, FORTRAN) y herramientas (Visual C++, Visual BASIC) permiten la creación de programas que se comunican con las herramientas Multimedia. Una de las características de estas herramienta es que permiten el diseño orientado a objetos. La Inteligencia Artificial no solo ha hecho uso de lenguajes declarativos, sino también de lenguajes procedimentales y además en los últimos años el uso de la programación orientada a objetos en IA se ha venido usando con éxito. Por lo tanto es posible mediante estos lenguajes construir programas que se comuniquen con la herramienta Multimedia y construir programas que realicen inferencias. Una red semántica se puede modelar con estos lenguajes mediante un diseño orientado a objetos. Así se tiene una relación concepto - objeto. Dentro de este objeto se pueden

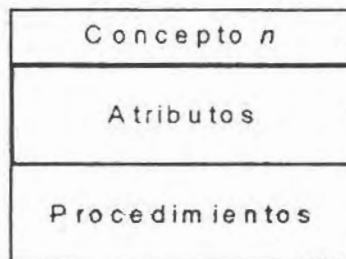


figura 3

Objeto *concepto* para representar un nodo de la red semántica

encapsular datos, procedimientos y relaciones de los conceptos. Como se puede apreciar en la figura 3

Mediante una representación de este tipo, los procedimientos encapsulados permiten realizar inferencias dentro de la red semántica que representa al estudiante. Incluso procedimientos del Modulo Tutor, como la búsqueda del objetivo pedagógico, de tal modo que un concepto representa el objetivo pedagógico y a consecuencia de las características que contiene (que en determinado momento es el Modelo del Estudiante) conduce el comportamiento de STI.

# Modelado del Estudiante

## 3.4 Algoritmo de Toma de decisiones basado en un Modelo del Estudiante

El Módulo Tutor que se propone en esta arquitectura se representa básicamente en la figura 4 . El diseño del Modulo Tutor se hace considerando al mismo tiempo el diseño del Modelo del Estudiante

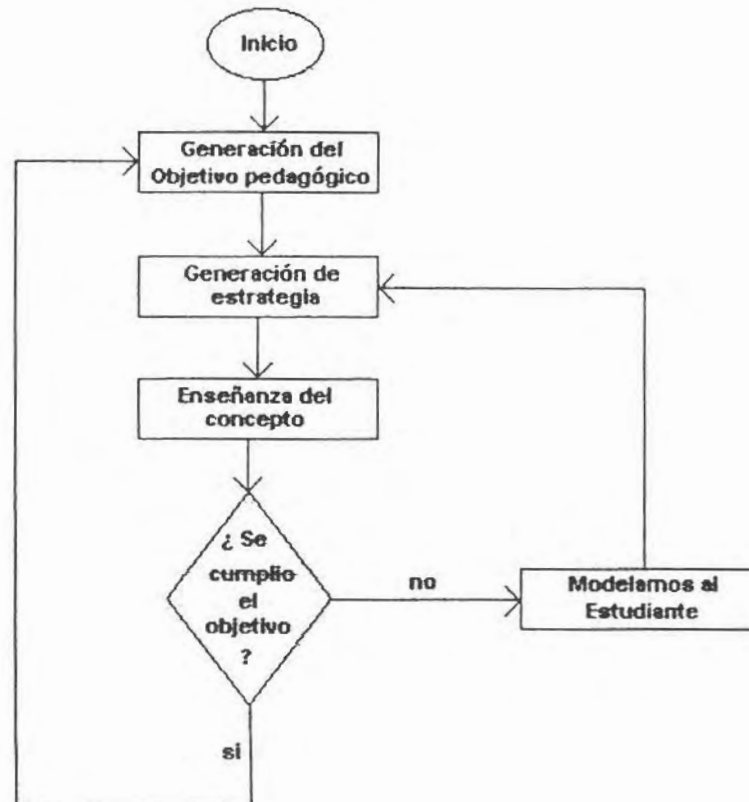


figura 4

Módulo Tutor (Administración)

El Módulo Tutor administra los procesos que lleva a cabo el sistema en general. Las funciones que desempeña y las decisiones que toma se hacen en base al Modelo del Estudiante. Al mismo tiempo este Modulo Tutor tiene la responsabilidad de generar el objetivo pedagógico del sistema.

# Modelado del Estudiante

La mayoría de las herramientas Multimedia (como Macromedia Authorware) permiten la programación en base a un flujo de procesos. Algunas de estas herramientas están enfocadas a la programación orientada a objetos, en el sentido de que encapsulan datos de objetos (audio, video, interacciones, texto) y permiten el envío de mensajes en cualquier parte del flujo hacia los iconos encapsulados.

La desventaja de estas herramientas es que en ellas no existen métodos para representar el conocimiento [35]. Por lo cual la inclusión de IA a sistemas basados en Multimedia es una tarea compleja.

Básicamente las decisiones que toma este módulo en base al Modelo del Estudiante son:

- Determinar si se examina o se diagnóstica
- Buscar el objetivo pedagógico
- Generar exámenes
- Presentar lecciones e indicadores.

En la arquitectura propuesta la estrategia de enseñanza se basa en una búsqueda del objetivo pedagógico en la red semántica. Para esto se tienen que definir los posibles estados en que se puede encontrar un concepto en determinado momento. Los estados considerados se basan en [35] y son los siguientes:

VALOR	SIGNIFICADO
2	El estudiante conoce este concepto y el sistema se ha cerciorado de este hecho.
1	El sistema mediante alguna inferencia a deducido (supuesto) que el estudiante conoce este concepto, pero no se ha cerciorado de ello.
0	El Sistema no tiene información respecto al dominio del concepto por parte del estudiante.
- 1	El estudiante no conoce este concepto y el sistema se ha cerciorado de este hecho o lo ha inferido
- 2	El sistema ha insistido en la enseñanza de este concepto y ha agotado su material de enseñanza.

Tabla 2

Significado de los valores que representan el estado de un concepto

En base a la proposición del esquema anterior se tiene que la red semántica posee conceptos con valores determinísticos. La búsqueda de un concepto para tomarlo como objetivo pedagógico esta sujeto a las relaciones de la red y a los valores que poseen los conceptos. Las relaciones y valores fueron escogidos de tal modo que formulen estrategias de enseñanza en base al Modelo del Estudiante y tomado en cuenta a la referencia [24] al definir los requisitos para un Modelo de Estudiante y a la referencia [33] cuando menciona las formas de explotar un ME determinístico.

Así se tiene que el Modulo Tutor buscara por toda la red semántica el objetivo pedagógico. La búsqueda se hace en profundidad.

# Modelado del Estudiante

## 3.4.1 Estrategias de enseñanza

Las estrategias de enseñanza son las siguientes:

- Si un concepto está en valor 2 entonces no puede ser enseñado ya que el estudiante lo conoce y el sistema verifico este hecho.
- Si un concepto está en valor 1 entonces no puede tomarse como objetivo pedagógico ya que este concepto se supone conocido por el estudiante y no es necesario enseñarlo (excepto en otras circunstancias como se vera más adelante).
- Si un concepto está en valor 0 entonces se toma como objetivo pedagógico, solo que en este caso y de acuerdo con el método de enseñanza se diagnostica primero al estudiante, ya que es mejor para el estudiante, así de este modo no le es presentada una lección si es que conoce el concepto (el STI no sabe si el estudiante domina o no el concepto, es decir existe incertidumbre).
- Si el valor del concepto es -1 quiere decir que el estudiante no conoce el concepto, por lo tanto el STI mostrara las lecciones que conllevan a la enseñanza de este concepto.
- Si el valor es -2 entonces este concepto está agotado, es decir de los recursos (lecciones) que posee el sistema para enseñar no queda ninguno. Este concepto no puede tomarse como objetivo pedagógico (salvo en otras circunstancias, como se vera más adelante) por lo cual el STI debe tomar una decisión.

Hasta aquí, el Modulo Tutor junto con el ME están definidos en forma de currícula. En donde el sistema no continuaría con el siguiente concepto hasta que el que esta como objetivo pedagógico sea dominado por el estudiante. El ME queda definido hasta aquí como una lista de elementos en donde el Modulo Tutor verifica el dominio del estudiante de cada concepto y avanza

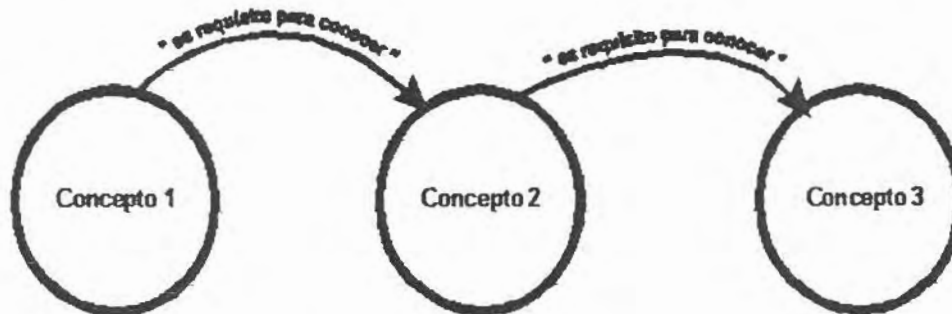


figura 5

Modelo de Estudiante exclusivo para una currícula

conforme a la lista. La muestra figura 5 como el Modelo del estudiante es una lista de conceptos.

Este Modelo se mejora con el conocimiento procedural plano [31], en donde un concepto que "es similar a " otro puede ser enseñado en vez del primero. En combinación con el conocimiento jerárquico y los valores que poseen los conceptos, permiten aumentar las estrategias de enseñanza, así se tienen diferentes situaciones en que el Modelo del Estudiante se encuentra en determinado momento.

# Modelado del Estudiante

Una parte del Modelo del Estudiante (Red semántica) se presenta en la figura 6, en esta figura se representan los posibles estados en que se puede encontrar un concepto y su relación a los conceptos más cercanos.

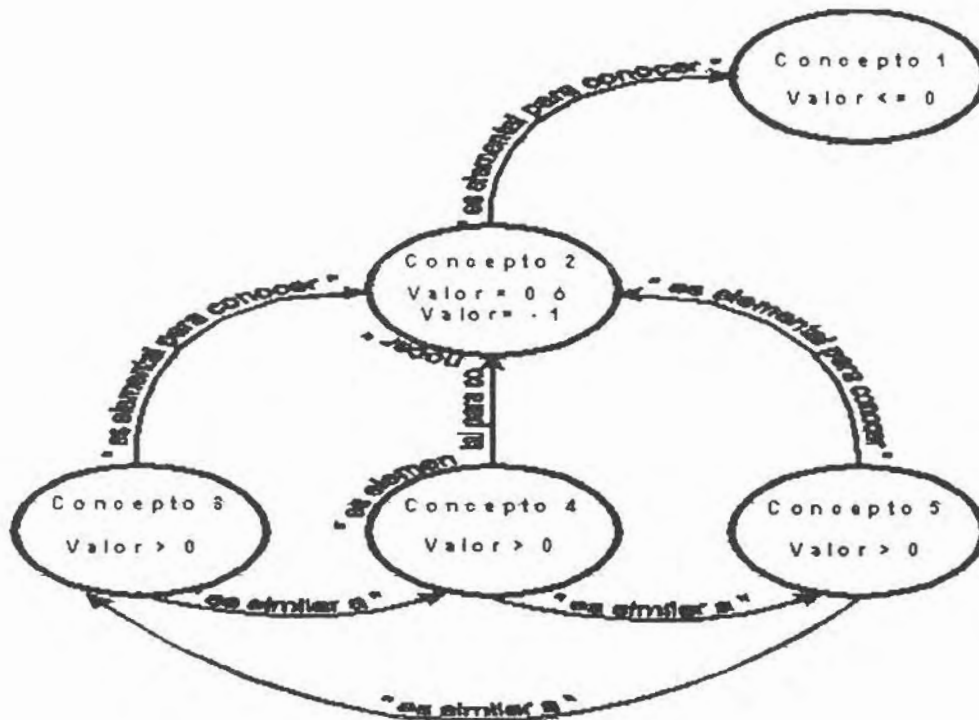


figura 6

Búsqueda por la relación "es elemental para conocer"

Aquí el concepto 1 es más complejo que los demás, por lo tanto la búsqueda continúa por la relación "es elemental para conocer a", cuando el Modulo Tutor encuentra el concepto 2 reconoce que este puede tomarse como objetivo pedagógico. En este punto el algoritmo verifica que todos los conceptos relacionados con el concepto 2 por la relación "es elemental para conocer a" sean del dominio del estudiante, ya sea que el STI haya verificado su dominio o lo haya asumido mediante una inferencia. Dadas estas premisas el concepto 2 se toma como objetivo pedagógico y de acuerdo a su valor el Tutor decide si enseñar el concepto (valor = 1) o diagnosticarlo (valor = 0).



# Modelado del Estudiante

El caso de la figura 7 es un caso parecido. Aquí el concepto 3 se toma como objetivo pedagógico. Los conceptos 4 y 5 los conoce el estudiante o el sistema los asume por conocidos. Pero el concepto 3 no se conoce, entonces el Modulo Tutor decide tomar este concepto como objetivo pedagógico ya que cuando el estudiante domine este concepto podrá continuar con el concepto 2. El concepto 2 no puede tomarse como objetivo pedagógico ya que el Modulo Tutor reconoce que el estudiante aún no domina los conceptos indispensables (concepto 3, concepto 4, concepto 5) para conocer el concepto 2.

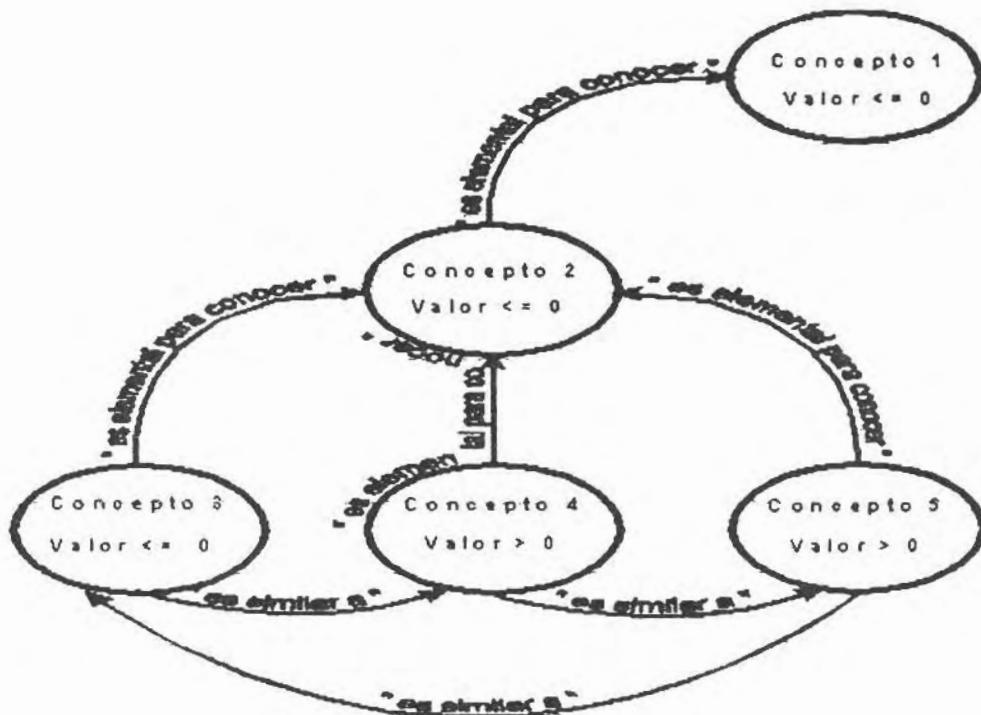


figura 7

Búsqueda de un concepto en profundidad por la relación "es elemental para conocer"

# Modelado del Estudiante

La figura 8 muestra el siguiente caso. Se tiene el concepto 3 como objetivo pedagógico, y el STI ha mostrado las lecciones correspondientes para que el estudiante domine este concepto. El Modulo Tutor puede asumir la postura de ser lo suficientemente "paciente" para continuar mostrando lecciones hasta que el estudiante domine el concepto. Aquí se implanta otra estrategia. Tomando la relación "es similar a" el Modulo Tutor busca un concepto por esta relación que pueda ser tomado como objetivo pedagógico. El Modulo Tutor verifica que los conceptos conectados por esta relación puedan ser tomados como objetivos pedagógicos en este caso pueden tenerse varios escenarios en los cuales el Modulo Tutor toma decisiones

La misma figura 8 muestra otro caso. Aquí el Modulo Tutor no encuentra un concepto que pueda tomarse como objetivo pedagógico. En este caso los conceptos similares ya son del

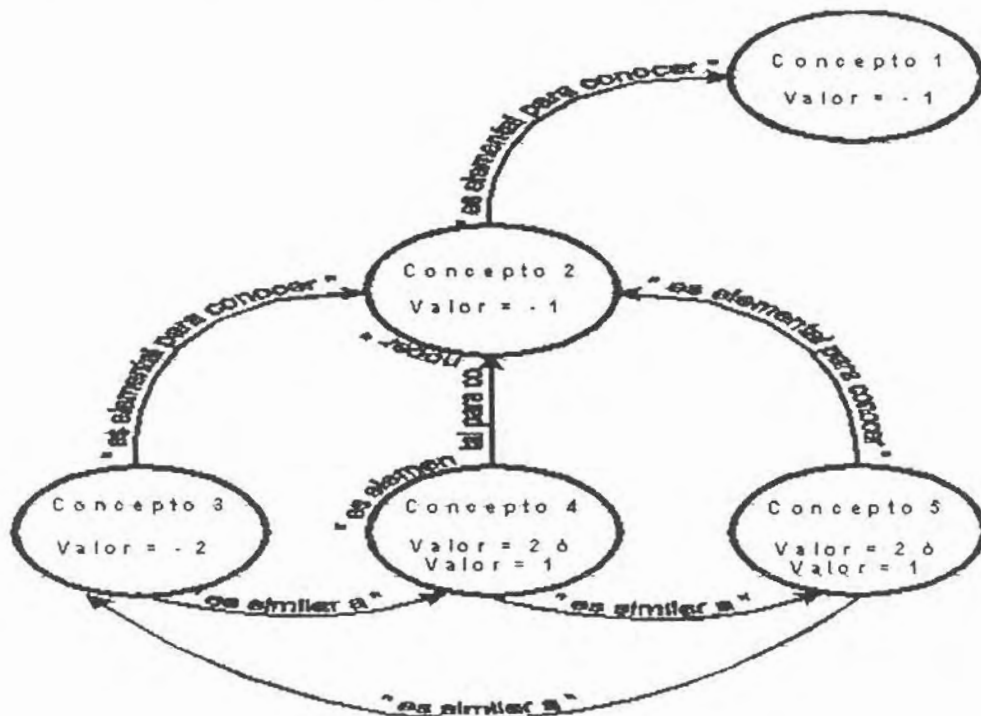


figura 8

Fallo en la búsqueda de conceptos similares, el Modulo Tutor continua con el mismo concepto

dominio del estudiante, entonces el Modulo Tutor continua con el mismo objetivo pedagógico

# Modelado del Estudiante

La figura 9 muestra el caso en que el Modulo Tutor encuentra un concepto similar que puede ser enseñado. En este caso el Modulo Tutor toma el concepto 4 como objetivo pedagógico. Si el alumno domina el concepto durante el transcurso de la enseñanza, entonces el Modulo Tutor continua nuevamente con el concepto 3 como objetivo pedagógico.

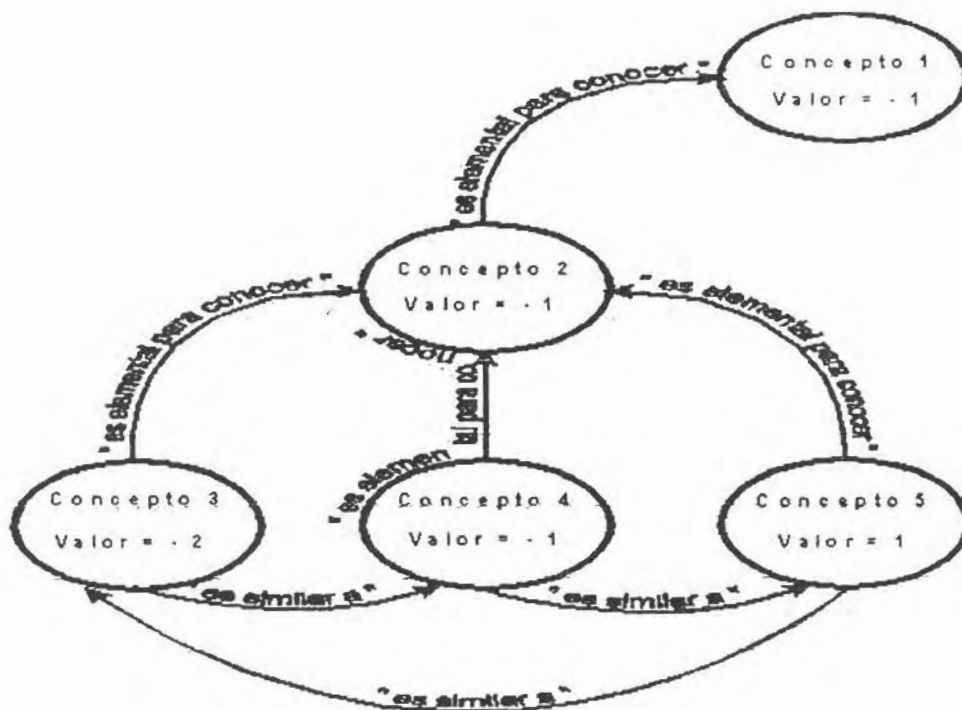


figura 9

Éxito en la búsqueda de un objetivo pedagógico por la relación "es similar a"

# Modelado del Estudiante

La figura 10 muestra el caso en que, tomando como premisa el caso anterior, aún existe la posibilidad de que los conceptos similares sean enseñados sin éxito. En este caso el Modulo Tutor pone a todos los conceptos similares al valor = -1. Es decir, supone que el estudiante necesita que todos los conceptos similares le sean enseñados

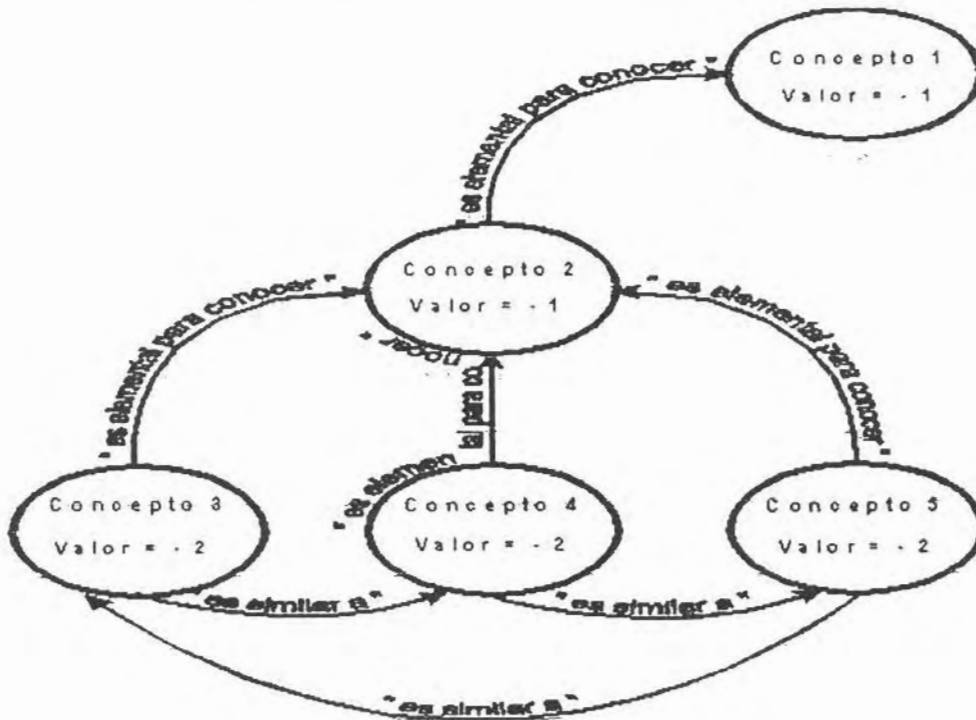


figura 10

Todos los conceptos por la relación "es indispensable para conocer" están agotados.

# Modelado del Estudiante

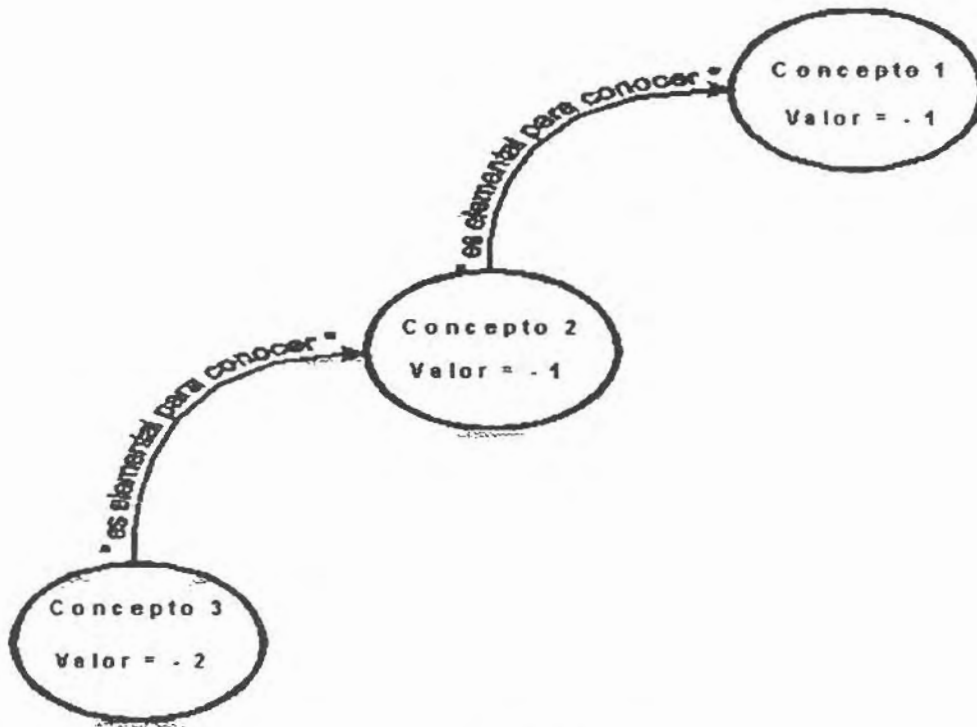


figura 11

Conceptos que no tienen conceptos relacionados por la liga "es similar a"

La figura 11 muestra un caso particular en el que el objetivo pedagógico (concepto 3) no tiene conceptos similares, como se puede apreciar en la figura.

En este caso el Modulo tutor continua enseñando el mismo concepto, ya que este concepto debe ser dominado por el estudiante antes de que el tutor enseñe el concepto 2. Aquí el Modulo Tutor solo tiene la posibilidad de seguir bajo la estrategia de currículo ya que el concepto 3 solo presenta esa característica al no tener asociado conceptos similares.

En muchas áreas académicas un concepto puede quedar en esta posición. Por ejemplo, el cálculo diferencial debe ser comprendido (dominado) antes del cálculo integral.

Puede suceder que el estudiante agote las lecciones que corresponde a un concepto. En este caso el Modulo Tutor busca por la relación "es similar a" la manera de encontrar un objetivo pedagógico o de modificar el Modelo del Estudiante.

# Modelado del Estudiante

La figura 12 muestra el caso en que, tomando como premisa que se tomó al concepto 2 como objetivo pedagógico y en el transcurso de la enseñanza fue agotado. El Modulo Tutor busca por la relación "es indispensable para conocer a" algún concepto que haya asumido como conocido por el estudiante (valor = 1) si encuentra un concepto bajo estas condiciones entonces pasa su valor a -1 y de este modo los enseña, ya que muy probablemente el alumno no domine estos conceptos y es la causa de que no domine el concepto 2.<sup>1</sup>

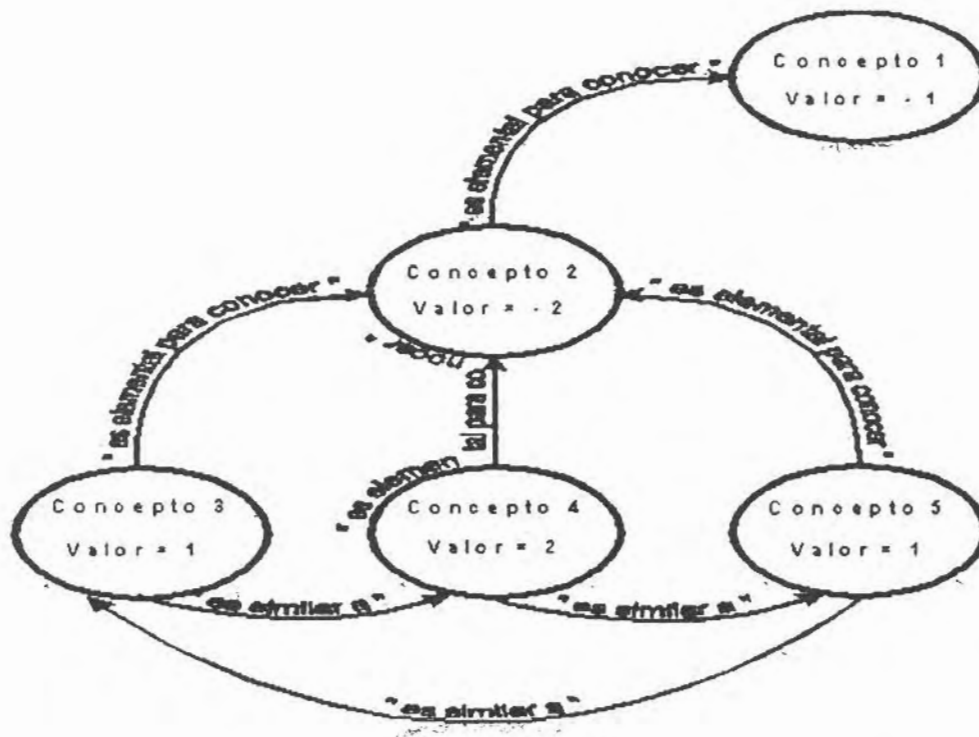


figura 12

Búsqueda de conceptos asumidos por conocidos

<sup>1</sup> Esta es una aseveración subjetiva. Muchos educadores pueden tomar otra decisión al respecto. Las ciencias cognitivas son un campo abierto.

# Modelado del Estudiante

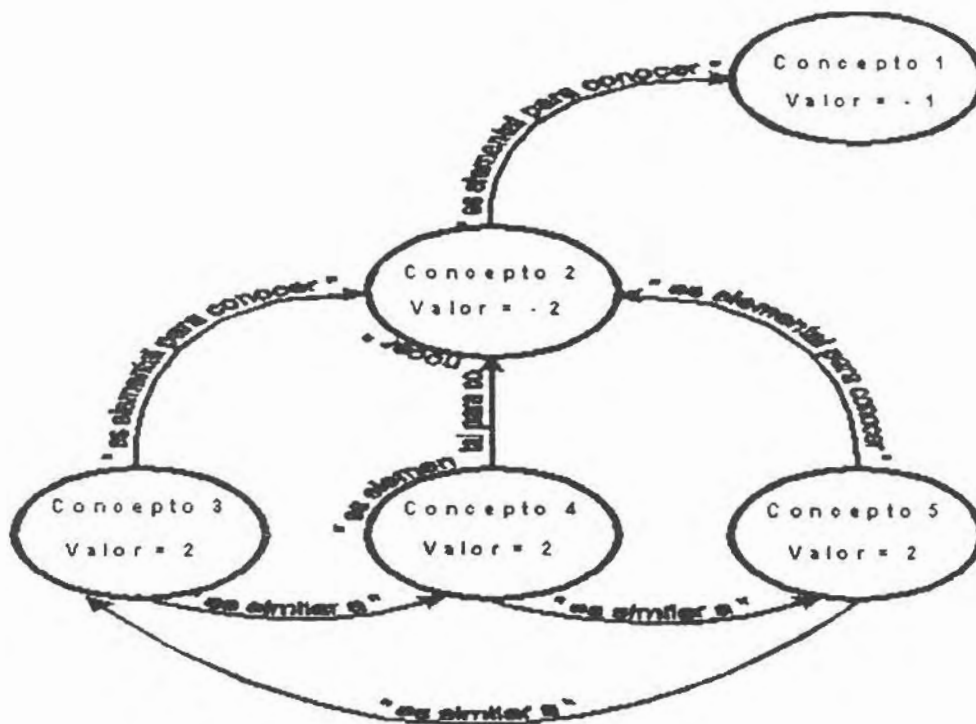


figura 13

Fallo en la búsqueda de conceptos asumidos por conocidos

En el caso de que el Modulo Tutor no encuentre conceptos con la relación "es indispensable para conocer a" y que hayan sido asumidos como conocidos por el estudiante entonces el Modulo Tutor continua con la enseñanza del concepto 2. La figura 13 ilustra este caso.

El Modulo Tutor antes de tomar un concepto como objetivo pedagógico verifica que el concepto pueda ser enseñado. Mediante esta estrategia hace una búsqueda en toda la red. La forma en que el Modulo tutor verifica es mediante la relación "es indispensable para conocer a". Si todos los conceptos ligados por esta relación al posible concepto que se tomara como objetivo pedagógico poseen el valor 2 ó el valor 1 (el STI verifico el dominio del alumno de este concepto ó lo asume como conocido) entonces este concepto se toma como objetivo pedagógico. Si alguno de los conceptos ligados por esta relación no poseen estos valores entonces busca a través de estos conceptos hasta encontrar uno en donde se verifique la enseñanza.

# Modelado del Estudiante

En la figura 14 se muestra cuando el Modulo Tutor verifica la enseñanza del concepto 2 y lo toma como objetivo pedagógico.

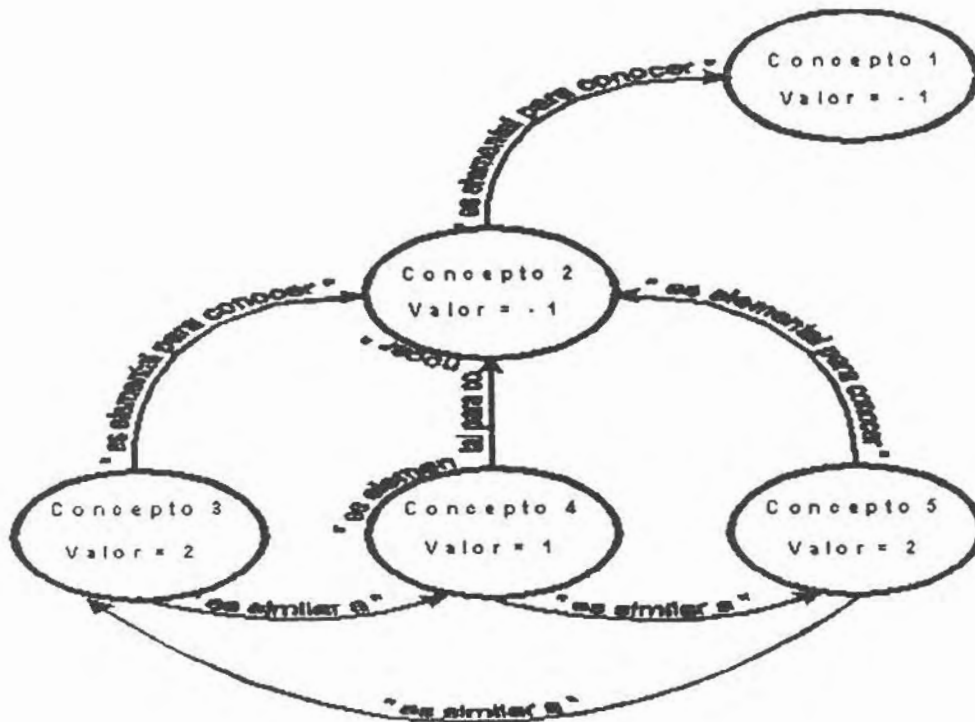


figura 14

Verificación de la enseñanza de un concepto



# Modelado del Estudiante

En la figura 15 se muestra el caso en que el Modulo Tutor no puede tomar al concepto 2 como objetivo pedagógico y continua su búsqueda en amplitud por la relación "es indispensable para conocer". Cabe señalar que en el caso particular de la figura, el concepto 3 es un nodo hoja (respecto a la liga "es elemental para conocer") y es el que se toma como objetivo pedagógico.

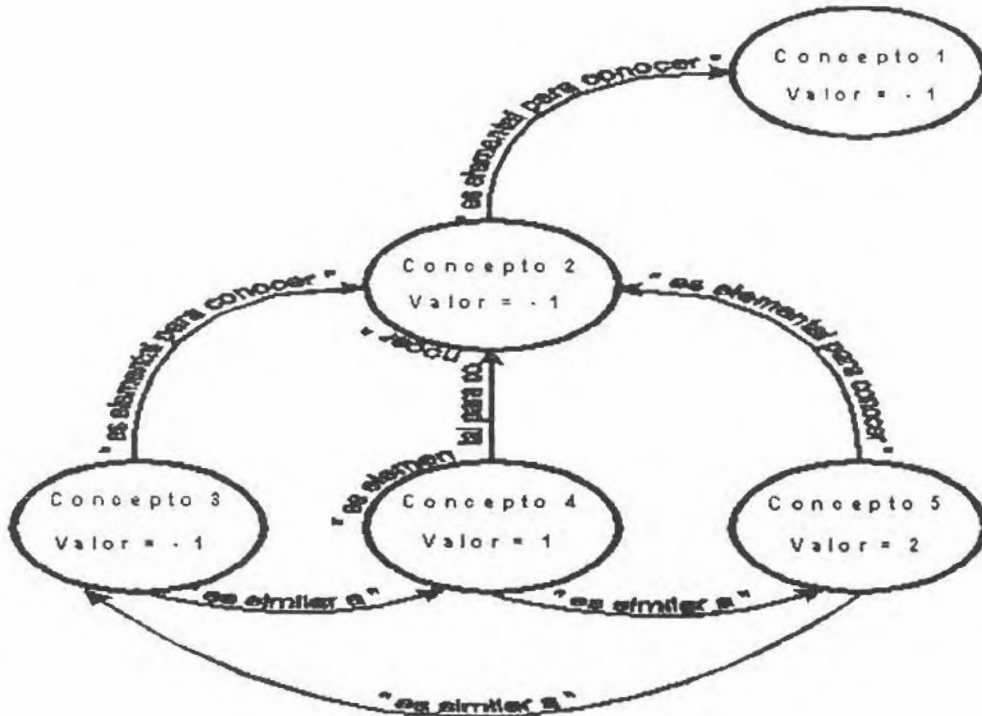


figura 15

Continuación de la búsqueda del objetivo pedagógico en profundidad al no verificarse la enseñanza de un concepto

# Modelado del Estudiante

En la figura 16 se muestra un caso general de búsqueda del Módulo Tutor. Aquí el concepto 2 se prueba como objetivo pedagógico pero no se puede tomar, entonces se prueba el concepto 3 que nuevamente no se puede tomar, luego el Módulo Tutor continúa su búsqueda hacia las hojas de la red hasta encontrar un concepto que pueda tomarse como objetivo pedagógico. En este caso el Módulo tutor termina su búsqueda en el concepto 7.

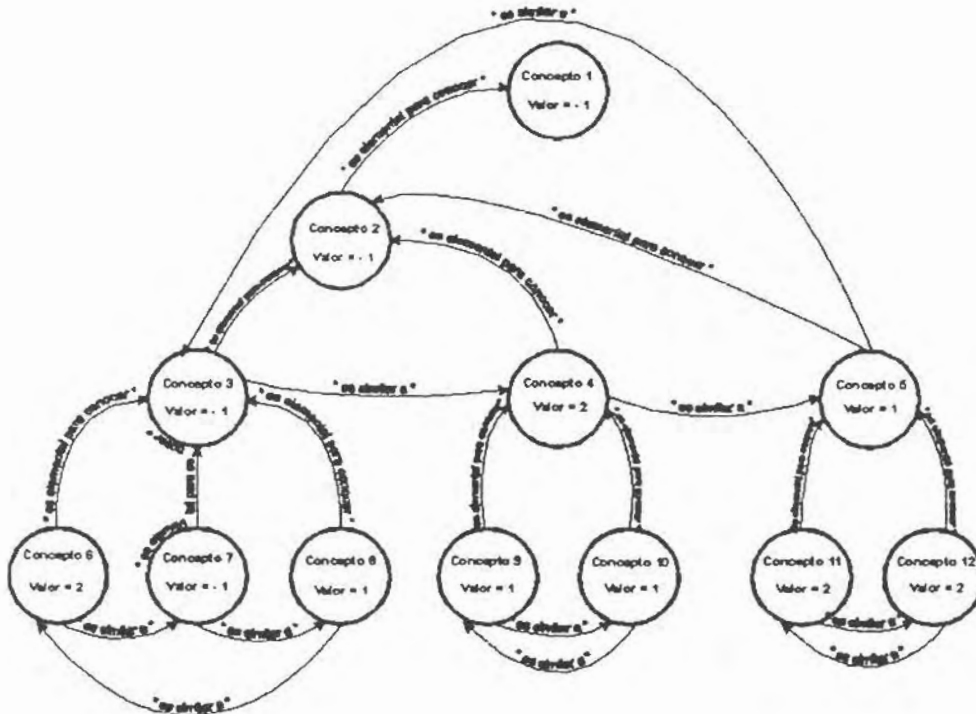


figura 16

Caso general de búsqueda del objetivo pedagógico

La construcción de la red semántica es única para cada dominio del conocimiento que se tome y en general su profundidad, amplitud y granularidad se analizan con la colaboración de un pedagogo y el experto en el dominio en el tema.

Los algoritmos presentados funcionan en cualquier diseño que se tome. Bajo las siguientes restricciones:

- Solo una red semántica debe ser tomada como Modelo del Estudiante.
- Si un concepto está ligado a otros mediante la relación "es similar a" forzosamente debe tener un nodo padre que lo relacione mediante la relación "es indispensable para conocer a".

Las estrategias funcionan aún en el caso en que se tenga solamente una currícula o solo un concepto.

# Modelado del Estudiante

## 3.5 Inicio del Modelo del Estudiante

El iniciar el Modelo del Estudiante es una tarea que requiere de especial atención. Cada estudiante que se presenta frente al sistema es diferente (las diferencias están basadas respecto al dominio que tiene del tema), es decir, el sistema debe adaptar su enseñanza a las necesidades del usuario. En general un estudiante cuyo dominio del tema es limitado (por ejemplo, es la primera vez que se le presenta el tema) necesita que todos o la mayoría de los conceptos le sean enseñados. En cambio un estudiante cuyo dominio del tema es mayor (o incluso total) se le deben enseñar pocos conceptos. La forma de iniciar el ME puede basarse en varias técnicas ya sea estadísticamente, haciendo un examen previo o por medio de la formación académica del estudiante (ingenieril, técnica, etc.) [33].

El método que sigue el algoritmo propuesto se basa en métodos probabilísticos, en un examen inicial para el estudiante y en base a las siguientes premisas:

- Se hace un examen para iniciar el ME
- El alumno solo hará una vez el examen
- Por cada alumno que ingrese al sistema el examen debe ser diferente
- Si un alumno decide no hacer un examen entonces el sistema asume que desea conocer todos los conceptos.
- El sistema debe guardar la información de cada alumno, de tal modo de que cuando el estudiante vuelva a ingresar al curso el sistema ya posea información respecto al mismo.

Ahora bien, se tienen todos los conceptos relacionados a preguntas, de ahí se toman las preguntas necesarias para realizar el examen que inicia el Modelo del Estudiante. La selección de los conceptos por cada examen que realice el sistema debe ser diferente.

Un método de selección aleatoria de los conceptos garantiza que los exámenes que se lleven a cabo sean diferentes. De un total  $n$  de conceptos se seleccionan  $m$  conceptos para llevar a cabo un examen inicial. La relación  $m/n$  debe analizarse en colaboración con el pedagogo y el experto en el dominio del tema ya que si la relación es muy pequeña el Modelo del Estudiante se inicia con muy poca información y si la relación es muy grande resulta en un examen casi completo del tema, esto daría como resultado un Modelo del Estudiante con información completa pero si el número  $n$  es muy grande (100 conceptos por ejemplo) el examen es un recurso largo y tedioso para el estudiante [33].

Ahora, una vez seleccionado el número  $m$  de conceptos que son examinados se escoge al azar<sup>2</sup> un concepto y se examina. Después el concepto seleccionado se elimina de la posible selección con el propósito de que no se tome nuevamente en el examen. Del resto de conceptos se selecciona nuevamente uno al azar y se examina. Este proceso continúa hasta que los  $m$  conceptos que conforman el examen son seleccionados.

Al empezar el sistema se inicia el Modelo del Estudiante con el valor de los conceptos igual a cero (valor = 0) y si el estudiante decide no hacer el examen inicial entonces el Modelo del Estudiante se inicia sin información respecto al alumno. En el examen las respuestas del alumno tienen repercusión de acuerdo a lo que el sistema conoce de él. Mediante el Modelo del Estudiante el STI posee información del alumno, por lo cual esta información se debe actualizar de acuerdo a las acciones del alumno. En el examen una respuesta incorrecta por parte del alumno debe quedar reflejada en el Modelo del Estudiante lo mismo que una respuesta correcta.

<sup>2</sup> La generación de números aleatorios es un tema aislado del propósito de esta tesis. Aquí suponemos que la generación de un número aleatorio es por medio de una función cuya distribución de probabilidad es lineal.

# Modelado del Estudiante

El Modelo del Estudiante es una red semántica. Cada respuesta verdadera implica una propagación de valores por la relación "es elemental para conocer a", lo mismo que una respuesta falsa por parte del alumno. La propagación de estos valores, solo se hace por medio de esta relación y se fundamenta en lo siguiente:

Si un concepto A cuyos fundamentos (conceptos por la relación "es elemental para conocer a") están en los conceptos B y C entonces el dominio del concepto A implica el dominio de los conceptos B y C. La doble implicación no es cierta en todos los casos ya que un alumno puede dominar el concepto B sin dominar el concepto A. Sin embargo en este algoritmo se tomará como cierto, hasta que durante el examen no se demuestre lo contrario.

La relación "es similar a" no se explota durante el examen.

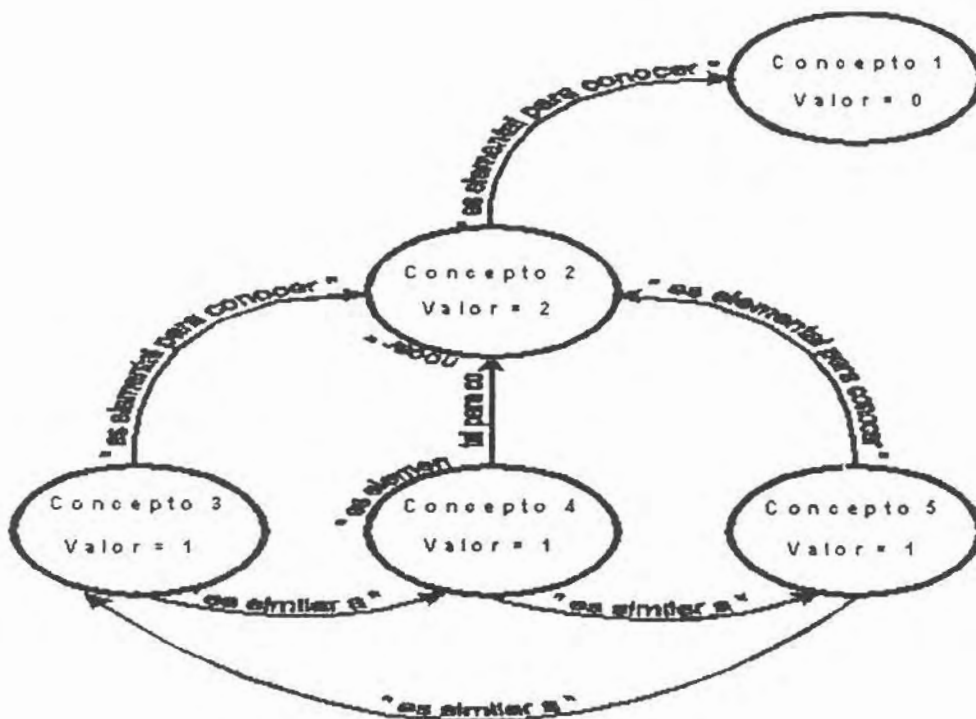


figura 17

Propagación cuando la respuesta es verdadera

Así se tienen los siguientes dos casos, en cada uno de los cuales el mismo concepto se examina. Cuando la respuesta del alumno es verdadera, el Módulo Tutor debe propagar el valor verdadero (1 ó 2) en todos los conceptos ligados por la relación "es elemental para conocer a". La figura 17 muestra el caso en que el concepto 2 se examina y la respuesta por parte del alumno es verdadera. entonces al concepto 2 se le asigna el valor = 2, ya que el sistema verifico que el estudiante domina el concepto (por medio del examen). Los conceptos relacionados al concepto 2 (los conceptos 3,4 y5) se les asigna el valor = 1, es decir el sistema asume que el estudiante domina estos conceptos, pero no lo verifico.

# Modelado del Estudiante

En el examen, si el alumno da una respuesta falsa a una de las preguntas entonces el Módulo Tutor propaga el valor falso (-1) en los conceptos que están ligados bajo la relación "es elemental para conocer a". En la figura 18 se muestra el caso en que el concepto 2 es examinado y la respuesta por parte del alumno es falsa. En este caso al concepto 2 se le asigna el valor -1 y se infiere que el alumno no conoce el concepto. Los conceptos ligados por la relación "es elemental para conocer a" También se les asigna el valor -1. Una de las razones es la siguiente: la red puede ser tan profunda que el dejar los valores de los conceptos relacionados a NOHAYINFO conduce a un ME incongruente también. La otra razón importante es ; Un alumno al ingresar al sistema lo hace con la intención de aprender, por lo cual el cuidado que tenga al responder a una pregunta es grande luego una respuesta equivocada por parte del alumno (respecto

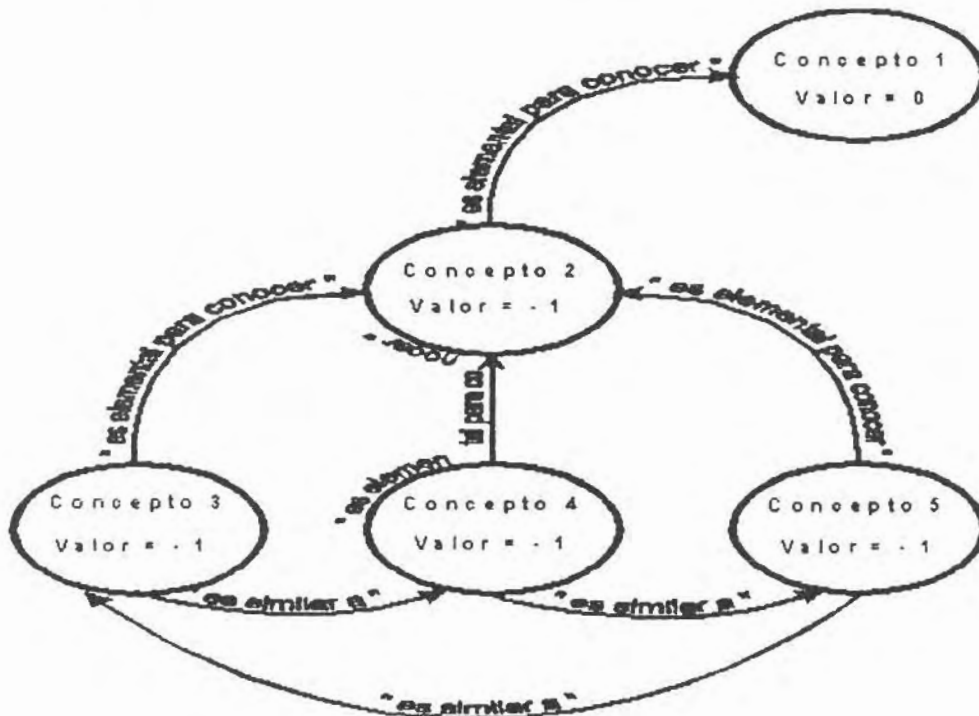


figura 18

Propagación cuando la respuesta es falsa

al conocimiento del alumno) es muy pequeña. Luego así la propagación de este modo se justifica.

Mediante estos dos métodos para propagar los valores se inicia el Modelo del estudiante. la forma en que se seleccionan los conceptos a examinar, garantiza que un concepto no se examina más de una vez. Existe la posibilidad de que un concepto a examinar sea padre o hijo de un concepto ya examinado ( si la red tiene una profundidad mayor puede ser un descendiente más profundo o un ascendiente más general). Entonces la propagación de los valores se debe controlar de la siguiente manera:

# Modelado del Estudiante

Cuando se propague un valor, ya sea por una respuesta falsa o verdadera, si al momento de descender por la red se encuentra un concepto que ya fue examinado entonces la propagación no continua por este concepto (y en consecuencia por los hijos de este). Así de este modo los conceptos cuyo dominio del alumno ya han sido inferidos no serán modificados. La figura 19 muestra el caso en que el concepto 1 ya fue examinado, obteniendo una respuesta falsa por parte del alumno. Aquí en los conceptos hijo se infirió el valor = -1. Después se examina el concepto 2, obteniendo una respuesta verdadera por parte del estudiante. Ahora, a partir del concepto 2 se modifica hacia los descendientes el valor = 1 ya que ninguno de los conceptos donde se propaga el valor ha sido examinado.

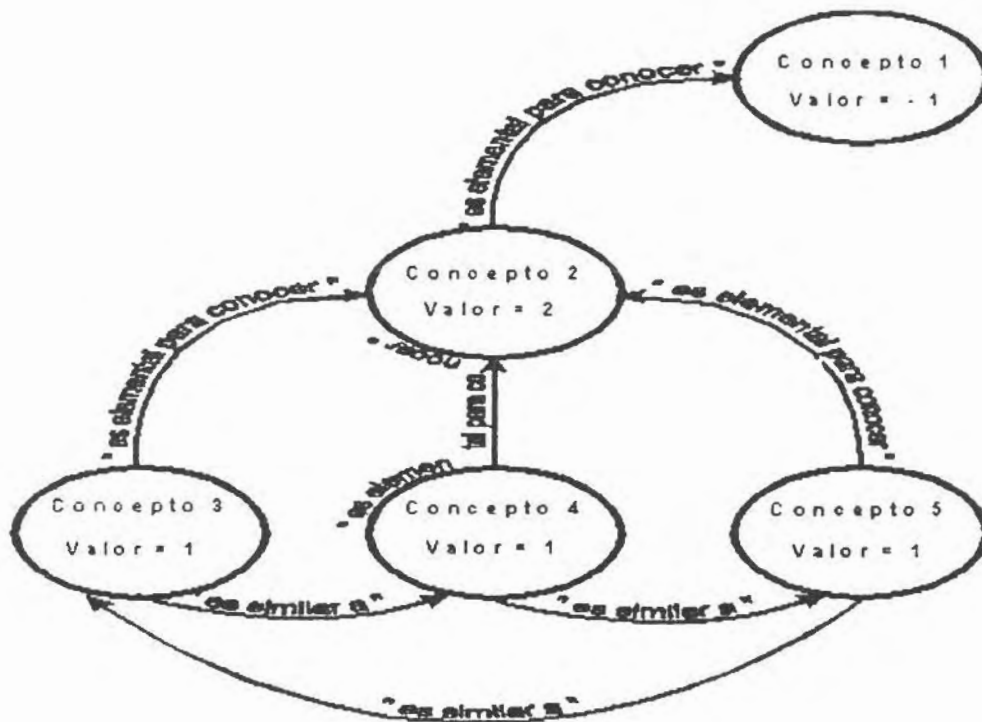


figura 19

Propagación condicional

# Modelado del Estudiante

En la figura 20 se tiene el caso contrario. Tomando solo en cuenta el orden en que fueron seleccionados los conceptos para el examen. Aquí primero fue seleccionado el concepto 2 teniendo una respuesta verdadera por parte del alumno. Después fue seleccionado el concepto 1 teniendo una respuesta falsa por parte del alumno. Al momento de que el Módulo Tutor propaga valor Falso llega al concepto 2 y reconoce que este concepto ya había sido examinado por lo cual no propaga valor alguno por este nodo.

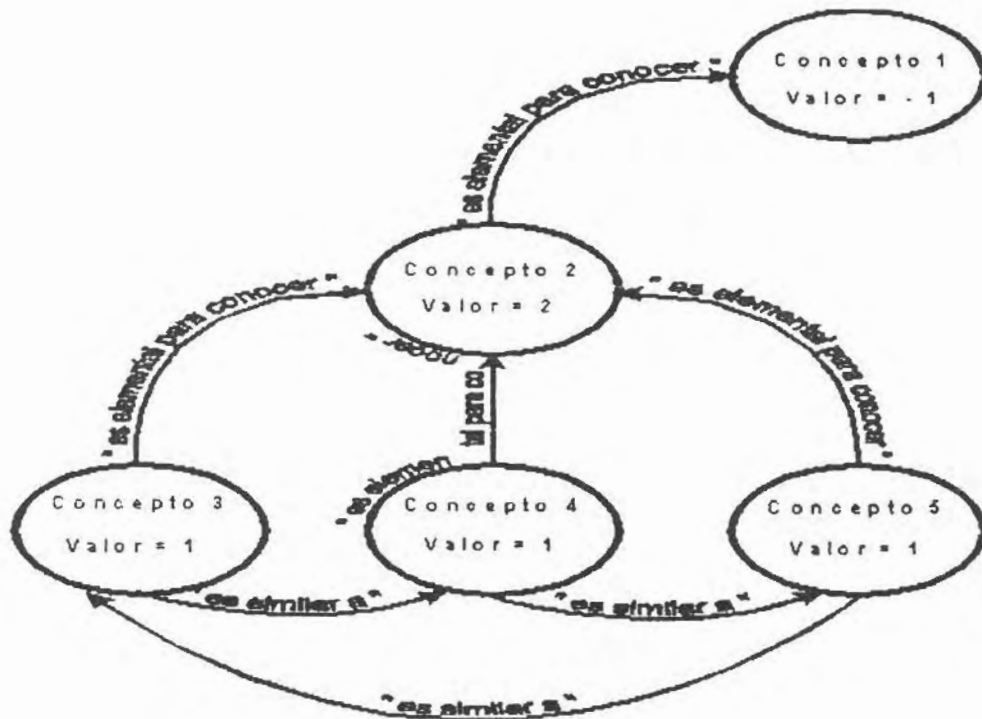


figura 20

Propagación condicional (continuación)

Los dos casos anteriores llevan a un modo de inicio idéntico. Además de ser congruente ya que conceptos más complejos no son dominados por el alumno y conceptos menos complejos no lo son. Esto es válido en cualquier estudiante y dominio del conocimiento.

# Modelado del Estudiante

Pero se podría en determinado momento tener un inicio no congruente. La figura 21 muestra el caso en que se llega a una incongruencia. Aquí, independientemente del orden en que se toman los conceptos a examinar se llega a la situación en que el concepto 1 fue examinado obteniendo una respuesta verdadera por parte del alumno, después fue examinado el concepto 2 obteniendo una respuesta falsa por parte del alumno. En ambos momentos la propagación de los valores se hizo respetando el hecho de que por un concepto examinado el valor no se propaga. Si el orden en que son seleccionados los conceptos es primero el concepto 2 y después el concepto

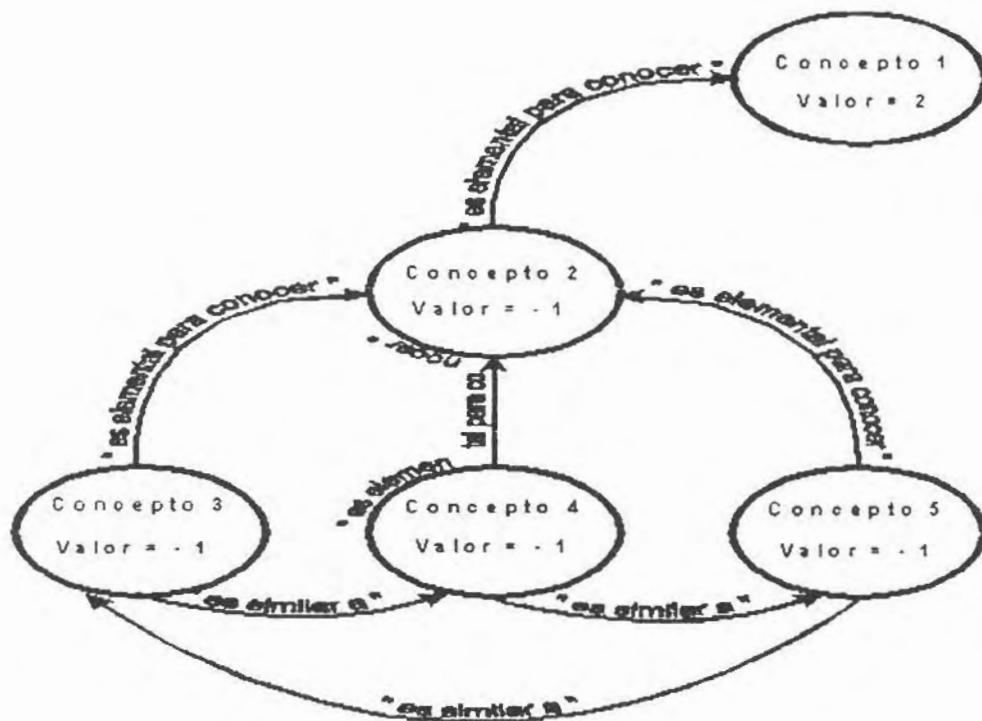


figura 21

Modelo del Estudiante incongruente

1 la configuración es la misma.

Continuando con la figura 21 se tiene que un concepto más complejo es dominado por el alumno mientras que conceptos más sencillos no lo son. Esto no es válido, en general en los diferentes dominios del conocimiento, el conocimiento de un dominio complejo implica el dominio de otros menos complejos. En el Modelo del Estudiante propuesto, este hecho se refleja en los conceptos y por medio de la relación "es elemental para conocer a". Por lo cual la figura 21 no es congruente.

El algoritmo propuesto, durante el examen sigue los métodos anteriores, durante este proceso pueden quedar incongruencias en el Modelo del Estudiante. Entonces el Tutor espera hasta que el examen haya concluido, una vez concluido el examen el Módulo Tutor hace un ajuste del Modelo del Estudiante.



# Modelado del Estudiante

## 3.6 Ajuste del Modelo del Estudiante

Al hacer el ajuste, una vez finalizado el examen, el Módulo Tutor hace un recorrido en profundidad por toda la red, buscando situaciones incongruentes. Aquí se pueden presentar dos casos.

El primer caso se muestra en la figura 22. Aquí un concepto fue examinado y la respuesta del estudiante es verdadera, luego el valor que se le asignó a este concepto es 2. Sus descendientes pueden tener cualquier valor, el Módulo Tutor al hacer el ajuste solo busca valores iguales ó menores a 0, si la búsqueda es exitosa, entonces al valor del concepto padre le asigna el valor 0. Esto debido a lo siguiente: Si un concepto es del dominio del estudiante y fue corroborado por el sistema y además uno de los conceptos que está ligado por la relación " es indispensable para conocer a " no es del dominio del estudiante, entonces el sistema no posee información suficiente acerca de este concepto. Así en la figura 22 el concepto 1 tiene el valor de 2 pero uno de sus descendientes tiene el valor igual a -1 por lo tanto y siguiendo la premisa de este párrafo el Módulo Tutor asigna el valor de 0 al concepto 1.

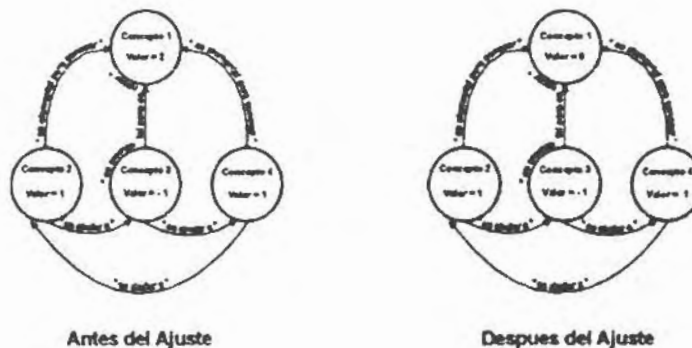


figura 22  
Ajuste a NOHAYINFO

# Modelado del Estudiante

El segundo caso se muestra en la figura 23, aquí un concepto tiene el valor igual a 1, puesto que en el examen se supuso que el estudiante domina el concepto. Sus descendientes pueden tener cualquier valor, el Modulo Tutor al hacer el ajuste busca valores menores ó iguales a 0, si la búsqueda es exitosa entonces al valor del concepto padre le asigna el valor -1. Esto de acuerdo a la siguiente premisa: Si un concepto se asume conocido por el estudiante y uno de los conceptos que esta ligado por la relación "es indispensable para conocer a" no es del dominio del estudiante entonces el concepto no es del dominio del estudiante. Así en la figura 23 el concepto 1 tiene el valor de 1 por lo tanto y siguiendo la premisa de este párrafo el Módulo Tutor mediante el ajuste asigna el valor de -1 al concepto.

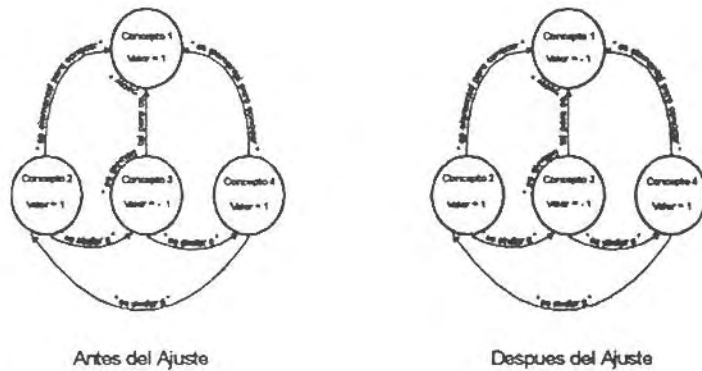


figura 23  
Ajuste a NOCONOCE

De esta manera la red es ajustada. Debe preverse que un recorrido de la red en profundidad para realizar el ajuste conduce a resultados diferentes si el recorrido se hace en amplitud.

# Modelado del Estudiante

## 3.6.1 Diferencia entre ajuste de primero en amplitud y primero en profundidad

El ajuste se debe de hacer en profundidad. Es decir llegar hasta un concepto que solo tiene hijos hoja y desde ahí hacia la raíz de la red realizar el ajuste. Mediante una búsqueda de primero en profundidad se logra que el algoritmo funcione. Ya que si el algoritmo empieza la

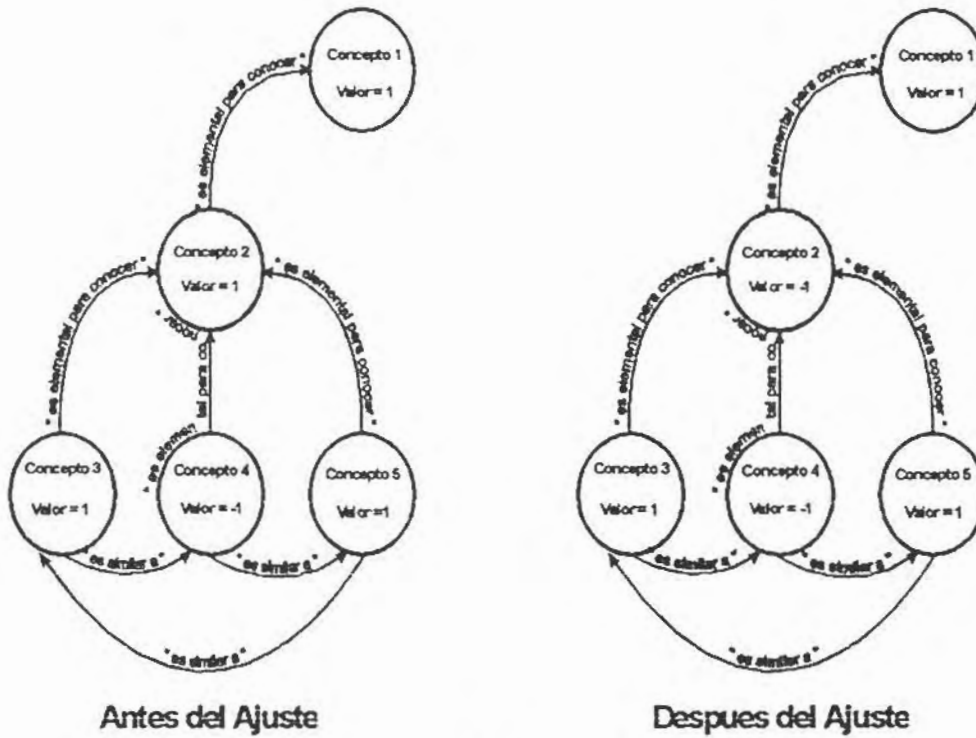


figura 24

Ajuste usando primero en amplitud

búsqueda de incongruencias desde la raíz entonces no se logra un ajuste completo y congruente como se muestra en la figura 24.

Aquí aún después del ajuste del Módulo Tutor se encuentra otra incongruencia debido a que la búsqueda es de la raíz hacia las hojas, por lo tanto la incongruencia no se corrige.

# Modelado del Estudiante

En la figura 25 se muestra la misma situación inicial de la figura 22 solo que aquí el Modulo Tutor sigue una búsqueda en profundidad

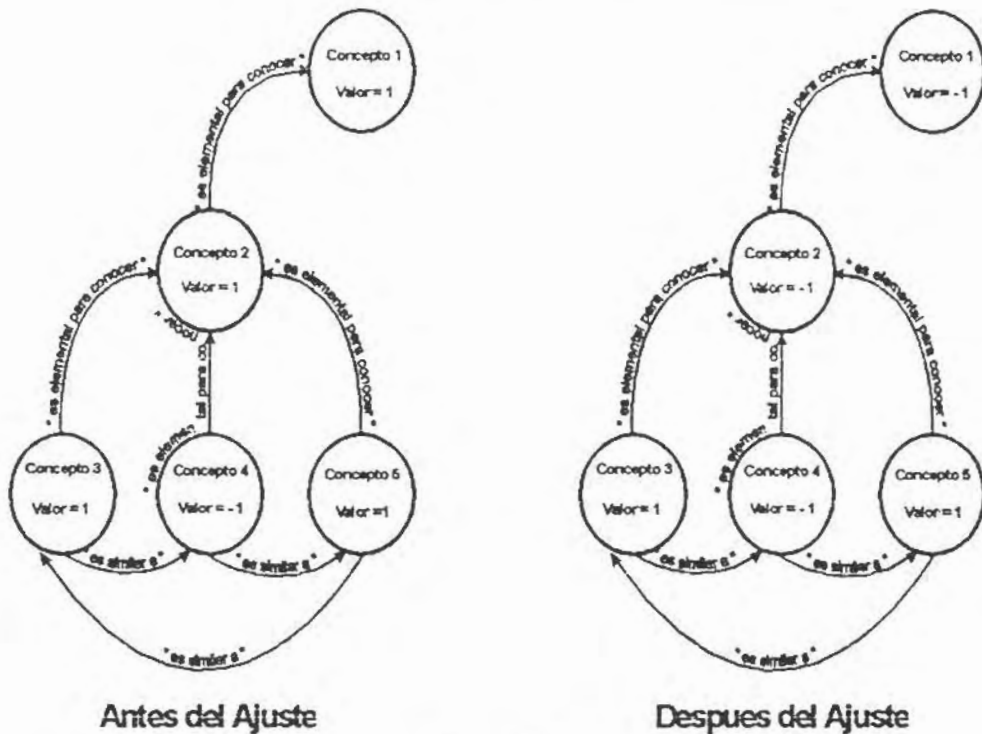


figura 25

Ajuste usando primero en profundidad

En resumen en el examen el sistema inicia el Modelo del Estudiante mediante respuestas que da el alumno. Cada respuesta propaga un valor en el ME. Al finalizar el examen el Módulo Tutor realiza un ajuste del ME en busca de incongruencias.

De este modo el Modelo del Estudiante queda iniciado y de acuerdo al algoritmo propuesto, este examen se hace solo una vez. Por lo cual el sistema debe conservar el Modelo del Estudiante con el fin de que en las siguientes sesiones en que el estudiante acceda al sistema ya se posea una referencia de él.

Durante el examen solo se hace uso de la relación "es elemental para conocer a", la relación "es similar a" no se usa. Esta relación es útil cuando el Sistema ya esta funcionando y cuando se cambia de concepto en la enseñanza.

Ahora bien, se aprecia que el Módulo Tutorial tiene varias funciones a desempeñar. Todas basadas en el Modelo del Estudiante. Como se había mencionado antes, el diseño del modelo del Estudiante se hace en base a una red semántica y con un diseño orientado a objetos.

# Modelado del Estudiante

## 3.7 Diseño del ME basado en la técnica orientada a objetos.

Un objeto representa un concepto, a este concepto se le agregan dos elementos necesarios para que el Modelo del Estudiante interactue con la Interfaz y la Base de Conocimientos. Al objeto se le agregan dos listas, una que contiene todas las lecciones que de acuerdo a la base de conocimientos están relacionadas con este y otra que contiene todas las preguntas (indicadores) que de acuerdo a la Interfaz están relacionadas con este concepto [22]. La figura 26 muestra un objeto con las características correspondientes a las lecciones y

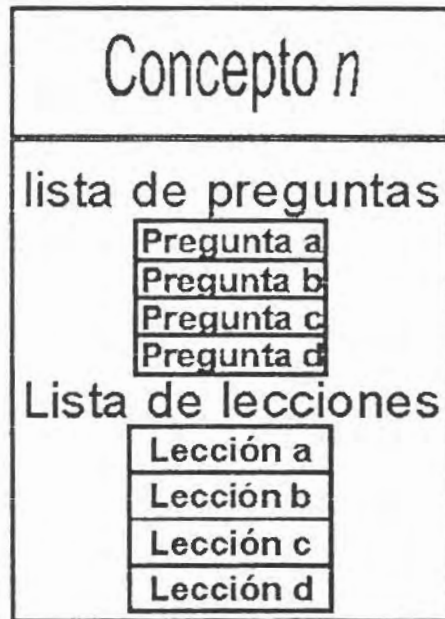


figura 26

Uso de un objeto para representar un nodo de la red semántica

preguntas.

La lista de lecciones y de preguntas, contienen los índices de estas. Como se recordara, la Base de conocimientos (lecciones) y la Interfaz (preguntas) están diseñadas en Multimedia. Luego el Modelo del Estudiante, la Base de Conocimientos y la Interfaz comparten los índices de las preguntas y de las lecciones. El Módulo Tutor se encarga de enviar los índices correspondientes del ME a la base de conocimientos e Interfaz, es decir, administra las funciones de los demás módulos.

El objeto propuesto funciona de la siguiente manera. Tiene las dos listas correspondientes y ambas listas pueden ser de tamaños diferentes pero no vacías. Así cuando el Módulo Tutor decide diagnosticar un objetivo pedagógico, este objetivo es un concepto que posee la lista de preguntas, de aquí toma la primera y la envía al Modulo Tutor el cual a su vez lo envía a la Interfaz y al estudiante se le presenta la pregunta adecuada. El Modulo Tutor puede decidir enseñar este concepto, por lo cual pide al objetivo pedagógico la lección correspondiente, el objeto toma de su lista la primera lección y la envía al Módulo Tutor este a su vez toma el índice y lo envía a la Base de conocimientos y se presenta la lección correspondiente. El objeto contiene dos funciones encapsuladas que retornan los índices correspondientes. Se sabe que las listas tienen diferentes tamaños (o inclusive pueden ser iguales). Las funciones van tomando los índices de acuerdo al orden que están en las listas, el orden de las lecciones se escoge de lecciones que son sencillas

# Modelado del Estudiante

hacia las más complejas, así mismo sucede con las preguntas<sup>3</sup>. Si en determinado momento en el proceso de enseñanza el objeto agota una de las listas entonces toma un índice al azar ya sea de las preguntas o de las lecciones.

Una de las condiciones por la cual el Módulo Tutor decide el cambio de concepto en la enseñanza es que las listas estén terminadas. Así otra función que debe realizar el concepto es informar al Módulo Tutor que el concepto está agotado<sup>4</sup>. De este modo el Módulo Tutor pone el valor del concepto a -2 y vuelve a buscar un objetivo pedagógico de acuerdo a las estrategias que ya tienen diseñadas. Al momento de que el objeto informa de este hecho al Módulo Tutor, reinicia las dos listas, así al momento de que este concepto sea tomado nuevamente como objetivo pedagógico, las listas estén disponibles nuevamente para su uso. Se sabe que el Módulo Tutor puede decidir volver a tomar este concepto como objetivo pedagógico, aún así el reiniciar las listas es válido.

El uso de las lecciones y preguntas en el diseño propuesto solo tienen uso en una sesión de enseñanza (los índices) por lo cual su estado no se guarda para su uso posterior en una nueva sesión con el mismo alumno.

De acuerdo con las funciones que debe realizar el Módulo Tutor y en base al Modelo del Estudiante, se diseña un objeto con funciones y valores encapsulados. El diseño satisface las características de una red semántica y con la técnica orientada a objetos. Así, de acuerdo a las funciones del Módulo Tutor y del Modelo del Estudiante se llega al siguiente diseño

Objeto: concepto	
Datos	
Nombre	Descripción
Valor	Posee los valores: -2 = AGOTADO -1 = NOCONOCE 0=NOHAYINFO 1=SEASUMECONOCIDO 2=CONOCIDO
Lista de indicadores	Es la lista de las preguntas que examinan o diagnostican este concepto
Lista de Lecciones	Es la lista de lecciones que sirven en la enseñanza de este concepto
Índice de indicadores	Este es un índice que indica la lección que puede utilizarse
Examinado	Una bandera que en el momento que se está haciendo el examen indica si un concepto fue ya examinado.
Padre	El siguiente concepto más complejo.
Indispensable	Lista de todos los conceptos que están ligados por la relación "es elemental para conocer a".

<sup>3</sup> Esta aseveración es válida en el algoritmo propuesto. Sin embargo algunos educadores podrían sugerir que lo inverso es válido también. La decisión depende del dominio del conocimiento y del pedagogo que colabore en el diseño.

<sup>4</sup> Denominamos concepto "agotado" cuando las dos listas han sido usadas, es decir todas las lecciones y preguntas para este concepto han sido presentadas al estudiante.

# Modelado del Estudiante

Similar	La liga hacia un concepto por la red que esta bajo la relación "es similar a".
Nombre	Descripción
VerificaEnseñanza	Comprueba que los conceptos que son indispensables para conocer este sean conocidos o se asuman conocidos
PropagarVerdadero	Propaga en valor verdadero (1) a partir del nodo examinado por medio de la relación "es indispensable para conocer a "
PropagarFalso	Propaga en valor Falso (-1) a partir del nodo examinado por medio de la relación "es indispensable para conocer a "
ModelaEstudiante	Esta función en el momento en que el objetivo pedagógico informa al Módulo Tutor que las lecciones y preguntas se acabaron es invocada por el mismo Módulo Tutor. De tal manera que de acuerdo con las estrategias de enseñanza modifica al ME para buscar otro objetivo pedagógico.
Retorna Estado	Esta función informa al Módulo Tutor acerca del uso de la lista de lecciones e indicadores, de tal manera que informa al Módulo Tutor cuando ambas han sido ocupadas en su totalidad.
Retorna Lección	Informa al Módulo Tutor de la lección que debe enseñarse
Retorna Indicador	Le informa al Módulo Tutor de la pregunta que debe presentarse al alumno.
GeneraObjetivo	Esta función es un método de inferencia. A partir de este concepto se busca en profundidad un concepto que pueda ser enseñado o diagnosticado. De este modo esta función informa al Módulo Tutor que la rama que le pertenece a este concepto ya es del dominio del estudiante o en caso contrario el concepto que debe tomarse como objetivo pedagógico (inclusive este mismo concepto)
Agotados	Esta función verifica si todos los conceptos similares a este concepto están agotados. Es decir, si su valor es 2. Este función la ocupa solamente la función GeneraObjetivo
QuitaAgotamiento	Esta función de acuerdo con las estrategias de enseñanza es usada por el Módulo Tutor cuando hace la búsqueda del Objetivo pedagógico. En el caso en que todos los conceptos similares estén agotados su valor los asigna a -1.

Además de estos valores y funciones, se requerirán en la implantación algunas otras, como por ejemplo funciones que armen la red.

# Modelado del Estudiante

Hasta esta parte se tienen definidos la Base de Conocimientos, la Interfaz y gran parte de la funciones del Módulo Tutor. Recuérdese el Problema de la representación del conocimiento en Multimedia<sup>5</sup>. En base a lo anterior solo el Modelo no se implanta mediante herramientas Multimedia y las funciones que el Módulo Tutor hace en base a este. Las funciones administrativas del Módulo Tutor se diseñan también en base a herramientas Multimedia ya que estas permiten la programación en base a flujos. Así en la figura 27 se muestra como los diferentes Módulos del STI quedan desarrollados con las herramientas software.

Multimedia	No Multimedia y Orientado a Objetos
Base de Conocimientos Lecciones Interfaz Interactores Módulo Tutor Funciones Administrativas	Modelo del Estudiante Red Semántica Módulo Tutor Todas las funciones que hacen en base al Modelo del Estudiante

figura 27

Implementación de los Módulos  
tomando en cuenta las herramientas a  
usar

## 3.8 Acceso a Bases de Datos

El STI debe guardar los datos del ME en una Base de Datos, de tal manera que en las diferentes sesiones en que el estudiante se presente frente al sistema, sea reconocido y no exista la necesidad de hacer un examen nuevamente, ni que se enseñen conceptos que ya son dominados por el alumno.

Así surge la necesidad de diseñar una base de datos, la conexión o explotación de esta base se hace ya sea mediante la herramienta Multimedia o la herramienta que se escoja para construir el Modelo del Estudiante. Las herramientas Multimedia son en general lenguajes de alto nivel, por lo cual el desarrollo de procesos que accedan a una base de datos es fácil. La herramienta que se escoja para la construcción del ME será por lo general de menor nivel que la herramienta Multimedia, por lo cual el desarrollo de procesos que accedan a una base de datos es un trabajo más largo. Se asume que se escoge la Herramienta Multimedia para acceder a la Base de Datos.

<sup>5</sup> Según reporte de IEEE y en base a que aún cuando exista esta herramienta las posibilidades de adquisición son bajas dado su costo



# Modelado del Estudiante

Se tienen los siguientes elementos que indican como debe ser diseñada la Base de Datos

- Existen en el ME  $m$  conceptos con 5 posibles valores.
- Las listas de lecciones e indicadores solo tienen existencia temporal en una sesión por lo cual no se guardan en la base de datos.
- Existen varios usuarios (estudiantes) que pueden acceder al STI y deben ser diferenciados.
- Los usuarios para ser identificados por el STI deben poseer un nombre y una clave de acceso.
- Se debe conocer el numero de sesiones que el usuario (alumno) ha ingresado al sistema, con el fin de no examinarlo más de una vez.
- Se debe conocer el nombre completo del alumno.

Las bases de datos relacionales permiten conservar esta información. En la figura 28 se muestra un diagrama entidad - relación para el STI.

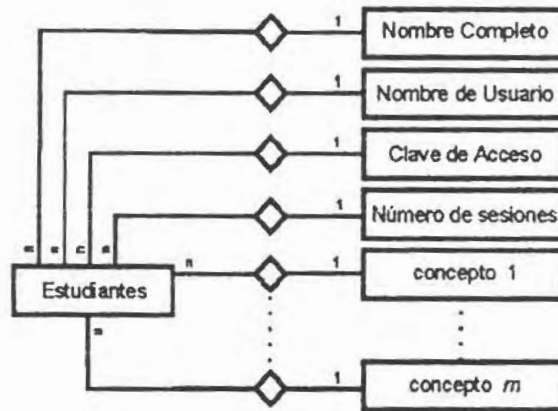


figura 28

Diagrama Entidad - Relación para la Base de Datos

Así el Módulo Tutor, dentro de sus funciones administrativas tiene que actualizar el Modelo del Estudiante con la información de la Base de Datos y al momento en que el alumno termine una sesión debe guardar el Modelo del Estudiante que representa al alumno.

La creación de la Base de Datos la debe hacer un administrador del STI. Debe insertar los registros que sean necesarios. Así mismo debe de asignar las claves de acceso y los nombres de usuario. El volumen de la BD depende de los objetivos que plantee el curso y de las exigencias de donde se implante el sistema.

Todos los campos que corresponden a los valores de los conceptos deben ser iniciados a 0 cada vez que se inserte un registro nuevo en la BD: Los demás datos se inician con las características particulares del estudiante.

Con la Inclusión de la BD las funciones del Módulo Tutor se incrementan. El diseño de objeto concepto solo necesita dos nuevas funciones. Una función que colabore en el traspaso de información de la BD al ME y otra función que haga el proceso inverso. Las funciones son la siguientes :

# Modelado del Estudiante

Función	Descripción
RetornaValor	Retorna el valor de un concepto en particular. Una vez obtenido el valor se pasa al Módulo Tutor el cual lo pone en la BD.
PonValor	El Módulo Tutor obtiene el valor de un concepto desde la BD y por medio de esta función pasa este valor al concepto correspondiente en el ME.

Tabla 3

Anexión de dos funciones más al objeto concepto.

## 3.9 Contexto del Modelo del Estudiante

De este modo quedan definidas las funciones que realiza el Módulo Tutor en Base al ME. También se define donde y con que herramientas se implementa. El diseño del ME en base a

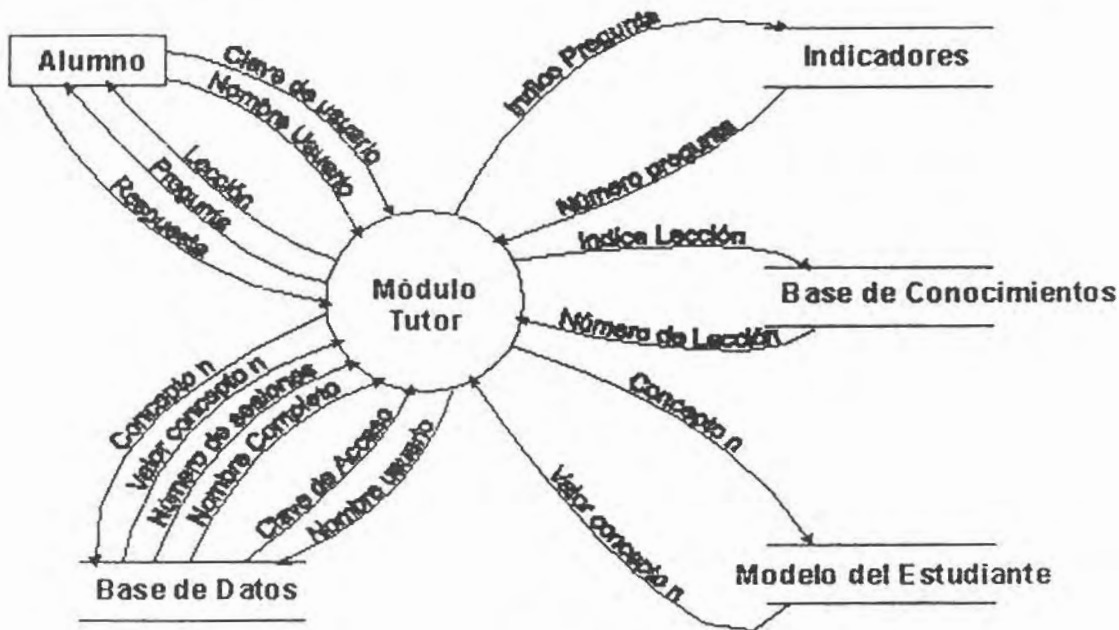


figura 29

Diagrama de Contexto del STI

POO y de las funciones que el Módulo Tutor explota del ME. La figura 29 muestra un diagrama de contexto que representa al STI propuesto en base al ME y el Módulo Tutor.

Este diagrama muestra la naturaleza de los diferentes Módulos que conforman un STI. Así se tiene que el Modelo del Estudiante, La Base de Conocimientos y la Interfaz son una organización de datos.

## 3.10 Diagrama de Flujo del Modulo Tutor en su parte administrativa

Se presenta ahora un diagrama de flujo que representa al Módulo Tutor diseñado para una Herramienta Multimedia. La figura 30 y la figura 31 muestra el diagrama de flujo correspondiente. Nótese que muchas de las funciones que fueron encapsuladas al diseñar el

# Modelado del Estudiante

objeto para el ME se usan. Este diagrama incluye también la parte correspondiente al uso de la Base de Datos. En este diagrama quedan definidas las funciones administrativas del Módulo Tutor y que están diseñadas para una herramienta Multimedia. La Interfaz está representada en este diagrama en el proceso "Presentación de la pregunta", el cual se repite dos veces; una cuando se examina y otra cuando se diagnóstica. La Base de Conocimientos está representada en el proceso "Presenta lección"

Así, los componentes de un STI quedan diseñados para su implementación con dos herramientas; una para Multimedia y otra que permita el desarrollo orientado a objetos. Se hace uso de estrategias de enseñanza, las cuales generan acciones en el Módulo Tutor el cual mediante su administración toma decisiones, todo con ayuda del Modelo del Estudiante.

En este capítulo se presentó el diseño completo del STI. Cabe señalar que el énfasis de esta tesis está en la creación del Modelo del Estudiante, Estrategias de enseñanza y el Módulo Tutor.

Hasta aquí el diseño es completo. El trabajo para finalizar un STI con este diseño es la implementación de un prototipo. La implementación requiere el uso de material pedagógico y del dominio del conocimiento.

# Modelado del Estudiante

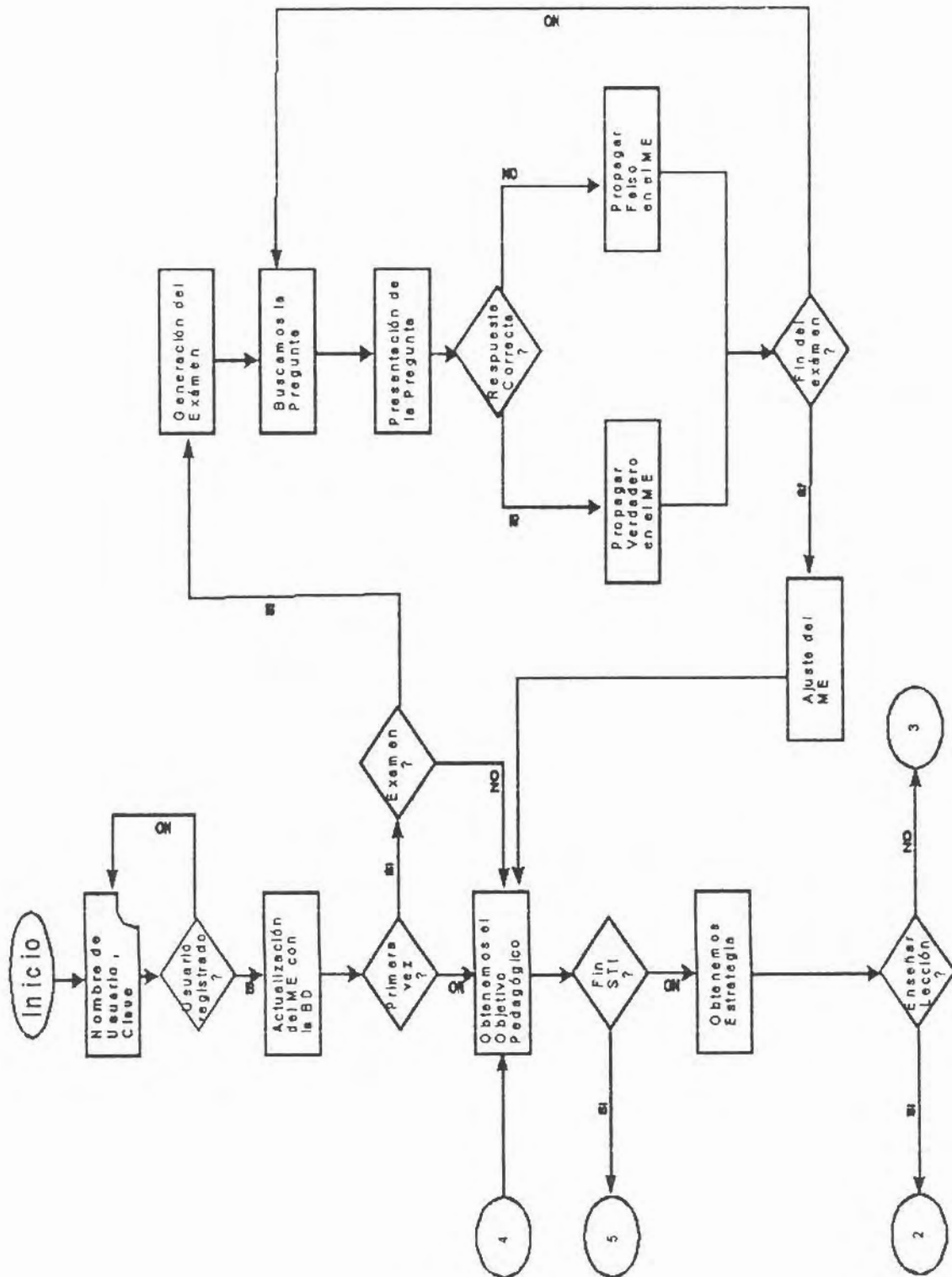


figura 30

Diagrama de flujo del Modulo Tutor (toma de decisiones)

# Modelado del Estudiante

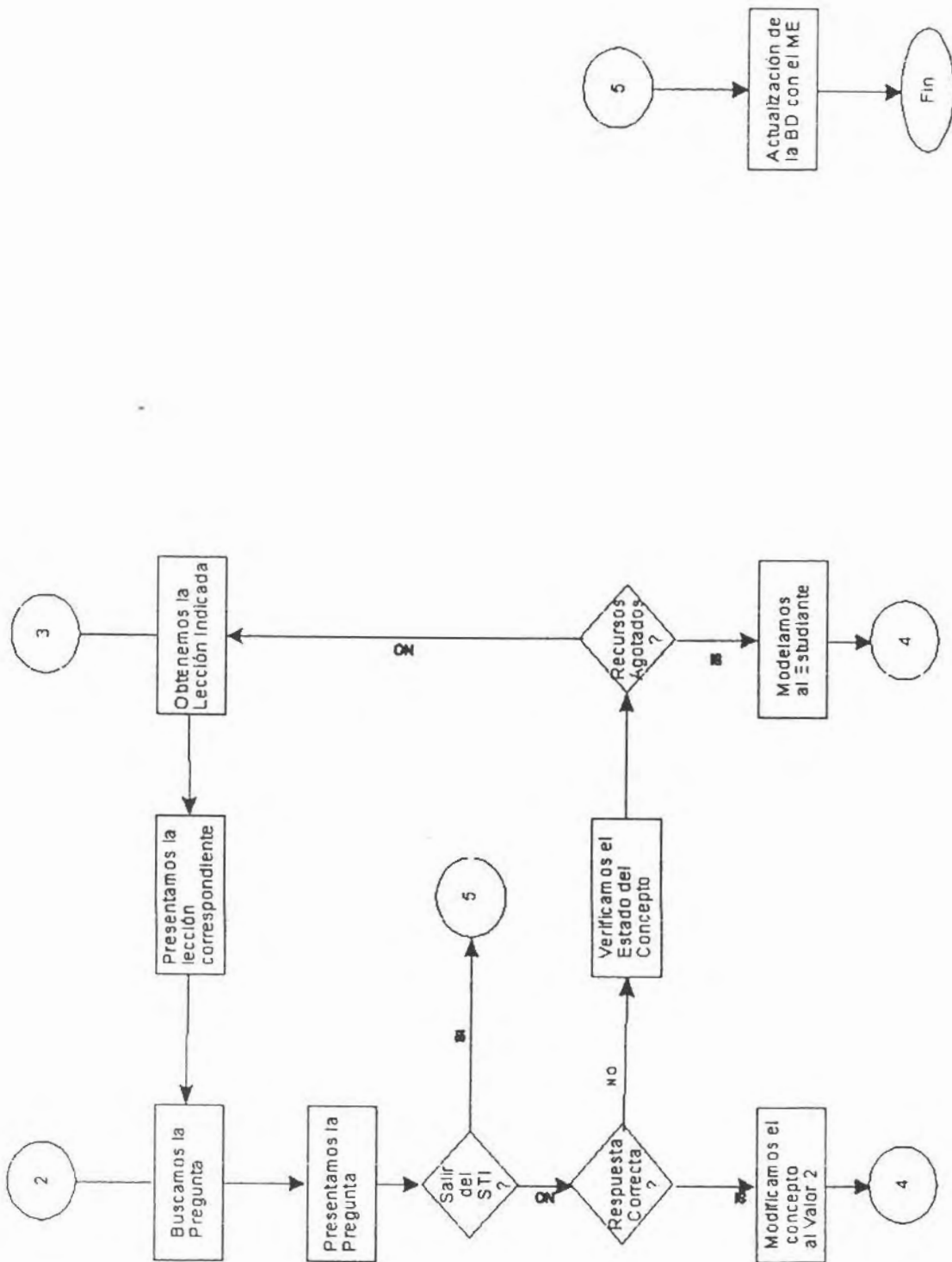


figura 31

Diagrama de flujo del Módulo Tutor (toma de decisiones) Continuación

# Capítulo cuatro

## Arquitectura

En base a herramientas computacionales y material disponible, se diseña la arquitectura de un STI. Esta incluye una plataforma Multimedia como base para su construcción.

Los componentes generalmente aceptados como parte de un Sistema Tutorial Inteligente y mencionado en párrafos anteriores, son los siguientes; Base de conocimientos, Interfaz de usuario, Modelo del Estudiante y el Módulo Tutor. La figura 32, muestra la arquitectura tomada de [20].

La Arquitectura propuesta para un Sistema Tutorial Inteligente tomando en cuenta los proceso que deben realizar los módulos de un STI se muestra en la figura 33

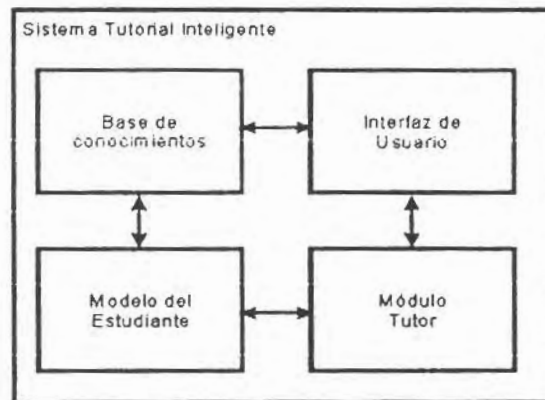


figura 32

Componentes de un STI generalmente aceptados

Así, se hace uso de la técnica de Superposición (Overlay) en donde el estudiante se compara con el experto en el dominio del área de conocimiento. El Modulo de dominio de aplicación y Modelo del Estudiante se combinan.

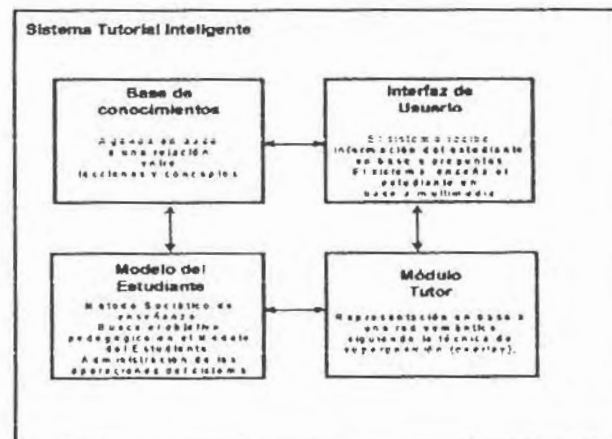


figura 33

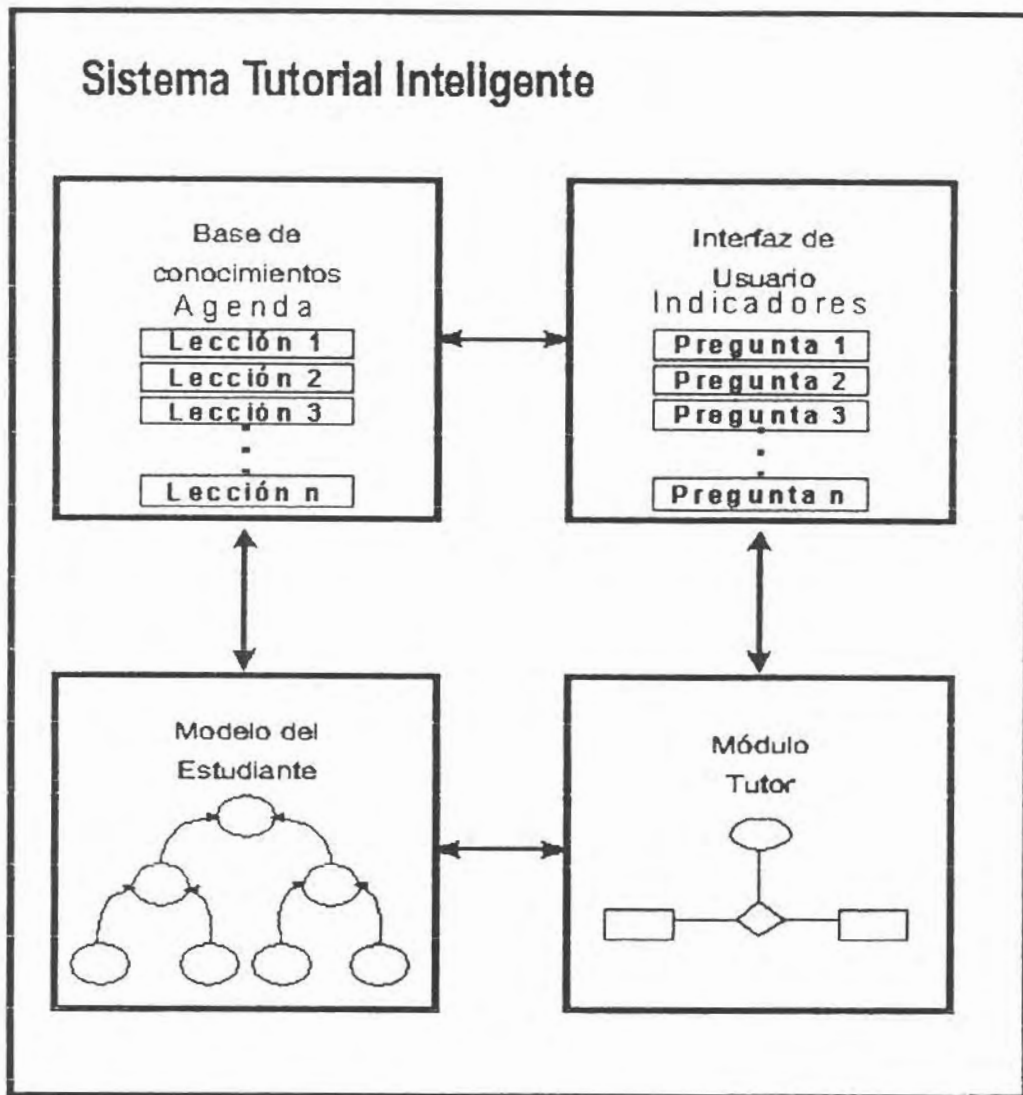


figura 34

Arquitectura de STI propuesto en base a técnicas de programación

De una arquitectura general para la construcción de un STI que se muestra en la figura 32. Una arquitectura particular, la cual es el objeto principal de investigación y la aportación más importante de este trabajo de tesis, se esquematiza en la figura 34.

## 4.1 Desglose de Conceptos del Dominio de Aplicación

El primer paso es desglosar el tema. Dentro del tema "El Sistema Eléctrico de Energía" se pueden distinguir una gran variedad de conceptos que pueden enseñarse con el fin de que este tema en general sea dominado por un estudiante. De aquí se generan 25 conceptos que se denominan hipótesis, cada una de estas hipótesis conforman el Modelo del Estudiante y a su vez ayudaran a generar las lecciones correspondientes. Con el objeto de tener un ME que pueda ser generalizado, la profundidad de unos temas no es la misma en toda la red. Así mismo las lecciones que se generan son diferentes, en numero para cada concepto, a las demás con el fin de validar el los procedimientos del Módulo Tutor. Los siguientes son los conceptos que se tomaran en cuenta en este prototipo. CX es una referencia del concepto.

# Arquitectura

- C0: Relación Campo Magnético Flujo Magnético.
- C1: Ley de Ampere.
- C2: Ley de Faraday.
- C3: Ley de Lenz.
- C4: Generación de Voltaje.
- C5: Generador simple.
- C6: Central de Combustibles Fósiles .
- C7: Central Turbogás.
- C8: Central de Ciclo Combinado.
- C9: Central Nucleoeléctrica.
- C10: Central Geotérmica.
- C11: Central Hidroeléctrica.
- C12: Centrales Generadoras.
- C13: Sistema de Generación.
- C14: Sistema Eléctrico de Potencia.
- C15: Subestación de Potencia .
- C16: Subestación Elevadora.
- C17: Subestación de Distribución.
- C18: Sistema de Transformación.
- C19: Nivel de Subtransmisión.
- C20: Nivel de Transmisión.
- C21: Nivel de Distribución.
- C22: Niveles de Tensión.
- C23: Sistema de Transmisión.
- C24: Sistema de Transporte.

Cabe mencionar que de los conceptos listados el concepto más general o más complejo es el numero 14.

## *4.2 Conformación del Modelo del estudiante*

Una vez desglosados los conceptos que conforman el dominio de aplicación se procede a armar una red semántica con las relaciones propuestas en el diseño. Así en la figura 35 se representa dicha red. Esta es la red semántica que utiliza el prototipo y representa básicamente dos ideas:

- Ya que se usa la técnica de Superposición, la red es en sí el conocimiento del experto que se pretende enseñar y a la vez
- Representa el dominio que el alumno tiene del tema ya que cada nodo (con los valores asociados ) representan un concepto.



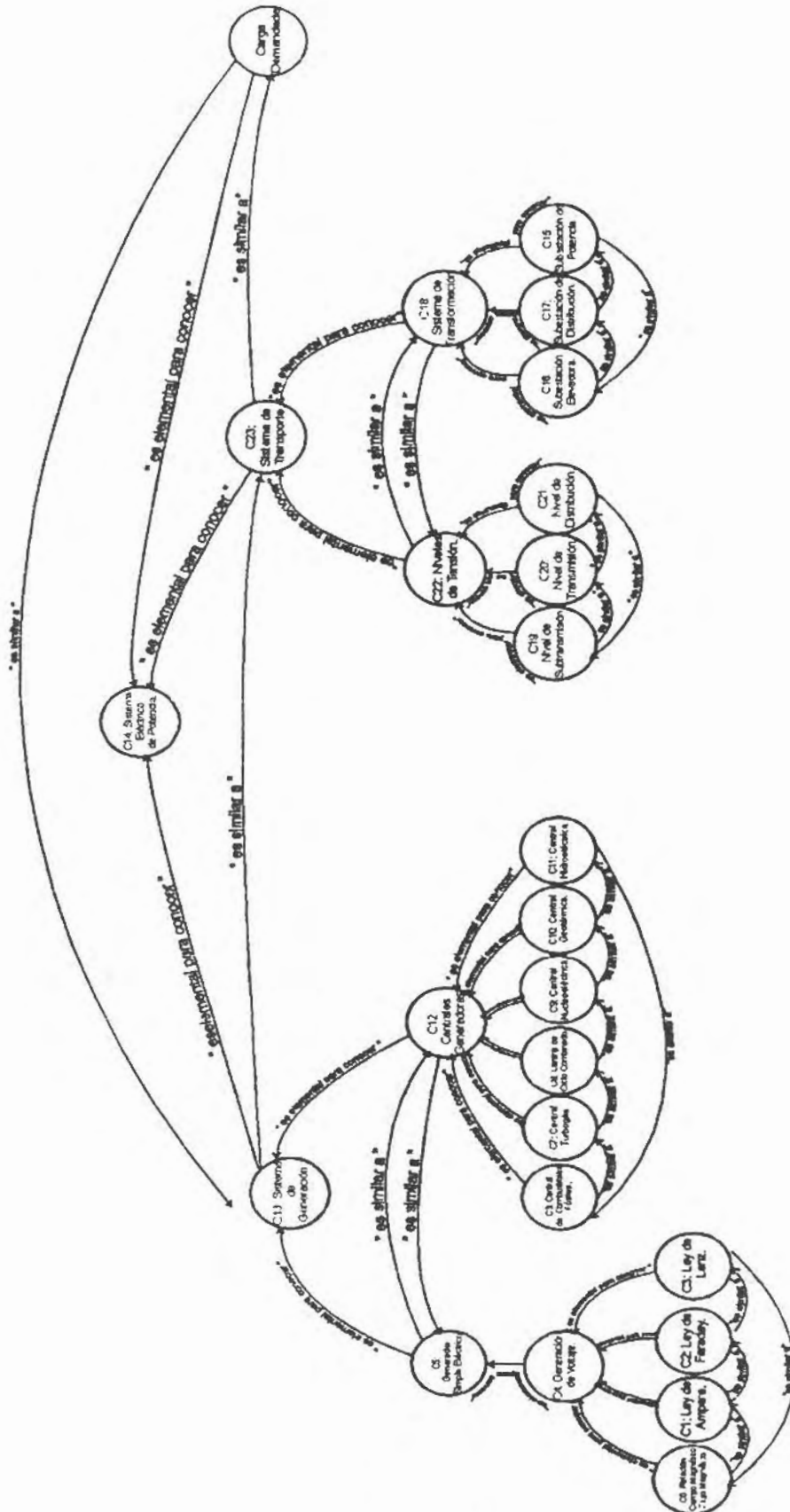


figura 35

Red Semántica que representa el Modelo del Estudiante y la Base de Conocimientos

## 4.3 Diseño de la Agenda

El dominio de aplicación que se tomará para implantar el prototipo es : "El Sistema Eléctrico de Energía". Las estrategias de enseñanza ya están definidas.

La construcción de la base de conocimientos involucra la decisión de haber seleccionado la combinación de conocimiento plano - jerárquico [31]. Y por otro lado la selección de un modelo de estudiante en base a una red semántica. Según [22] una red semántica es la forma más adecuada para modelar al alumno cuando se enseñan conceptos. Además de una estrategia de organización del conocimiento basada en Agenda [22]

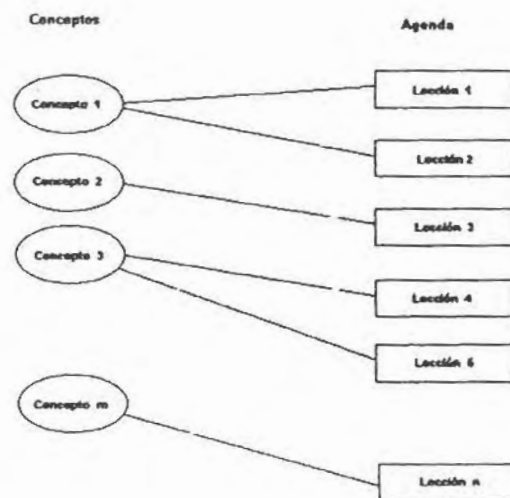


figura 36

Relación entre conceptos y lecciones

Así en la arquitectura propuesta se construyen lecciones con objetivos pedagógicos específicos. Tales objetivos representan un concepto del Modelo del Estudiante. De ahí se forma una relación entre conceptos y lecciones. Con el fin de facilitar el aprendizaje se construyen varias lecciones para un mismo concepto. En la figura 36 se puede observar dicha relación.

La mayoría de las herramientas Multimedia permiten la creación de lecciones. Uno de los primeros problemas a resolver es como organizar  $n$  número de lecciones al mismo tiempo de tal modo que se puedan tener acceso a ellas y que el control lo tome el Módulo Tutor después de presentar una de las lecciones. Se tiene una relación número de concepto - índice de lección. Se puede hacer que los otros Módulos tengan una referencia de las lecciones (el ME). De este modo cuando el Módulo Tutor reciba un índice de una lección sabrá que lección tiene que presentar.

# Arquitectura

Las lecciones se organizan en diferentes caminos y se acceden a ellas mediante una condición, así cualquier lección puede ser presentada en cualquier momento (cuando lo decida el Módulo Tutor). La figura 37 muestra un diagrama de flujo correspondiente.

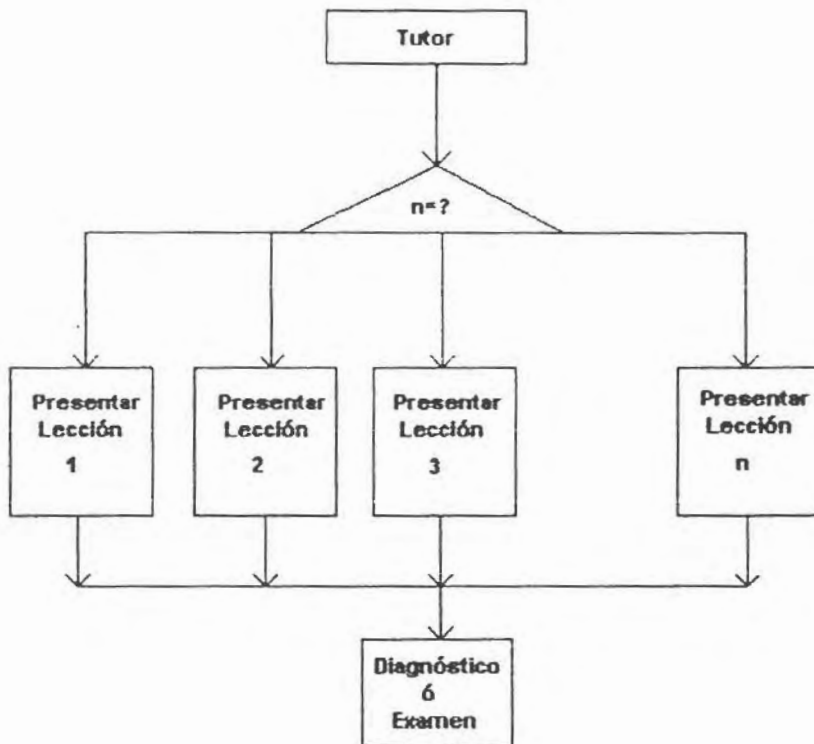


figura 37

Acceso del Módulo Tutor a las lecciones

Esto da lugar a una lista que relaciona el número de lección con su concepto correspondiente

lección 1 ... concepto 4  
lección 2 ... concepto 3  
lección n... concepto m

# Arquitectura

La creación del número de lecciones que corresponde a un concepto se decide en colaboración con el experto en el dominio del conocimiento. Algunas veces pueden existir conceptos que con pocas lecciones (incluso una) pueda lograrse su comprensión, en algunas otras ocasiones será necesario tener varias lecciones para lograr la comprensión de un concepto en particular. Esto depende de la granularidad y flexibilidad de enseñanza que se desee del tema.

## 4.4 Diseño Interfaz

La Interfaz se diseña con el objetivo de obtener información del estudiante, proporcionar información y retener su atención durante el proceso de enseñanza.

El diseño de la Interfaz además se fundamenta en la estrategia de enseñanza en base al método socrático [25]. Es decir la enseñanza se lleva a cabo mediante la formulación de preguntas.

El Módulo Tutor y la base de conocimientos presentan por medio de la Interfaz (desde un punto de vista técnico) la información al usuario y es a través de este que se establece la comunicación Estudiante y Sistema.

En la arquitectura propuesta, la Interfaz es un conjunto de indicadores (preguntas) diseñadas con el fin de diagnosticar y examinar al estudiante. Cada pregunta esta relacionada con un concepto. Se diseñan varias preguntas con el fin de tener una mejor flexibilidad al interactuar con el estudiante. La figura 38 muestra como se relacionan las preguntas a los conceptos.

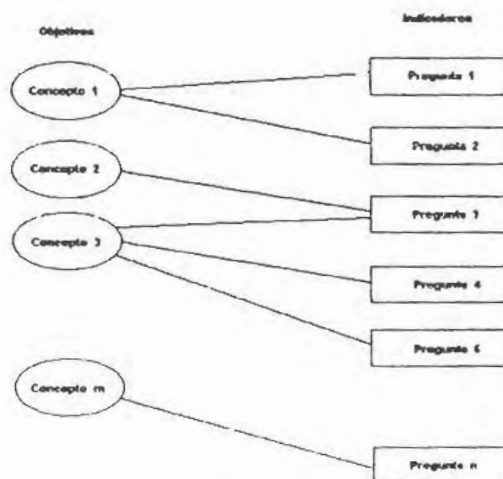


figura 38

Relación entre conceptos y preguntas

Ahora bien, la Interfaz se presenta mediante una herramienta Multimedia. La única diferencia que existe con la base de conocimientos es que una pregunta puede estar diseñada para diagnosticar (examinar) más de un concepto. La solución es la misma que en el caso de la

Base de Conocimientos. La figura 39 muestra un diagrama en donde se representa el flujo para acceder a los indicadores, mediante un diseño con una herramienta Multimedia.

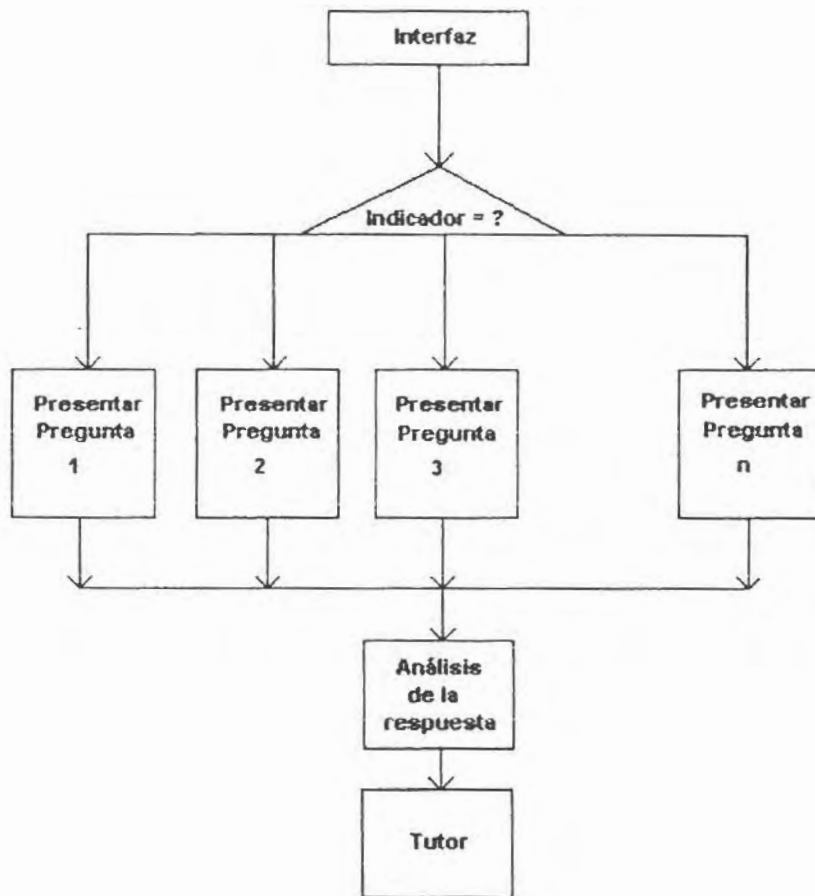


figura 39

Acceso del Módulo Tutor a las preguntas

Así como en la Base de Conocimientos, en la Interfaz se crea una lista que relaciona las preguntas con los conceptos, por ejemplo.

*pregunta 1... concepto 3*  
*pregunta 2 .. concepto 5*  
*pregunta n ... concepto m*

Así mismo al igual que en la Base de Conocimientos, las preguntas se diseñan junto con el experto en el tema y el pedagogo. Cada pregunta esta relacionada con un concepto, por lo cual su objetivo es el diagnostico (o en su caso examen) de este.

# Arquitectura

## 4.5 Generación de Lecciones

Se generan también lecciones que contribuyen a la enseñanza del tema y en especial cada lección se genera con el fin de mejorar el conocimiento del alumno en determinado concepto. Estos conceptos son buscados por el Módulo Tutor para generar el objetivo pedagógico.

A continuación se presenta la lista de lecciones creadas para el prototipo:

Numero de Lección	Objetivo pedagógico = Concepto = Elemento del Modelo del Estudiante
1	Sistema Eléctrico de Potencia, Carga Demandada.
2	Sistema Eléctrico de Potencia, Sistema de Generación, Sistema de Transporte.
3	Sistema Eléctrico de Potencia, Sistema de Generación.
4	Sistema de Generación
5	Central de Combustibles Fósiles
6	Central de Combustibles Fósiles
7	Central de Combustibles Fósiles
8	Central Turbogás
9	Central Turbogás
10	Central Turbogás
11	Central Ciclo combinado
12	Central Ciclo combinado
13	Central Ciclo combinado
14	Central Nucleoeléctrica
15	Central Nucleoeléctrica
16	Central Nucleoeléctrica
17	Central Geotérmica
18	Central Geotérmica
19	Central Geotérmica
20	Central Hidroeléctrica
21	Central Hidroeléctrica
22	Central Hidroeléctrica
23	Sistema de Transporte, Sistema de Transmisión, Sistema de Transformación
24	Niveles de Tensión
25	Sistema de Transformación.
26	Carga Demandada
27	Sistema Interconectado Nacional
28	Relación Campo Magnético Flujo Magnético
29	Ley de Ampere
30	Ley de Ampere
31	Ley de Faraday
32	Ley de Faraday
33	Ley de Faraday
34	Ley de Lenz
35	Generación de Voltaje
36	Generador Eléctrico Simple
37	Nivel Transmisión
38	Nivel de Subtransmisión
39	Nivel de Distribución
40	Subestación Elevadora
41	Subestación de Potencia
42	Subestación de Distribución

**Tabla 4**

Relación Lecciones - Conceptos

# Arquitectura

Esta lista es de lecciones que son programadas mediante Authorware. El índice se controla mediante un icono de Iteración que Authorware define para poder acceder a una línea de flujo.

## 4.6 Generación de Indicadores

El siguiente paso es la generación de indicadores para diagnosticar (examinar) el nivel cognoscitivo del estudiante. Cada pregunta se diseña para un concepto en particular. Así, los siguientes indicadores son los que se usan en el prototipo:

- 1- Cual es la función de un SEE ?
  - **Generar Energía Eléctrica Con Voltajes y Frecuencias Constantes**
  - Generar Vapor de Agua para las Turbinas
  - Elevar el Voltaje entre las Estaciones Transmisoras
  
2. Principal subdivisión de los componentes de un SEE ?
  - Sistemas de Elevación de Voltaje y Sistemas de Reducción de Voltaje
  - **Sistemas de Generación y Sistemas de Transporte (Transmisión)**
  - Sistema de Generación de Energía únicamente
  
3. Dispositivos Principales que componen un SEE
  - Generador Síncrono, Línea de Subtransmisión, Carga demandada
  - **Generador Síncrono, Transformador, Línea de Transmisión, carga Demandada.**
  - Sistema de Generación, Sistema de Transporte,
  
- 4.- Siempre que hay un flujo de corriente en un conductor (Cristian Oersted Andersen)
  - Existe un flujo magnético en dirección contraria al flujo eléctrico.
  - **Existe un Campo Magnético en torno al conductor.**
  - Existe un Flujo Magnético en torno al conductor.
  
- 5.- Cuando una corriente circula por un conductor, a su alrededor de producirá un campo magnético concéntrico a la trayectoria de la misma, cuya dirección depende del sentido de la corriente eléctrica, esta el la ley de ?
  - **Ampere**
  - Faraday.
  - Lenz
  
- 6.- El Según la Ley de Ampere campo el Magnético esta relacionado linealmente con la corriente que lo produce y
  - **El número de espiras que lo produce y/o el numero de espiras que lo produce**
  - El numero de bobinas que contiene
  - El campo Magnético que circula por el conductor
  
- 7.- La regla de la mano derecha para las bobinas es una consecuencia de la ley de
  - Faraday
  - Lenz
  - **Ampere**
  
- 8.- Indica que siempre que se tenga un Campo Magnético Variante en el tiempo, se generará un cierto campo eléctrico en el espacio.
  - Ley Faraday
  - Ley de Ampere
  - Ley de Lenz

9.-  $e = - d\phi_b / dt$  es la formula de la ley de ?

- Faraday
- Lenz
- Ampere

10.- La corriente inducida circulara en un sentido tal que se oponga al cambio que la produce

- Ley de Ampere
- Ley de Faraday
- Ley de Lenz

11.- El principio de la conservación de energía se sugiere por un signo negativo en la ley de:

- Lenz
- Faraday
- Ampere

12.- La generación de voltaje involucra los principios de

- Ley de Coulomb, Ley de Ampere.
- Ley de Faraday, Ley de Lenz, Ley de Ampere
- Ley de Maxwell, Ley de la conservación de la energía

13.- En un Generador Eléctrico Simple se utilizan :

- Estator, Anillos Rozantes, Rotor, fuente variable de CD
- Estator , bobinas, rotor
- Fuente de Calor, Generador de Vapor, Generador Eléctrico

14.- Su objetivo es el de transmitir energía eléctrica desde los centro de generación hasta los consumidores

- Sistema de Transporte
- Sistema de Transmisión
- Sistema de Transformación

15.- División del Sistema de Transporte

- Sistema de Transmisión y Sistema de Transformación
- Sistema de Generación, Sistema de Transporte, Sistema de Transformación.
- Sistema de generación, carga Demandada, Sistema de Transmisión

14.- Los niveles de tensión se dividen en:

- Nivel de Subtransmisión, Nivel de Distribución, Nivel de transmisión
- Nivel de Carga, Nivel de Tensión, Nivel de Distribución.
- Nivel de Energía, Nivel de Distribución, Nivel de Transmisión.

15.- A este nivel el objetivo es el movimiento de grandes bloques de energía de los centros de generación a las subestaciones importantes ubicadas en los principales centro de consumo,

- Nivel de Transmisión
- Nivel de Consumo
- Nivel de Subtransmisión

16.- A este nivel los movimiento de energía se realizan a grandes distancias, por lo cual los niveles de tensión son altos

- Nivel de Transmisión.



# Arquitectura

- Nivel de Consumo
- Nivel de Subtransmisión

17.- Movimiento de Cantidades regulares de bloques de energía eléctrica de las subestaciones de extra y alta tensión a las subestaciones de distribución.

- Nivel de Transmisión.
- Nivel de Consumo
- **Nivel de Subtransmisión**

18.- Su objetivo es el transporte de energía de las subestaciones a los consumidores

- Nivel de Transmisión
- **Nivel de Distribución**
- Nivel de Transporte

19.- Este sistema se encuentra constituido por subestaciones, en las subestaciones se modifican los parámetros de tensión y corriente

- **Sistema de Transformación**
- Sistema de Potencia
- Sistema de Distribución

20.- Subestación elevadora:

- **Eleva el voltaje que recibe**
- Eleva la corriente que recibe
- No existe

21.- Si en una subestación, la carga esta dirigida a otras subestaciones entonces se conoce como:

- Subestación de Potencia
- **Subestación de distribución**
- Subestación elevadora

22.- Si la carga de una subestación esta dirigida a los usuarios, entonces se conoce como

- **Subestación de distribución**
- Subestación elevadora
- Subestación de distribución

23.- Se denomina de este modo la energía eléctrica que recibe el usuario.

- **Carga Demandada**
- Energía de Tensión
- Voltaje Final

24.- Su objetivo principal es la generación de la energía eléctrica demandada.

- **Sistema de generación**
- Sistema de Potencia
- Sistema de Transmisión

25.- Dentro del proceso de transformación de energía, es el ultimo tipo de energía a que se llega antes de su transformación final a energía eléctrica.

- Energía Química
- **Energía Mecánica**
- Energía Potencial



26.- En esta central se usa como fuente primaria un combustible fósil como carbón, combustóleo o gas natural.

- **Central de Combustibles Fósiles**
- Central Nucleoeléctrica
- Central Hidroeléctrica

27.- En esta central se produce energía en base a un ciclo de una turbina de gas, usando como combustible, Diesel o gas natural.

- **Central Turbogás**
- Central de Combustibles Fósiles
- Central Nucleoeléctrica

28.- La Central de ciclo combinado esta formada por :

- **Una central Turbogás y Una Central Termoeléctrica**
- Central Nucleoeléctrica y Hidroeléctrica.
- Central de combustibles Fósiles y Central Nucleoeléctrica.

29.- En esta central se utiliza el vapor geotérmico extraído del subsuelo.

- **Central Geotérmica**
- Central Nucleoeléctrica
- Central de Combustibles Fósiles

30.- En esta central se utiliza como fuente primaria la energía cinética del agua.

- Central de ciclo combinado
- Central Turbogás
- **Central Hidroeléctrica**

31.- En esta central se utiliza como fuente primaria de energía Uranio enriquecido

- Central de Ciclo Combinado
- Central de Combustibles Fósiles
- **Central Nucleoeléctrica.**

Cada una de estas preguntas (indicadores) esta diseñada para examinar (diagnosticar) un concepto en particular o varios al mismo tiempo. La siguiente lista muestra la relación que existe entre los conceptos y las preguntas.

Objetivo pedagógico = Concepto = Elemento del Modelo del Estudiante	Número de Pregunta
Concepto 0	Pregunta 4
Concepto 1	Pregunta 7
Concepto 1	Pregunta 6
Concepto 1	Pregunta 5
Concepto 2	Pregunta 9
Concepto 2	Pregunta 28
Concepto 3	Pregunta 10
Concepto 4	Pregunta 12
Concepto 5	Pregunta 13
Concepto 6	Pregunta 28
Concepto 7	Pregunta 29
Concepto 8	Pregunta 30
Concepto 9	Pregunta 33
Concepto 10	Pregunta 31
Concepto 11	Pregunta 32
Concepto 12	Pregunta 27
Concepto 13	Pregunta 26
Concepto 14	Pregunta 3
Concepto 14	Pregunta 2
Concepto 14	Pregunta 1
Concepto 15	Pregunta 23
Concepto 16	Pregunta 22
Concepto 17	Pregunta 24
Concepto 18	Pregunta 21
Concepto 19	Pregunta 19
Concepto 20	Pregunta 17
Concepto 20	Pregunta 18
Concepto 21	Pregunta 20
Concepto 22	Pregunta 16
Concepto 23	Pregunta 14
Concepto 23	Pregunta 15
Concepto 24	Pregunta 25
Concepto 24	Pregunta 3

**Tabla 5**

Relación Conceptos - Indicadores

Todas las preguntas y lecciones son la interfaz que se tiene con el alumno. Estas preguntas están implementadas en Authorware. En las preguntas, la respuesta sólo se analiza como verdadera o falsa. En el texto de párrafos anteriores, la respuesta verdadera está en negritas. Y la pregunta se muestra como texto en el monitor y el alumno responde por medio de accionar un botón que tiene la respuesta correcta.

## 4.7 Control del flujo en la Herramienta Multimedia

Se tienen dos herramientas a usar, Authorware 3.5 y Visual C++. Como se había planeado en secciones anteriores, en Authorware se desarrollan las lecciones con las herramientas disponibles en la Unidad de Resultados de Simulación del IIE, así también se desarrollan todos los indicadores. Así, en Authorware se tiene un icono de decisión (ver figura 40). Los iconos de decisión proporcionan la capacidad de continuar el flujo mediante diferentes tipos de eventos, a cada evento se asocia una ruta diferente. Tales eventos son: acciones sobre un botón, condicionales, etc. Las lecciones o indicadores poseen un índice, así mediante este icono se puede continuar el flujo hacia una lección o indicador mediante la condición de que una variable posea el valor de un índice. Esta variable tiene en determinado momento (cuando el modulo Tutor lo decida) el valor de un índice (indicador o lección).

# Arquitectura

Cada vez que se presenta una lección al estudiante el flujo debe ir al Módulo Tutor en la parte que diagnóstica. De tal modo que se presenta una lección que termina, el flujo continua hacia el diagnóstico que posee el Módulo Tutor.

La figura 41 muestra el dialogo para el tipo de respuesta que se quiere monitorear. De estas opciones se toma la de tipo condicional.

La figura 42 muestra el diálogo para la respuesta condicional, se considera que el flujo continua por aquí cuando la variable *variable* es igual a 1.

La figura 43 muestra cada uno de los flujos que pueden tomarse condicionalmente mediante la variable *variable*. El icono de decisión continua el flujo por las diferentes opciones de acuerdo con el valor de la variable *variable*



Interaction Options/Text Entry Options dialog box

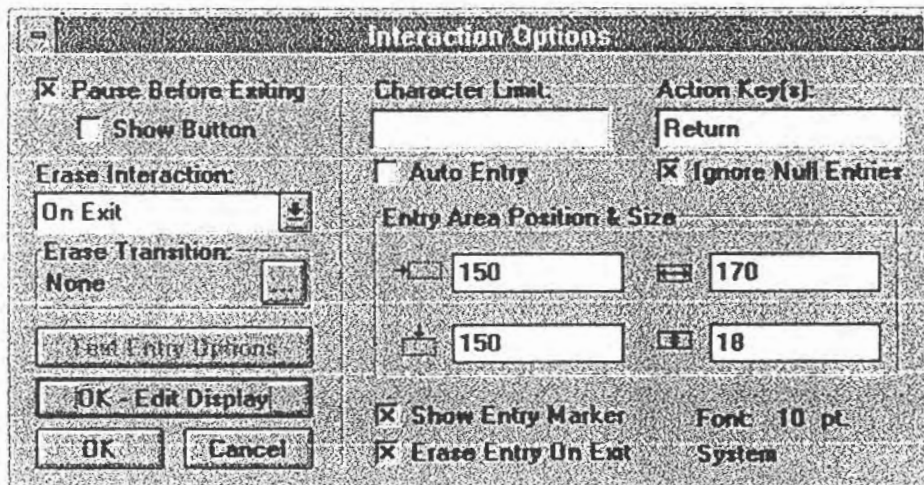


figura 40

Icono de decisión y sus opciones de interacción para llevar a cabo la codificación de la interfaz y base de Conocimientos

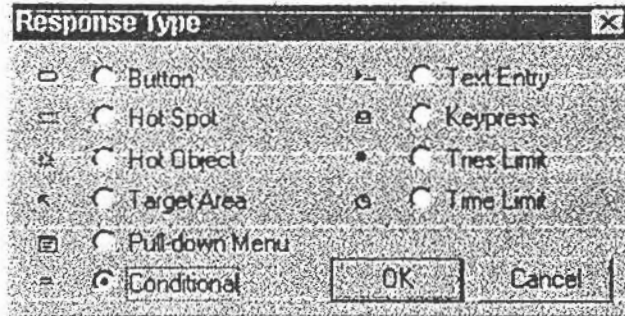


figura 41

Tipo de respuesta para un icono de decisión

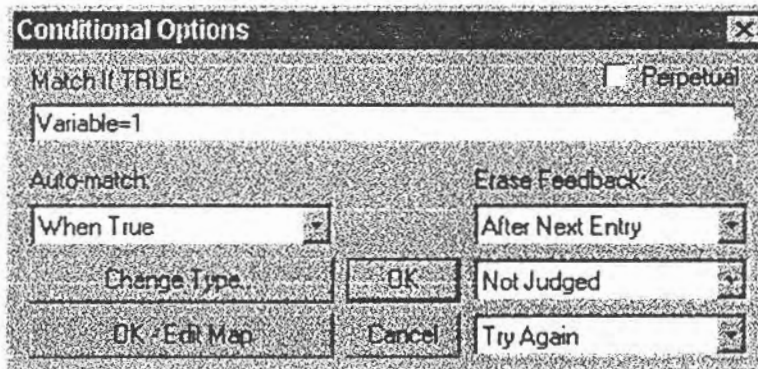


figura 42

Opciones para un tipo de respuesta condicional en un icono de decisión

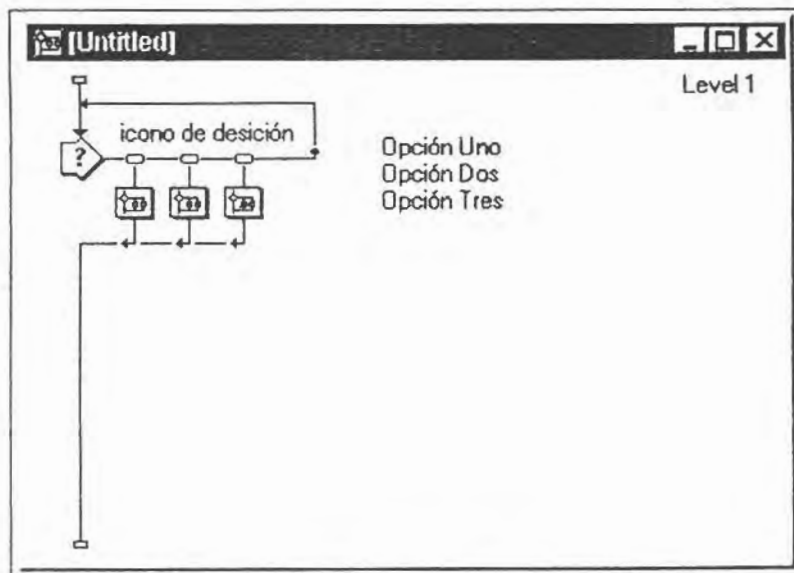


figura 43

Ejemplo del uso de un icono de decisión

## 4.8 Codificación Agenda

De este modo en la figura 44 se observa como se accede a las lecciones. Aquí, en la caja con el título "Invocamos a Retorna Lección" se invoca al Módulo Tutor para que decida en base al Modelo del Estudiante que lección es la que se va a enseñar

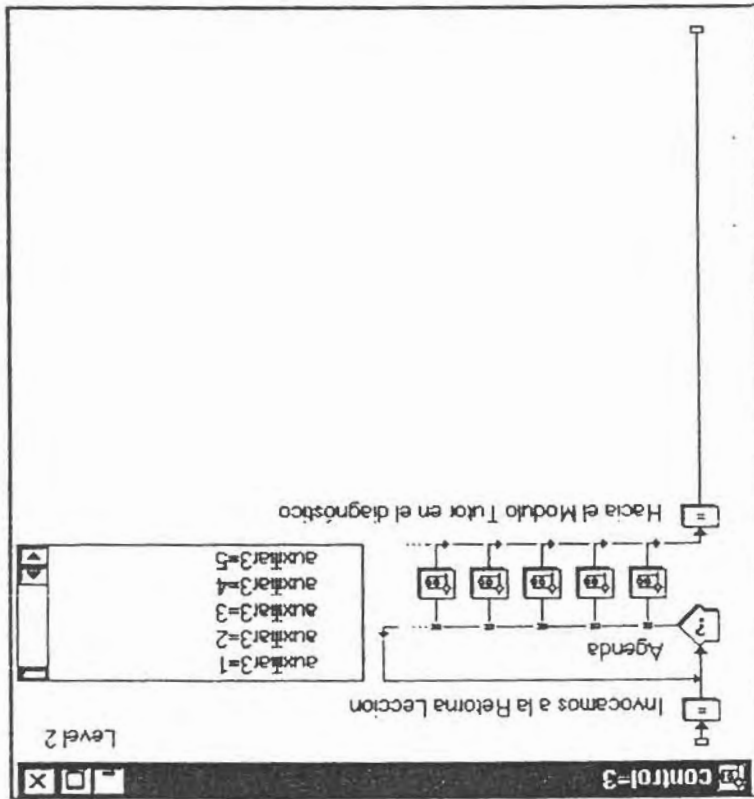


figura 44

Icono de decisión *Agenda* que controla el acceso a las lecciones

Después el flujo continua hacia el Módulo Tutor para que diagnostique. La implementación del icono con el título "Invocamos a RetornaLeccion" contiene una invocación a una función que será explicada más adelante en este capítulo.

Por otra parte el proceso que controla a los indicadores (preguntas) no esta completamente aislado, ya que el proceso que presenta una pregunta puede ser invocado sea examen o sea en función de un diagnóstico. Una vez que el alumno responde, el flujo lo toma el Módulo Tutor , y dependiendo de la respuesta, de las condiciones del examen y del objetivo pedagógico se deciden diferentes acciones. Estas acciones pueden ser:

- Volver a preguntar.
- Poner una lección.
- Buscar otro objetivo ó
- Modelar al Estudiante.

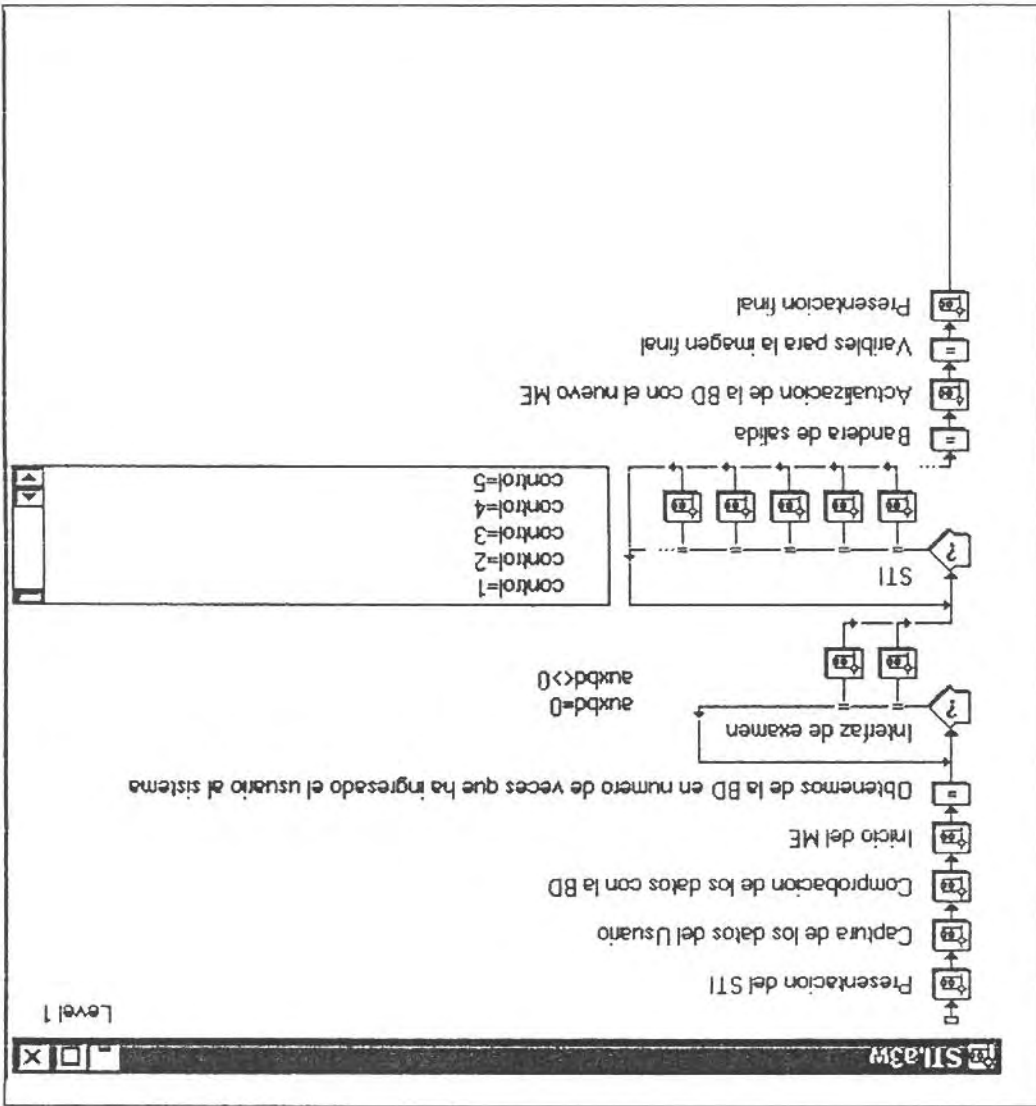


Figura 45

Módulo Tutor (Toma de decisiones)

El icono STI controla el acceso a indicadores y procedimientos del sistema

# Arquitectura

En la figura 45 se aprecian otras funciones del Tutor que serán explicadas posteriormente en este capítulo.

De este modo el icono de decisión que controla a los indicadores, se combina con flujos del Módulo Tutor. Así, como se observa en la programación del Módulo Tutor esta implícita dentro del icono de decisión de los indicadores. Al mismo tiempo dentro de los flujos que posee este icono se incluye el proceso de Agenda de tal modo que las acciones del Módulo Tutor puedan acceder las lecciones en el momento que sea necesario.

De este modo la interfaz se conforma por pantallas y preguntas. El modo de codificar una pregunta se muestra en la figura 46. Aquí se hace uso de botones que contienen la respuestas posibles a una pregunta. En el apéndice D se puede observar la manera en que se presenta esta interfaz.

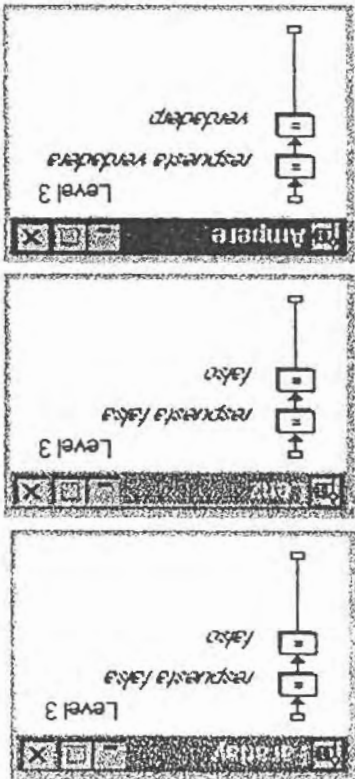
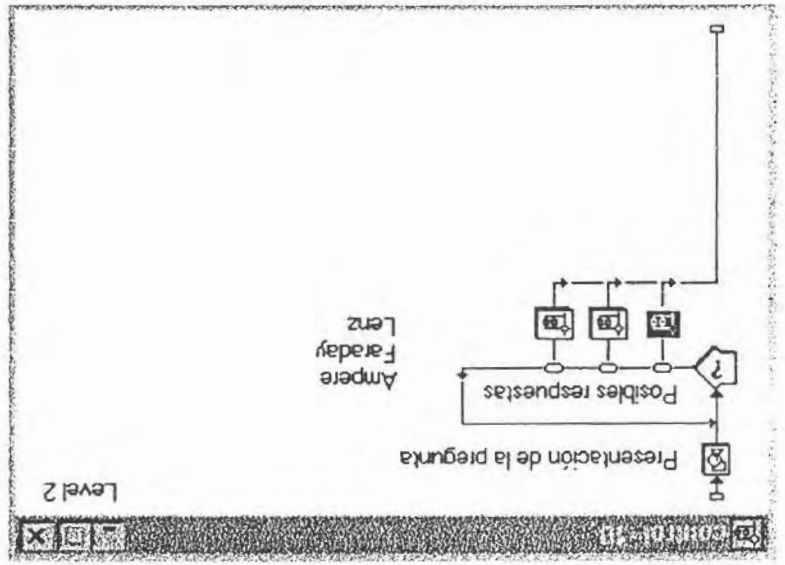


figura 46

Esta es la interfaz definida dentro del STI propuesto. Como sistema el STI necesita definir una comunicación con el usuario (alumno) en este caso se diseñan pantallas que justifican las acciones del STI e informan de las actividades que realiza. Los siguientes requerimientos de pantallas son necesarios para el STI:

- Presentación del STI con el fin de informar al usuario de: Que tipo de Sistema, tema a tratar, quien lo desarrolla.
- Interfaz para obtener el nombre de usuario y clave de acceso



# Arquitectura

- Interfaz para obtener la decisión del usuario acerca de la decisión de hacer un examen inicial.
- Interfaz que indique que el examen ha finalizado.
- Interfaz que indique al usuario que su respuesta es correcta y que cambiara de concepto
- Interfaz que indique al usuario que su respuesta es incorrecta y que intentara cambiar de concepto.
- Interfaz que informe al usuario que el concepto esta agotado y que intentara cambiar de concepto.
- Interfaz que informe al usuario que ha concluido su curso satisfactoriamente.

Estas pantallas son parte del Módulo Tutor y se implementan en modulo de Authorware. Estas pantallas son mostradas en el Apéndice B.

## 4.10 Codificación del Módulo Tutor y Modelo del Estudiante

Mediante Authorware se pueden importar funciones de Librerías de Ligado Dinámico (DLL por sus siglas en ingles). Mediante Visual C++ se genera un DLL que contenga al Modelo del Estudiante y a su vez las funciones que el Tutor hace sobre de él. Un DLL es básicamente un programa ejecutable que posee diferentes características, de este se toma la posibilidad que tiene para exportar funciones y la permanencia de un DLL como un proceso en determinado momento.

El DLL se crea con el lenguaje C++, aquí se genera una clase que posea los atributos definidos en el diseño.

//Clase que define un nodo el la red semántica, a la vez con las instancias generadas se //forma el Modelo del Estudiante

```
class Hipotesis
{
public:
//Constructores
Hipotesis(unsigned char,unsigned char, int, int);
Hipotesis(void){Hijos=0;Valor=0,Identificador=0;} //Constructor de default

char Valor;//-2=AGOTADO-1=NOCONOCE,0=NOHATINFO,1=SEASUMECONOCIDO,
2=CONOCIDO
char Hijos;//Cuantos conceptos son indispensables para conocer este concepto
unsigned char Identificador;//Numero que identifica a la hipótesis
unsigned char *ptrInd;//Apuntador a los indicadores (preguntas) de este objeto
unsigned char *ptrLecciones;//Apuntador a las lecciones de este objeto
int NumeroInd;//Numero de indicadores (preguntas) de este objeto
int NumeroLecciones;//Numero de lecciones de objeto
int IndiceInd;//Indice del ultimo indicador usado de este objeto
int IndiceLecciones;//Índice de la ultima lección usada de este objeto
int Checado;//auxiliar en el examen, indica cuando un concepto ya se examino
//apuntadores, son las ligas de la red semántica
//Hermano = "es similar a"
//hijo = "es indispensable para conocer a"
Hipotesis *Papa;//Apuntador al primer concepto más complejo
Hipotesis *Hijo;//Apuntador a cualquiera de los concepto indispensables para..
Hipotesis *Hermano;//Para navegar entre todos los conceptos indispensables
```

# Arquitectura

```
unsigned char RetornaHijos(void); //Retorna el numero de Hijos de este objeto
short VerificaDiagnostico(void); //Verifica si este concepto se puede diagnosticar
short VerificaEnsenanza(void); //Verifica si este concepto se puede enseñar
void Ajuste(void); //Verifica la consistencia de este nodo y sus hijos
void PropagaVerdadero(void); //propaga los valores SE ASUME CONOCIDO en los hijos
void PropagaFalso(void); //Propaga los valores NO CONOCE en los hijos
void ModelaEstudiante(void); //Verifica el estado de los hijos para saber si es que el
//STI necesita cambiar de estrategia a ANALOGIAS O INDISPENSABLES
unsigned char RetornaLeccion(void); //Busca la lección a enseñar
unsigned char RetornaInd(void); //Busca la pregunta correspondiente
short RetornaEstado(void); //Verifica si ya se acabaron las preguntas y lecciones
char RetornaValorHip(void); //Retorna el Valor de este concepto
```

```
Hipotesis* GeneraObjetivo(void); //Busca un objetivo pedagógico a partir de este concepto
short Agotados(void); //Verifica si todos los conceptos que tienen la liga "es similar a" están
//agotados
void QuitaAgotamiento(void); //Si todos los conceptos que poseen la liga "es similar a "
//están agotados esta función quita tal estado
```

```
//funciones miembro que sirven para construir la red
```

```
void PonValor(int);
void PonHermano(Hipotesis *otra);
void PonPapa(Hipotesis *otra);
void PonHijo(Hipotesis *otra);
void PonLeccion(int ,unsigned char);
void PonInd(int, unsigned char);
```

```
};
```

Los apuntadores Papa, Hijo y Hermano, son en realidad las relaciones "es elemental para conocer" y la relación "es similar a ". El puntero Hermano representa a la relación "es similar a ". El puntero Hijo en conjunción con el puntero hermano, representan la relación "es indispensable para conocer. De esta manera el objeto (Hipotesis) instancia no queda restringido a un número finito de objetos con la relación "es elemental para conocer". El puntero Papa se usa para lograr una mejor navegación dentro de la red pero no tiene uso por el Tutor de una manera "inteligente".

Authorware importa todas las funciones del DLL. Cuando un módulo de Authorware se ejecuta, se invoca la ejecución del DLL, este permanece en memoria hasta que el módulo de Authorware finaliza. De este modo el Modelo del Estudiante permanece en la memoria activa durante la ejecución del STI.

La figura 47 muestra la interfaz que provee Authorware para importar funciones desde un DLL. En la figura solo son mostradas algunas de las funciones que son importadas por Authorware. Todas las funciones exportadas por el DLL e importadas por Authorware son las siguientes:

```
void __declspec(dllexport) Iniciacion(void)
void __declspec(dllexport) PropagaVerdadero(void)
void __declspec(dllexport) PropagaFalso(void)
short __declspec(dllexport) GeneraObjetivo(void)
short __declspec(dllexport) GeneraListaconIncertidumbre(void)
short __declspec(dllexport) Examen(void)
void __declspec(dllexport) BorraListaconIncertidumbre(void)
void __declspec(dllexport) ModelaEstudiante(void)
void __declspec(dllexport) MarcaHipotesis(int value)
void __declspec(dllexport) IniciaHipotesis(int Index, int Value)
short __declspec(dllexport) RetornaLeccion(void)
```

```
short __declspec(dllexport) RetornaEstado(void)
short __declspec(dllexport) ObtieneEstadoTutor(void)
short __declspec(dllexport) RetornaInd(void)
void __declspec(dllexport) AjusteA(void)
short __declspec(dllexport) RetornaValorHip(void)
```

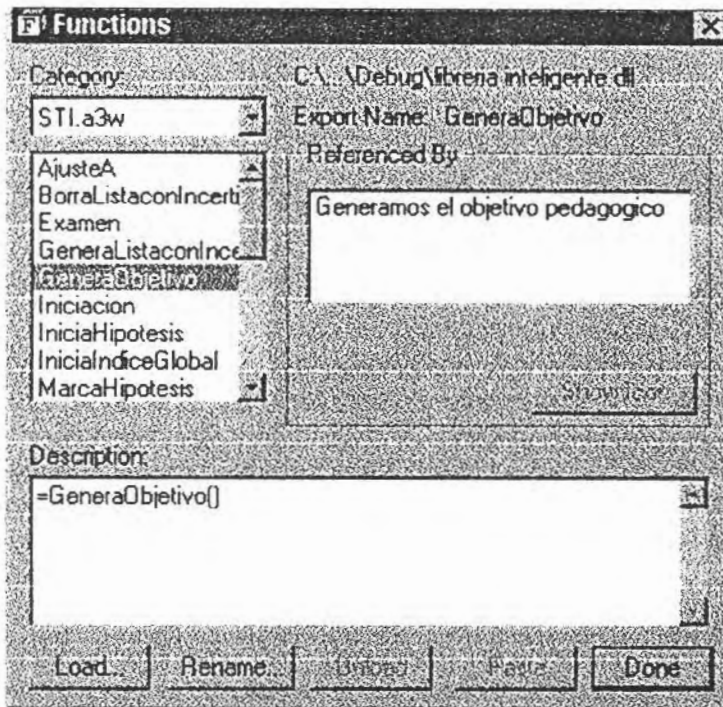


figura 47

Interfaz para importar funciones de una librería de ligado dinámico

Las funciones que se quieren importar son básicamente las funciones miembros de la clase. Authorware no puede importar funciones que contengan el operador de resolución :: (dos dobles puntos) que define C++. Así la estrategia para resolver este problema es incluir las funciones miembro de las clases dentro de funciones que contengan el mismo nombre pero sin el operador de resolución. C++ permite que las funciones sean sobrecargadas. Por ejemplo :

```
extern "C"
short __declspec(dllexport) RetornaLeccion(void)
{
    return ObjetivoPedagogico->RetornaLeccion();
}
```

Así se tiene en el modulo que posee el DLL la red semántica que representa el Modelo del Estudiante. Las funciones miembro de la clase son exportadas en su mayoría por el DLL, excepto las funciones miembro que sirven para construir la red y las funciones que realizan el examen inicial. Así, se tienen tres etapas diferentes que definen procesos dentro de la librería de ligado dinámico.

# Arquitectura

- Inicio y construcción de la red semántica (Modelo del Estudiante)
- Examen
- Uso de la red semántica por el Modulo Tutor

Para el inicio de la red semántica se usan las siguientes funciones miembro que no se exportan al modulo que posee Authorware.

```
void PonHermano(Hipotesis *otra);
void PonPapa(Hipotesis *otra);
void PonHijo(Hipotesis *otra);
void PonLeccion(int ,unsigned char);
void PonInd(int, unsigned char);
```

Después de que la red semántica esta construida en memoria se inicia con los valores que se tienen en la base de datos del alumno en particular. Así se exportan funciones que reciban los datos por el módulo de Authorware e inicien los valores en la red semántica. Así mismo el mismo módulo del DLL debe poseer una función que realice el proceso inverso, el de pasar los datos desde la red semántica hacia la Base de Datos. Las funciones son las siguientes;

```
void __declspec(dllexport) IniciaHipotesis(int Index, int Value)//de la BD al ME
short __declspec(dllexport) RetornaValorHip(int index)//del ME a la BD
```

Las funciones que llevan a cabo el proceso de realizar el examen solo manipulan la red y se llevan a cabo cada vez que el alumno ingresa por vez primera al sistema. Estas funciones también se exportan desde el DLL pero no son funciones miembro de la clase.

```
short __declspec(dllexport) GeneraListaconIncertidumbre(void)
short __declspec(dllexport) Examen(void)
void __declspec(dllexport) BorraListaconIncertidumbre(void)
```

Las funciones que definen al Módulo Tutor en base al Modelo del Estudiante (la red semántica) también son exportadas. Las funciones son las siguientes.

```
void __declspec(dllexport) PropagaVerdadero(void)
void __declspec(dllexport) PropagaFalso(void)
short __declspec(dllexport) GeneraObjetivo(void)
void __declspec(dllexport) ModelaEstudiante(void)
void __declspec(dllexport) MarcaHipotesis(int value)
void __declspec(dllexport) IniciaHipotesis(int Index, int Value)
short __declspec(dllexport) RetornaLeccion(void)
short __declspec(dllexport) RetornaEstado(void)
short __declspec(dllexport) RetornaInd(void)
void __declspec(dllexport) AjusteA(void)
```

La codificación completa de cada una de las funciones exportadas, funciones miembro de la clase hipotesis aparecen en el apéndice A. La especificación de cada una de estas funciones se encuentra en el capítulo III.

## 4.11 Acceso a Bases de Datos

En el momento en que el usuario ejecute el STI se necesita acceder la base de datos que contiene información de los usuarios. Un UDC es una librería de ligado dinámico comercial para el uso de Authorware. Por medio de un UDC que contiene funciones que accesan a la base de datos a través de un administrador de bases de datos comercial conocido como ODBC (Open Database

# Arquitectura

Connectivity) Administrator. De este modo se crea una tabla, definida en el diseño, con el número de usuarios que se deseen. Para el prototipo se crea una tabla con usuarios del departamento de Simulación del IIE y de académicos de la UTM. La tabla se crea mediante el uso de Lotus 123, la tabla generada es de tipo dBase IV, ya que este formato es el más usado en PC's.

## Características de la tabla historia.dbf

Versión: dBase IV

campo	tipo	descripción
N	entero	Numero de ingresos al sistema
USUARIO	carácter	nombre de usuario
NOMBRE	carácter	nombre completo del usuario
CLAVE	carácter	clave de acceso del usuario
H0	entero	valor de la hipótesis 0
H1	entero	valor de la hipótesis 1
H2	entero	valor de la hipótesis 2
H3	entero	valor de la hipótesis 3
H4	entero	valor de la hipótesis 4
H5	entero	valor de la hipótesis 5
H6	entero	valor de la hipótesis 6
H7	entero	valor de la hipótesis 7
H8	entero	valor de la hipótesis 8
H9	entero	valor de la hipótesis 9
H10	entero	valor de la hipótesis 10
H11	entero	valor de la hipótesis 11
H12	entero	valor de la hipótesis 12
H13	entero	valor de la hipótesis 13
H14	entero	valor de la hipótesis 14
H15	entero	valor de la hipótesis 15
H16	entero	valor de la hipótesis 16
H17	entero	valor de la hipótesis 17
H18	entero	valor de la hipótesis 18
H19	entero	valor de la hipótesis 19
H20	entero	valor de la hipótesis 20
H21	entero	valor de la hipótesis 21
H22	entero	valor de la hipótesis 22
H23	entero	valor de la hipótesis 23
H24	entero	valor de la hipótesis 24

Tabla 6

Características de la tabla historia.dbf

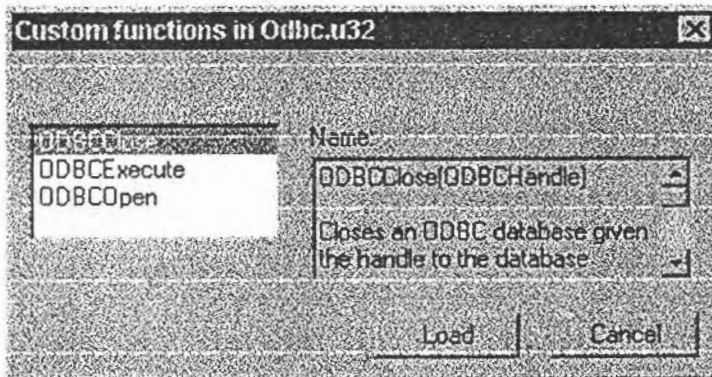
ODBC posee un manejador para dBase (además de otros), a este manejador se le asigna la fuente de datos que se va a usar, en este caso la tabla HISTORIA.DBF y la fuente de datos la se denomina BDSTI. Mediante el UDC se definen las instrucciones que manipulan la base de datos. El UDC denominado ODBC-" posee tres funciones por medio de las cuales se establece la comunicación con el administrador ODBC las cuales son:

Función	Sintaxis / Descripción
ODBCOpen()	<p>ODBCHandle := ODBCOpen(WindowHandle, ErrorVar, Database, User, Password)</p> <p>Abre un fuente de datos ODBC. Retorna un gancho, al cual se hará referencia cada vez que se quiera acceder a esta fuente. ErrorVar es una variable de tipo cadena de caracteres que recibe el posible mensaje de error. Database es la fuente de datos ODBC. User y password no se usan cuando se definen bases de datos de tipo dBase.</p>
ODBCExecute()	<p>data := ODBCExecute(ODBCHandle, SQLString)</p> <p>Envía un comandó SQL al interprete de SQL de ODBC. ODBCHandle es el gancho que retorno la función ODBCOpen function. SQLString es un comando de SQL válido SQL command. El valor de retorno contiene los datos que fueron seleccionados.. Posibles errores se retornan en la variable definida por el usuario.</p>
ODBCClose()	<p>ODBCClose(ODBCHandle)</p> <p>Cierra una fuente de datos ODBC cuyo gancho sea ODBCHandle.</p>

**Tabla 7**

Funciones para acceder a bases de datos proporcionadas por el UDC *ODBC32.UDC*

La figura 48 muestra el modo en que se importan las funciones del UDC y como se acceden a ellas a través de Authorware.



**figura 48**

Interfaz para importar funciones desde un UDC

Esta funciones se usan al inicio y final de una sesión con el STI, La función **ODBCExecute** ejecuta un comando de SQL mediante una cadena de caracteres que se le pasa como argumento. Una vez que se obtiene un dato del registro correspondiente al usuario se pasa al módulo que posee el DLL.

Las instrucciones para obtener datos de la BD y pasarlos al módulo del DLL son las siguientes:

```
--Las siguientes instrucciones se realizan 25 veces
Instruccion:="select h"
Instruccion:=ReplaceString(Instruccion,9,2,AuxString)
Instruccion:=ReplaceString(Instruccion,12,34," from historia.dbf where usuario=")
numero:=CharCount(UserName)
Instruccion:=ReplaceString(Instruccion,46,numero,UserName)
Instruccion:=ReplaceString(Instruccion,46+numero,1,"")
datos:=ODBCExecute(Handle,Instruccion)
IniciaHipotesis(control,datos)
control:=control+1
```

Estas instrucciones se realizan dentro de un ciclo de tal manera que se inicien las 25 Hipótesis del prototipo. La figura 49 y figura 50 muestran como se implementas las funciones en el módulo de Authorware. En la figura 50 se muestra el flujo principal, ahí se encuentra el icono denominado "Inicia las Hipotesis" el cual contiene el código mostrado anteriormente.

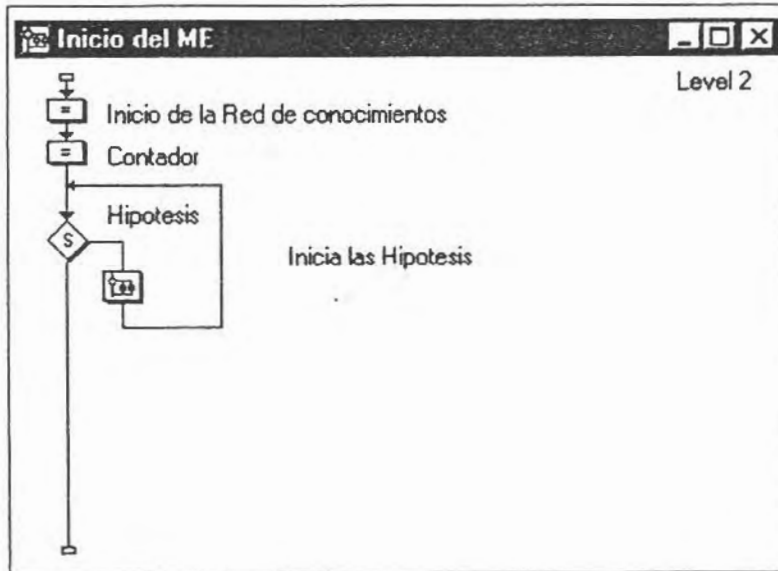


figura 49

Flujo para iniciar las hipótesis desde la BD



figura 50

Iconos que contienen código para acceder a la BD y actualizar el ME

Las instrucciones para actualizar la BD desde el módulo que posee el DLL después de una sesión es el siguiente:

```
auxbd:=RetornaValorHip()
Instruccion:="update historia.dbf set h"
numero:=CharCount(AuxString)
Instruccion:=ReplaceString(Instruccion,26,numero,AuxString)
Instruccion:=ReplaceString(Instruccion,29+numero+1,1,"=")
numero2:=29+numero+1
numero:=CharCount(auxbdstring)
Instruccion:=ReplaceString(Instruccion,numero2+1,numero,auxbdstring)
Instruccion:=ReplaceString(Instruccion,numero2+numero+1,16," where usuario=")
numero2:=numero2+numero+16
numero:=CharCount(UserName)
Instruccion:=ReplaceString(Instruccion,numero2+1,numero,UserName)
Instruccion:=ReplaceString(Instruccion,numero2+numero,1,"")
datos:=ODBCExecute(Handle,Instruccion)
```

La figura 51 y figura 52 muestran la manera en que estas instrucciones se implementan en el módulo de Authorware:





figura 51

flujo para guardar las hipótesis en la BD

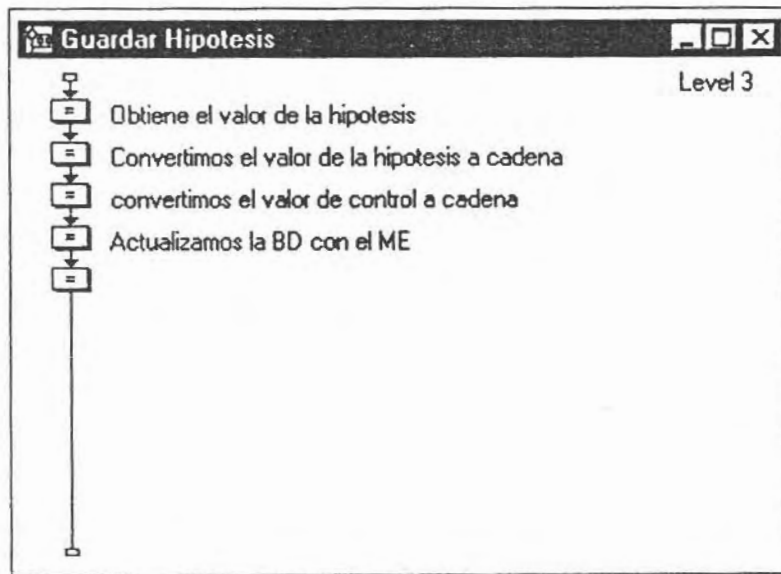


figura 52

Iconos que contienen código para acceder a la BD y actualizarla

Estos iconos se colocan al inicio y al final de una sesión del STI como se mostró en la figura 45

Las instrucciones que verifican el numero de veces que el usuario a ingresado al sistema son las siguientes:

```
Instruccion:="select n from historia.dbf where usuario=""  
numero:=CharCount(UserName)  
Instruccion:=ReplaceString(Instruccion,43,numero,UserName)  
Instruccion:=ReplaceString(Instruccion,43+numero,1,"")  
auxbd:=ODBCExecute(Handle,Instruccion)
```

En la figura 45 en el icono "Obtenemos de la BD en numero de veces que ha ingresado el usuario al sistema" se encuentran codificadas estas instrucciones.

Cuando la variable **auxbd** es igual a cero entonces se le presenta al usuario la interfaz que le permite hacer el examen, como se muestra en la figura 45

El código siguiente una vez obtenido en nombre de usuario busca la clave en la base de datos:

```
Instruccion:="select clave from historia.dbf where usuario=""  
numero:=CharCount(UserName)  
Instruccion:=ReplaceString(Instruccion,47,numero,UserName)  
Instruccion:=ReplaceString(Instruccion,47+numero,1,"")  
datos:=ODBCExecute(Handle,Instruccion)
```

Este código se inserta en el icono que posee el nombre de "Obtención de la clave de acceso para su verificación" como se muestra en la figura 53.

Una vez que se accesa a la base de datos, en el icono que posee el titulo "Comprobación de los datos con la BD" se codifican las siguientes instrucciones:

```
Instruccion:="select nombre from historia.dbf where usuario=""  
numero:=CharCount(UserName)  
Instruccion:=ReplaceString(Instruccion,48,numero,UserName)  
Instruccion:=ReplaceString(Instruccion,48+numero,1,"")  
auxbdstring:=ODBCExecute(Handle,Instruccion)  
NombreDelUsuario:=auxbdstring
```

Estas Instrucciones retornan el nombre completo del usuario.



figura 53

Flujo para comprobar la contraseña del usuario

La figura 54 muestra la relación que existe entre los iconos de calculo y las instrucciones codificadas.

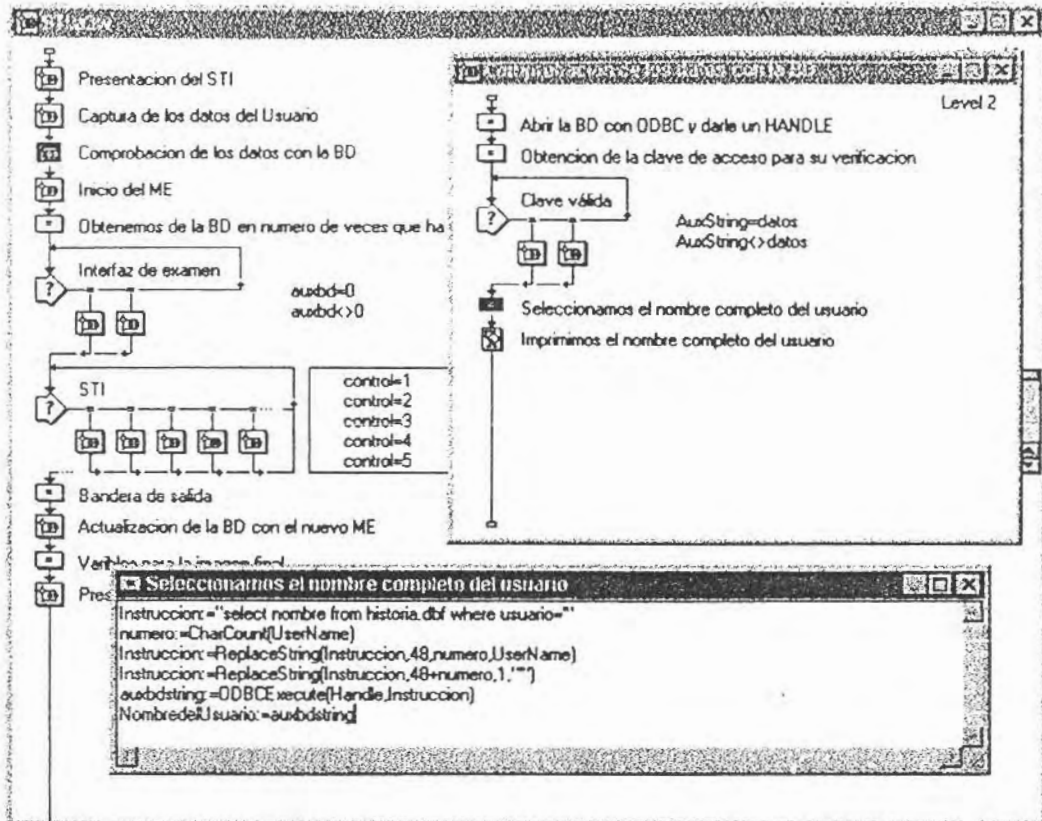


figura 54

Relación entre iconos de calculo e instrucciones

Finalmente en la figura 55 (página siguiente) se muestra de manera esquemática la arquitectura del Prototipo de STI implantado con herramientas Multimedia ( Macromedia Authorware) y lenguaje orientado a objetos (Microsoft Visual C++)..

# Arquitectura

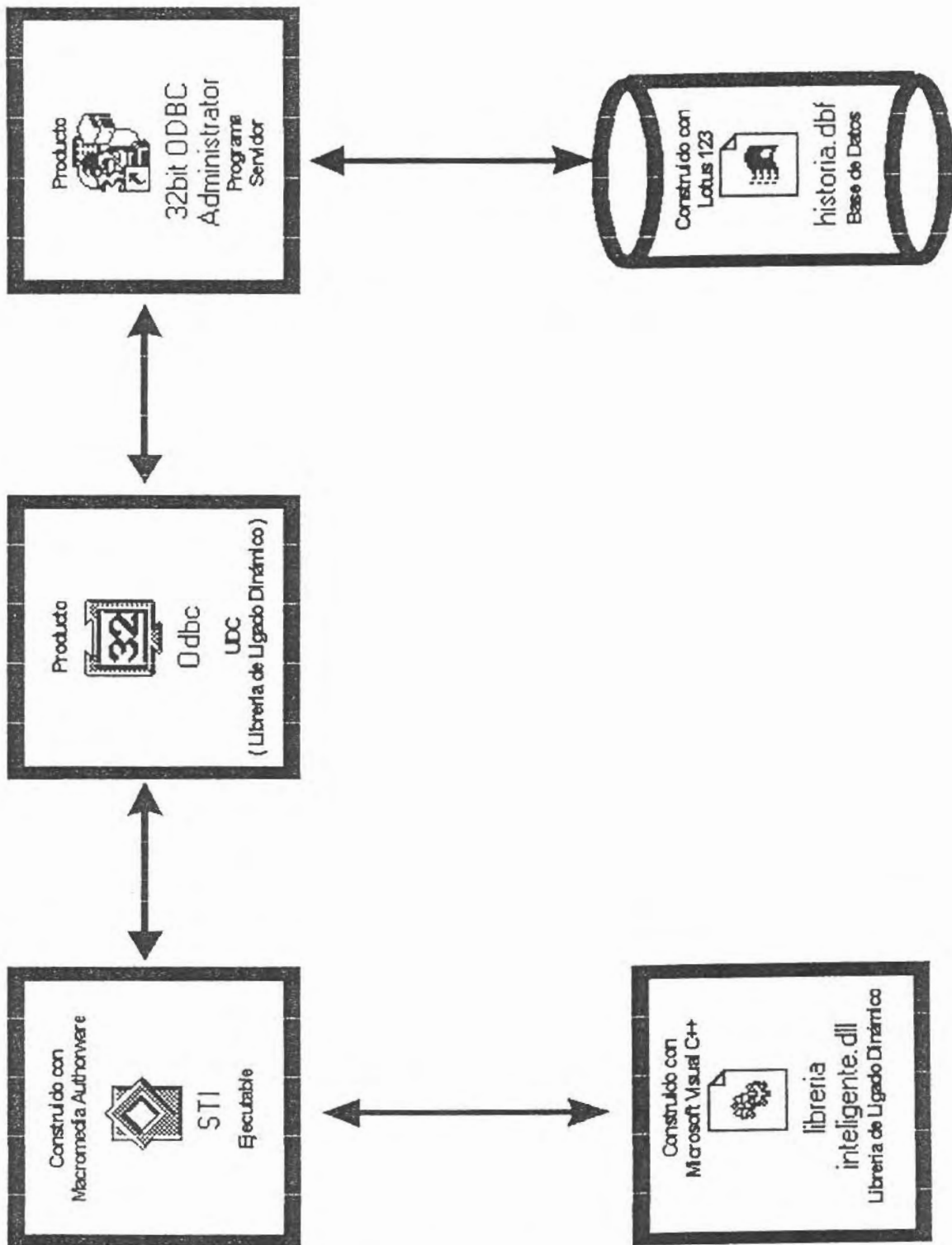


figura 55

Arquitectura en base a las herramientas y lenguajes usados para construir el prototipo

## 4.12 Validación

La validación se llevo a cabo en dos etapas principalmente: la primera etapa consistió en probar exhaustivamente el sistema que posee el algoritmo de toma de decisiones, haciendo énfasis en los Módulo Tutor. En la segunda etapa se probó el prototipo con investigadores del departamento de Simulación del IIE.

En las primera etapa se probaron diferentes tipos de usuarios simulando diferentes tipos de respuestas por parte del estudiante a fin de verificar las diferentes estrategias de enseñanza que sigue el Módulo Tutor en unión con el Modelo del Estudiante.

Específicamente se probaron las características del prototipo siguientes.

- **Mecanismo de ajuste**
- **Búsqueda de objetivo pedagógico**
- **Búsqueda de conceptos similares**
- **Búsqueda de conceptos similares**
- **Búsqueda de conceptos similares (especial)**
- **Búsqueda de un concepto por la liga es elemental para conocer**
- **Búsqueda de conceptos asumidos como conocido**
- **Proceso Agotado y Quita Agotamiento**
- **Los procesos retorna lección y retorna indicador**
- **Para probar la consistencia de la Base de Datos**

### 4.12.1 Pruebas Locales a los Módulos que conforman el prototipo de STI

**Mecanismo de ajuste** : Este mecanismo se probó de tal manera que existieran diferentes incongruencias por parte de las respuestas del alumno, a fin de que en el momento del ajuste el Modelo del Estudiante fuera consistente con la información que posee después de un examen inicial. Esta prueba se hizo mediante el uso de una bitácora que muestra la situación del Modelo del Estudiante al inicio del examen y al finalizarlo. Durante el examen se combinan respuestas incongruentes para forzar el inicio de del Modelo del Estudiante de manera incongruente. Una vez obtenido el Modelo del Estudiante Incongruente se hace el ajuste a mano y se corrobora al momento de que el Sistema proporciona ya sea la primer pregunta para hacer un diagnóstico o la lección correspondiente de un concepto. Como se tiene el Modelo del Estudiante a la mano entonces se sabe de antemano que lección debe proporcionar o la pregunta que debe hacer. Aquí también se verificaron los procesos **PropagaVerdadero** y **PropagaFalso**

**Búsqueda de objetivo pedagógico:** Para la búsqueda del objetivo pedagógico las pruebas consistieron en que una vez obtenido el Modelo del Estudiante inicial, el Módulo Tutor en conjunción con el Modelo del Estudiante (en concreto la función miembro de la clase **Hipotesis** que corresponde a **Hipotesis::GeneraObjetivo** ) debe buscar por la red en cada uno de los valores que posee el ME por cada concepto, ya indican cual debe tomarse como objetivo pedagógico, los valores -1 y 0 indican que debe tomarse como objetivo de enseñanza o diagnóstico respectivamente, así cuando el ME esta iniciado esta función debe retornar el concepto adecuado. Así el Sistema muestra la lección o la pregunta correspondiente al concepto tomado como objetivo pedagógico. En esta prueba se contestan las preguntas de manera acertada tanto las de diagnóstico como aquellas que se presentan después de cada lección . Se sigue por el Modelo del Estudiante iniciado el orden de los conceptos que deben ser tomados y se verifica que el Sistema los tome como objetivo.

# Arquitectura

**Búsqueda de conceptos similares:** Esta prueba se hizo con el fin de verificar que el ME y el Módulo Tutor buscan conceptos similares cuando se agota un concepto. La prueba consistió en agotar alguno de los conceptos del ME del prototipo una vez iniciado el ME. Los conceptos que se tomaron en cuenta son : del concepto 0 al concepto 3; del concepto 6 al concepto 11; del concepto 19 al concepto 21; del concepto 15 al concepto 17; el concepto 5 y concepto 12; el concepto 13, 23 y 24; el concepto 22 y el concepto 18. Aquí la prueba consiste en agotar cualquiera de estos conceptos y esperar una reacción del sistema que indique el agotamiento, así el Sistema busca otro concepto (si es que los conceptos similares pueden tomarse como objetivo pedagógico) y lo presenta como objetivo.

**Búsqueda de conceptos similares (especial):** Un caso especial de búsqueda de conceptos similares se prueba en los conceptos 4 y 14. Aquí la prueba consiste en agotar estos conceptos y esperar que el Sistema no encuentre concepto similar que pueda ser tomado como objetivo pedagógico. De este modo el Sistema continúa con la enseñanza del mismo concepto. Aquí también se prueba que el Sistema reinicie las listas del concepto (objeto) a fin de continuar con la enseñanza.

**Búsqueda de un concepto por la liga "es elemental para conocer" :** Esta prueba consistió en probar cualquiera de los conceptos que no son hojas en la red semántica. Por ejemplo del concepto 0 al concepto 3 fueron puestos con un valor 2 o 1. De este modo al momento en que el Sistema busca el objetivo pedagógico debe retornar el concepto 4. Así el comportamiento del sistema fue probado con todos los conceptos que no son hoja.

**Búsqueda de conceptos asumidos como conocido:** Aquí la prueba consistió en localizar un concepto que tuviera varios conceptos *elementales para conocer* . Por ejemplo , el concepto 4 del ME del prototipo posee 4 conceptos ligados de este modo. Aquí se forzó a que el sistema asumiera a uno de estos conceptos como conocido (valor =1 ) una vez obtenido este estado del ME el concepto 4 fue tomado como objetivo pedagógico. Aquí se forzó a que el concepto se agotara, de este modo, el sistema reacciono de manera tal que modifico los valores de los conceptos asumidos como conocidos y los tomo como objetivo pedagógico. Aquí los conceptos que se tomaron como objetivo fueron forzados a ser aprendidos por el alumno (forzando las respuestas a ser acertadas por el estudiante). Después el Sistema tomo nuevamente al concepto 4 como objetivo pedagógico. Para verificar la consistencia de este proceso, la misma prueba se hizo con los conceptos 12, 13, 22, 18, 23 y 14. Dicho de otra manera en esta prueba se verifico se verificó el Proceso **ModelaEstudiante** ya que es el encargado de modificar al ME en estas condiciones

**Proceso Agotado y Quita Agotamiento:** La prueba de este proceso se verificó de la manera siguiente: Se agotaron todos los conceptos similares que correspondan a un concepto por medio de la liga "es elemental para conocer". Por ejemplo los conceptos 0,1,2,3. Una vez que todos estos conceptos están en esta condición, se verifico que el sistema restableciera todas las listas que corresponden a estos conceptos, quitara por medio del proceso **QuitaAgotamiento** esta condición y nuevamente los tomara como objetivo pedagógico. Estos procesos prueban a su vez el proceso **RetornaEstado** ya que verifica cuando las listas de indicadores y lecciones están agotadas.

Los procesos **retorna lección** y **retorna indicador** se probaron de la manera siguiente: Como de antemano se conocen desde la implementación cuantos indicadores y lecciones poseen cada concepto entonces al momento de agotar un concepto se verificó que ambas listas fueran usadas completamente. Así mismo se verifico que cada vez que una lista fuera agotada entonces al momento de que el objeto tomara un elemento de esta fuera de manera aleatoria.

Para probar la **consistencia de la Base de Datos** y las funciones implementadas para la Actualización del ME desde la BD y viceversa, la prueba consistió en abandonar una sesión. Al momento de que una nueva sesión se abre con el mismo usuario el Sistema continuo en el

contexto anterior en que fue abandonado el sistema. Así mismo se verificó que el Sistema no vuelve a presentar la opción de hacer un examen preliminar.

Todas estas pruebas fueron hechas de manera exhaustiva con el fin de verificar la consistencia de los procesos que sigue el Módulo Tutor en conjunción con el Modelo del Estudiante.

## 4.12.2 Evaluación del prototipo de STI frente a usuarios.

La segunda parte de la prueba consistió en tomar una muestra de posibles estudiantes. Así se tomaron 6 personas del departamento de Simulación del IIE. El objetivo de esta prueba era que los usuarios verificaran el comportamiento del prototipo como un medio de enseñanza inteligente. De este modo, se logró que los participantes llevaran a cabo una sesión completa con el prototipo. En este punto se podrían aseverar que las pruebas de un sistema como este prototipo son subjetivas, por ejemplo, todos los estudiantes estuvieron de acuerdo con el diseño, la presentación y funcionalidad del prototipo.

Como conclusión de esta segunda etapa de pruebas se puede decir que el sistema se adaptó de manera apropiada a las necesidades de cada uno de los estudiantes. De tal modo que para completar el curso del área de estudio los tiempos fueron diferentes, así como la presentación de lecciones, preguntas y exámenes iniciales. El prototipo del Sistema cumplió con los objetivos de un STI, donde uno de los principales es el de ajustarse a las necesidades del estudiante que está frente al sistema. También se puede concluir que además de las características que ofrecen las aplicaciones Multimedia para el aprendizaje, los alumnos cumplieron con el objetivo de aprender de manera guiada.



# Capítulo cinco

## Conclusiones y trabajos futuros

### 5.1 Conclusiones

En el presente trabajo se abordó el tema de STI, dedicando mayor esfuerzo y atención al Modelo del Estudiante, el cual, la mayoría de los investigadores en el área lo consideran como el componente principal de un STI, ya que es el encargado de retroalimentar al STI para guiar y completar el proceso de enseñanza sobre un tema.

Así el Modelo del Estudiante seleccionado para la construcción del prototipo de STI es del tipo de superposición, implantado utilizando la representación de conocimiento de redes semánticas, lo cual permite que fácilmente se determine el objetivo pedagógico en función de la información que el estudiante recibe durante todo el proceso de enseñanza - aprendizaje.

El prototipo de Sistema Tutorial Inteligente barca los cuatro Módulos básicos mencionados en la mayoría de los artículos sobre el tema analizados en la etapa de revisión bibliográfica; interfaz, dominio de aplicación, algoritmo de toma de decisiones (Modulo Tutor) y Modelo del Estudiante. Estos dos últimos son los que le dan las características de inteligente al prototipo de STI

En [24] se mencionan 5 componentes de instrucción en los STI. El Algoritmo propuesto posee las cinco características propuestas por el mismo.

El diseño del prototipo de STI presentado estuvo restringido a la utilización de herramientas disponibles en el lugar de trabajo. Así la interfaz y la base de conocimientos fueron construidas usando la herramienta Macromedia Authorware dado la rápida implementación de material de enseñanza. El algoritmo de toma de decisiones y el modelo del estudiante fueron construidos usando la herramienta Microsoft Visual C++ dadas las características que posee para la implementación de programación orientada a objetos. El problema radicó en que la herramienta Authorware no esta diseñada para representar conocimiento en el sentido impuesto por las técnicas de IA.

Finalmente la arquitectura del prototipo desarrollado permite fácilmente cambiar de dominio de aplicación sin llevar a cabo modificaciones de fondo, solo de forma, desarrollando nuevas lecciones y codificando el nuevo modelo del estudiante. Lo cual permite establecer que la arquitectura propuesta es genérica, concepto ampliamente buscado en el ámbito de los STI.

### 5.2 Perspectivas y trabajos futuros

El diseño de la arquitectura desarrollada considera exclusivamente la enseñanza de conceptos académicos. En seno de la Unidad de Resultados de Simulación del IIE se trabaja generalmente en simuladores para centrales eléctricas, lo cual sugiere la enseñanza de habilidades. Esta parte puede incluirse en la arquitectura del STI propuesto considerando por un lado que en el Modelo del Estudiante (red semántica) los nodos en lugar de conceptos a aprender sean habilidades (maniobras a realizar) guardando las misma relaciones. Por otro lado, es interesante validar el prototipo de STI desarrollado en grupo de alumnos interesados en el tema eléctrico con el fin de evaluar su capacidad de enseñanza.

Otra perspectiva importante es la de considerar la incorporación de más relaciones semánticas como las de "especialización", "generalización" con el objetivo de proponer más estrategias de enseñanza.

Finalmente considerar la posibilidad de implementar un STI utilizando solamente una herramienta computacional (visual C++, Visual J++ o Common Lisp) es una perspectiva interesante.

## Referencias

### Reportes

- [1] Barr Avron & Edward A. Feigenbaum "The Handbook of Artificial Intelligence" Department of Computer Science, Stanford University.
- [2] Buenabad, María de Los Angeles "Sistema de Capacitación de Operadores en la Sintonización de Control"; Programa de adiestramiento en Investigación Tecnológica, grupo X, Instituto de Investigaciones Eléctricas.
- [3] Cialdea, Martha "Meta - Reasoning and Student Modeling" Dipartimento di Informatica e Sistemistica di Roma, La Sapienza, via Salaria, Roma, Italia.
- [4] Corbett, Albert T. & Jhon R. Anderson "Knowledge Tracing: Modeling the Acquisition of procedural Knowledge" Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- [5] Corbett, Albert T. & Jhon R. Anderson " Student Modeling in an Intelligent Programming Tutor" Psychology Department, Carnegie Mellon University, Pittsburgh. PA.
- [6] Corral Díaz, Sonia Edith "Tutor para el entrenamiento en la operación de Reguladores Automáticos de Tensión" Programa de adiestramiento en Investigación Tecnológica, grupo X, Instituto de Investigaciones Eléctricas.
- [7] Chang, Kuo-En & Wen-Chuan Hou "Application of Neural Network for Implementing a Practical Student Model" Department of Information Computer Education, National Taiwan Normal University.
- [8] Davis, Randall & Howard Ahrobe & Peter Szolovits "What is a Knowledge Representation ?" MIT AI Lab, MIT AI Lab and Symbolics, Inc., MIT Lab for Computer Science  
<http://medg.lcs.mit.edu/ftp/psz/k-rep.html>
- [9] Elsom, Mark - Cook "Student Modeling in intelligent tutoring systems"; Artificial Intelligence Review 7, 227-240, 1993, Kluwer Academic Publishers, Printed in Netherlands.
- [10] Errico, Bruno & Alessandro Micarelli "From Student Modeling to User Modelling"; Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma La Sapienza, Roma Italia, Instituto de Elettrotecnica, Università di Cagliari, Piazza d'Armi, Cagliari, Italia.
- [11] Goodkovsy, Valdimir A "Intelligent Tutor: Top Down Approach to ITS Design" Moscow State Engineering and Physics Institute, Moscow Russia.  
[Http://sih.readadp.com/p1484/goodkov/goodkov.htm](http://sih.readadp.com/p1484/goodkov/goodkov.htm)
- [12] Harp, Steven & A. Tariq Samad & Michael Villano " Modeling Student Knowledge with Self - Organizing Feature Maps" IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No.5, May 1995.
- [13] Holt, Peter & Shelli Dubs & Marlene Jones & Jim Greer "The State of Student Modeling" Computer Science and Information System, Athabaska University, Athabaska, Canada
- [14] Ignizio, James P. "Introduction to Expert Systems".

# Referencias

- [15] Itoh, Yukihiro & Isamu Takahashi & Tatsuhiro Konishi "On Recognition of Student's Actions in ILE" Faculty of engineering, Shizuoka University.
- [16] Kay J. & R.J. Kummerfeld "An Individualized Course for the C Programming Language" University of Sydney.  
[Http://www.cs.su.oz.au/~bob/kay-kummerfeld.html](http://www.cs.su.oz.au/~bob/kay-kummerfeld.html)
- [17] Livergood, Normand D "From Computer Assisted Instruction to Intelligent Tutoring Systems" School of Computer and Applied Sciences, Los Angeles, CA.
- [18] Mandl, Heinz & Aemilian Hron "Cognitive Theories as a Basis for Student Modeling" Universitat Muchen, Institut fur Empirische Padagogische Pyscologie, Leopoldtr FRG: Deutsches Institut fur Fernstudien an der Universitat Tuingen, Arbeitsbereich Lernforschung, Konrad - Adenauer Alle Tuingen, FRG.
- [19] McCalla, Gordon I. "The Central Importance of Student Modeling to Intelligent Tutoring" ARIES Laboratory, Department of Computational Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada.
- [20] Niedzwecki, Jhon M.& Jessie A. Earles "Strategy for Design of Intelligent Tutoring Systems" Journal of Computing in Civil Engineering, Vol. 5 No.2, April 1991.
- [21] Nour, Mohamed & Ehsan Abed & Nadia Hegazi "A Proposed Student Model Algorithm for an Intelligent Tutoring System" Electronics Research Institute, National Research Center Building, Cairo, Egypt.
- [22] Petrushin, V.A " Intelligent Tutoring Systems: Architecture and Methods of Implementation" Journal of Computer and Systems Sciences International..
- [23] Scaruffy, Piero & Caterina Barbetti "A domain Independent Framework for Tutoring System" Olivetty Artificial Intelligent Center.
- [24] Self, Jhon "The Ebb and Flow of Student Modeling" Department of Computing Lancaster University, Lancaster LA, England.
- [25] Shank, Roger C. "Teaching Architectures" The Institute for the Learning Sciences.
- [26] Shute, Valerie J. "SMART: Student Modeling Approach for Responsive Tutoring" User Modeling and User-Adapted Interaction 5: 1-44, 1995 Kluwer Academic Publishers. Printed in Netherlands.
- [27] Sleeman, Derek; "Intelligent Tutoring Systems"  
<http://pigeon.cad.abdn.ac.uk/research/its.html>
- [28] Stauffer, Karen; "Student Modeling and Web-based Learning Systems"  
<http://ccism.pc.athabascau.ca/html/ccis/research/initsm.htm>
- [29] Syang, Angel & Nell B. Dale " Computerized Adaptive Testing in Computer Science" Department of Computer Sciences, The University of Texas, Austin.
- [30] Tsybenko, Yury "Device Models in Student Modeling" Intelligent Tutoring System Lab, Glushkov Institute for Cybernetics, Kiev, Ukraine.
- [31] VanLehn, Kurt "Student Modeling" Foundation of Intelligent Tutoring Systems.

# Referencias

- [32] Verdejo, M.F. "Building a Student Model for an Intelligent Tutoring System" E.T.S. Ingenieros Industriales U.N.E.D. Ciudad Universitaria, Madrid, España.
- [33] Villano, Michael "Probabilistic Student Models: Bayesian Belief Networks and Knowledge Space Theory" Sensor and Systems Development Center Honeywell Inc., Minneapolis.
- [34] Woolf, Beverly Park & Tom Murray "Using Machine Learning to Advise a Student Model" Department of Computer Science, University of Massachusetts.
- [35] Woolf, Beverly Park & Wendy Hall "Multimedia Pedagogues" IEEE Computer, May 1995.

## Libros, Manuales y revistas

Authorware Reference Macromedia Authorware, Macromedia.

Connecting with Databases Macromedia Authorware, Macromedia.

Getting Ready Macromedia Authorware, Macromedia.

High - Tech Training, Byte Magazine, December 1994.

Multimedia Essentials for Windows, Macromedia.

Taking the Plunge Macromedia Authorware, Macromedia.

Tips & Tricks Macromedia Authorware, Macromedia.

Using Authorware Macromedia Authorware, Macromedia.

Visual C++ Books Online, Microsoft Corp.

## Apéndice A

En este apéndice se encuentra el código completo de la librería de ligado dinámico **librería inteligente.dll**. Esta librería fue compilada y ligada usando Microsoft Visual C++ versión 4.0. La compilación con contiene ninguna especificación en especial. El ligado (link) solo contiene la característica de generar un DLL.

```
//Liberia de ligado dinámico
//librería inteligente.dll
//para el uso del sistema STI.EXE
#include<windows.h> //librería para incluir las características de Windows
#include <time.h> //librería para incluir las funciones de números aleatorios
//para verificar si un concepto puede ser diagnosticado
#define NOHAYDIAGNOSTICO 500
#define HAYDIAGNOSTICO 501

//para verificar si un concepto puede ser enseñado
#define NOHAYENSENANZA 502
#define HAYENSENANZA 503

//Estrategias de enseñanza
//también puede ser NORMAL
//depende del modelado
#define ANALOGIA 504

//Estados del Tutor
#define ENSENA 506
#define DIAGNOSTICA 507
#define EXAMINA 508

#define NUMERODEHIPOTESIS 24 //Numero de hipótesis (conceptos) que contiene el ME

#define FINTUTOR 509 //para definir cuando el STI a finalizado con la enseñanza

#define NOHAYEXAMEN 512//define cuando el estudiante no ha querido hacer un examen inicial

//estados del STI, en caso de que el alumno no muestre comprensión acerca de un concepto y se
//agoten las lecciones o indicadores el STI pasa al estado de MODELA
#define MODELA 514
#define NORMAL 515

class Hipotesis
{
public:
    //Constructores
    Hipotesis(unsigned char,unsigned char, int, int);
    Hipotesis(void){Hijos=0;Valor=0,Identificador=0;}

    char Valor;//-1=NOCONOCE, 0=NOHATINFO, 1=SEASUMECONOCIDO, 2=CONOCIDO
    char Hijos;// Cuantos conceptos son indispensables para conocer este concepto
    unsigned char Identificador;// Numero que identifica a la hipótesis
    unsigned char *ptrInd;//Apuntador a los indicadores de este objeto
    unsigned char *ptrLecciones;//Apuntador a las lecciones de este objeto
    int NumeroInd;//Numero de indicadores de este objeto
    int NumeroLecciones;//Numero de lecciones de objeto
    int IndiceInd;//Indice del ultimo indicadores usado de este objeto
    int IndiceLecciones;//Indice de la ultimo lecciones usada de este objeto
    int Checado;

    Hipotesis *Papa; //Apuntador a el primer concepto más complejo
    Hipotesis *Hijo;// Apuntador a cualquiera de los concepto indispensables para..
    Hipotesis *Hermano; //Para navegar entre todos los conceptos indispensables

    unsigned char RetomaHijos(void);// Retoma el numero de Hijos de este objeto
    short VerificaDiagnostico(void);// Verifica si este concepto se puede diagnosticar
    short VerificaEnsenanza(void);//Verifica si este concepto se puede enseñar
```

# Apéndice A

```
void Ajuste(void); //Solo para verificar la consistencia de este nodo y sus hijos
void PropagaVerdadero(void); //propaga los valores SE ASUME CONOCIDO en los hijos
void PropagaFalso(void); //Propaga los valores NO HAY INFO en los hijos
void ModelEstudiante(void); //Verifica el estado de los hijos para saber si es que el
//STI necesita cambiar de estrategia a ANALOGIAS O INDISPENSABLES
unsigned char RetornaLeccion(void);
unsigned char RetornaInd(void);
short RetornaEstado(void);
char RetornaValorHip(void);

Hipotesis* GeneraObjetivo(void);
short Agotados(void);
void QuitaAgotamiento(void);

//funciones miembro que sirven para construir la red

void PonValor(int);
void PonHermano(Hipotesis *otra);
void PonPapa(Hipotesis *otra);
void PonHijo(Hipotesis *otra);
void PonLeccion(int ,unsigned char);
void PonInd(int, unsigned char);
};

Hipotesis::Hipotesis(unsigned char Children, unsigned char Identifier, int Lessons, int Indicators)
{
    Hijos=Children;
    Identificador=Identifier;
    NumeroInd=Indicators;
    NumeroLecciones=Lessons;
    IndiceInd=1;
    IndiceLecciones=1;
    ptrInd=new unsigned char[NumeroInd];
    ptrLecciones=new unsigned char[NumeroLecciones];
    Checado=0;
}

unsigned char Hipotesis::RetornaHijos(void) //Para conocer el numero de conceptos que tienen la
//liga "es indispensable para conocer"
{
    return Hijos;
}

char Hipotesis::RetornaValorHip(void) //para conocer el estado de un concepto
{
    return Valor;
}

short Hipotesis::VerificaDiagnostico(void) //verifica el diagnostico de un nodo respecto a sus descendientes
{
    if(Hijos==0)
    {
        return HAYDIAGNOSTICO;
    }
    else
    {
        Hipotesis *Auxiliar;
        Auxiliar=Hijo;
        for(int i=0; i<Hijos;i++)
        {
            if(Auxiliar->Valor<1)
            {
                return NOHAYDIAGNOSTICO;
            }
            else
            {
                Auxiliar=Auxiliar->Hermano;
            }
        }
    }
}
```

# Apéndice A

```
        return HAYDIAGNOSTICO;
    }
}

short Hipotesis::VerificaEnsenanza(void) //verifica la enseñanza de un nodo respecto a sus descendientes
{
    if(Hijos==0)
    {
        return HAYENSENANZA;
    }
    else
    {
        Hipotesis *Auxiliar;
        Auxiliar=Hijo;
        for(int i=0; i<Hijos;i++)
        {
            if(Auxiliar->Valor<1)
            {
                return NOHAYENSENANZA;
            }
            else
            {
                Auxiliar=Auxiliar->Hermano;
            }
        }
        return HAYENSENANZA;
    }
}

void Hipotesis::Ajuste(void)//ajusta el modelo del estudiante si es que encuentra incongruencias
{
    Hipotesis *Busca;
    if(Hijos==0) //si no hay hijos no es necesario ajustar
        return;
    else
    {
        Busca=Hijo;
        for(int i=0; i<Hijos ; i++)
        {
            Busca->Ajuste();//Hay que bajar primero
            if(Valor==2 && Busca->Valor<=0)
            {
                Valor=0; //modifica a NOHAYINFO
            }
            else if(Valor==1 && Busca->Valor<=0)
            {
                Valor=-1; //modifica a NOCONOCE
            }
            else
            {
                Busca=Busca->Hermano;
            }
        }
        return;
    }
}

void Hipotesis::PropagaVerdadero(void)//En el examen una respuesta verdadera genera la //propagación de valores por la liga
"es indispensable para conocer"
{
    Hipotesis *Auxiliar;
    if(Hijos==0)
    {
        Valor=1;
        return;
    }
    else
    {
        Auxiliar=Hijo;
        for(int i=0; i<Hijos; i++)
        {
            if (Auxiliar->Checado==1)//si este nodo fue examinado entonces no se
```

# Apéndice A

```
//propagan los valores por este nodo
    }
    else
    {
        Auxiliar->Valor=1;//asumimos que los demás son conocidos
        Auxiliar->PropagaVerdadero();//y aun hasta abajo del concepto
    }
    Auxiliar=Auxiliar->Hermano;
}
}
```

void Hipotesis::PropagaFalso(void) //En el examen una respuesta falsa genera la  
//propagación de valores por la liga "es indispensable para conocer"

```
{
    Hipotesis *Auxiliar;
    //Valor=-1 //No conoce
    if(Hijos==0)
    {
        Valor=-1;
        return;
    }
    else
    {
        Auxiliar=Hijo;
        for(int i=0; i<Hijos; i++)
        {
            //propagan los valores por este nodo
            if (Auxiliar->Checado==1) //si este nodo fue examinado entonces no se
            {
            }
            else
            {
                Auxiliar->Valor=-1;
                Auxiliar->PropagaFalso();
            }
            Auxiliar=Auxiliar->Hermano;
        }
    }
}
```

void Hipotesis::ModelaEstudiante(void) //cuando el estudiante no comprende un concepto y las  
//lecciones fueron agotadas y los indicadores entonces se baja por la relación es indispensable en  
//busca de un concepto que haya sido asumido por conocido, si se encuentra el ME se modifica

```
{
    IndiceInd=1;
    IndiceLecciones=1;
    if(Hijos==0)
    {
        Valor=-2;
        return;
    }
    else
    {
        Hipotesis *Auxiliar;
        Auxiliar=Hijo;
        char edo=0;
        for(int i=0; i<Hijos ; i++)
        {
            if(Auxiliar->Valor==1)
            {
                edo=1;
                Auxiliar->Valor=-1;
            }
            else
            {
            }
            Auxiliar=Auxiliar->Hermano;
        }
        if (edo==0)
    }
}
```



```

        {
            Valor=-2;
            return;
        }
        Valor=0;
        return;
    }
}
void Hipotesis::PonLeccion(int Index, unsigned char Lesson)//agrega un índice que hace referencia
//a una lección de la agenda de este concepto
{
    if((Index>0) && (Index<=NumeroLecciones))
        ptrLecciones[Index-1]=Lesson;
}

void Hipotesis::PonInd(int Index, unsigned char Indicator)//agrega un índice que hace referencia a
//un indicador (pregunta ) de este concepto
{
    if((Index>0) && (Index<=NumeroInd))
        ptrInd[Index-1]=Indicator;
}

unsigned char Hipotesis::RetornaLeccion(void)//retorna una lección para este concepto
{
    if(IndiceLecciones>NumeroLecciones)
        {/si todas las lecciones fueron ya usadas se toma una al azar
        float NumeroAleatorio=(float)((float)rand()/(32767.00));
        int Nuevolndice=(int)(NumeroAleatorio*(float)(NumeroLecciones));
        if (Nuevolndice<0)
            Nuevolndice=0;
        else
        {
        }
        return ptrLecciones[Nuevolndice];
    }
    else
    {
        unsigned char raro=ptrLecciones[IndiceLecciones-1];
        IndiceLecciones++;
        return raro;
    }
}

unsigned char Hipotesis::RetornaInd(void) //retorna un indicador (pregunta) para este concepto
{
    if(IndiceInd>NumeroInd)
        {/si todos los indicadores fueron ya usados se toma una al azar
        float NumeroAleatorio=(float)((float)rand()/(32767.00));
        int Nuevolndice=(int)(NumeroAleatorio*(float)(NumeroInd));
        if (Nuevolndice<0)
            Nuevolndice=0;
        else
        {
        }
        return ptrInd[Nuevolndice];
    }
    else
    {
        unsigned char raro=ptrInd[IndiceInd-1];
        IndiceInd++;
        return raro;
    }
}

short Hipotesis::RetornaEstado(void)//retorna el estado del STI, si las lecciones e indicadores
//fueron agotados entonces el STI pasa al estado MODELA a fin de que el Módulo Tutor realice
//una modificación del ME
{

```

# Apéndice A

```
if((IndiceInd>NumeroInd)&&(IndiceLecciones>NumeroLecciones))
{
    return MODELA;
}
else
{
    return NORMAL;
}
}

void Hipotesis::PonValor(int Value)// para el uso de inicio del ME ya sea en el examen o como una
//actualización desde la BD
{
    if(Value>2) Valor=2;
    else if(Value<=-2) Valor=-2;
    else Valor=Value;
}

void Hipotesis::PonHermano(Hipotesis *otra)//para armar la red semántica (ME)
{
    Hermano=otra;
}

void Hipotesis::PonPapa(Hipotesis *otra) //para armar la red semántica (ME)
{
    Papa=otra;
}

void Hipotesis::PonHijo(Hipotesis *otra) //para armar la red semántica (ME)
{
    Hijo=otra;
}

Hipotesis* Hipotesis::GeneraObjetivo(void)
{
//Esta función miembro hace una búsqueda en profundidad por el ME (red semántica) en busca del
//objetivo pedagógico en base a las estrategias de enseñanza diseñadas
    Hipotesis *Aux;
    Hipotesis *NuevoObjetivo;
    if(Hijos!=0)
    {
        //////////////////////////////////////
        //Primero verificamos a los hijos
        Aux=Hijo;
        for(int i=0; i<Hijos; i++)
        {
            NuevoObjetivo=Aux->GeneraObjetivo();//recursividad hay que bajar //primero
            if(NuevoObjetivo!=NULL)//Se encontró objetivo por este nodo
                return NuevoObjetivo;
            else//No se encontró objetivo por este nodo, hay que moverse por otro
                Aux=Aux->Hermano;
        }
        //Ninguno de los hijos puede tomarse como objetivo
        //Nótese que No estamos en una hoja de la red semántica
        //////////////////////////////////////

        if(Valor==1)//si este concepto tiene valor NOCONOCE entonces debe ser
        //tomado como objetivo pedagógico
        {
            return this;
        }
        else if(Valor==0) //si este concepto tiene valor NOHAYINFO entonces debe ser
        //tomado como objetivo pedagógico
        {
            return this;
        }
        else if(Valor==-2)
        {
            //Este concepto esta agotado
        }
    }
}
```

# Apéndice A

```
Aux=Hermano;
if(Papa!=NULL)
{
    for(int i=0; i<Papa->RetornaHijos() ; i++)
    {
        if((Aux->Valor==1)&&(Aux->VerificaEnsenanza()==HAYENSENANZA))
            return Aux;
        else if((Aux->Valor==0) && (Aux->VerificaDiagnostico()==HAYDIAGNOSTICO))
            return Aux;
        else
            Aux=Aux->Hermano;
    }
}
else
{
    //es el concepto más abstracto y la cabeza del árbol, es el
    //ultimo que hay que enseñar
    Valor=1;
    return this;
}

//No se encontró concepto análogo
//se busca si los conceptos por la liga "es similar a" están todos agotados
int Estado=Agotados();
if(Estado==1)//Si están todos los conceptos agotados
{
    QuitaAgotamiento();//se modifican los conceptos
    Valor=1;//se enseña este concepto
    return this;
}
else//No están todos los conceptos agotados
{
    Valor=1;//se enseña este concepto
    return this;
}
}
else if(Valor==1)//este concepto se asume comprendido por el alumno, no se puede tomar como
//objetivo pedagógico
    return NULL;
else //este concepto es comprendido por el alumno, no se puede tomar como objetivo pedagógico
    return NULL;
}
else//si el concepto dentro del ME es hoja, entonces realiza la recursividad para buscar en profundidad
{
    if(Valor==1)
    {
        return this;
    }
    else if(Valor==0)
    {
        return this;
    }
    else if(Valor==2)
    {
        Aux=Hermano;
        for(int i=0; i<Papa->RetornaHijos(); i++)
        {
            if((Aux->Valor==1) && (Aux->VerificaEnsenanza()==HAYENSENANZA))
                return Aux;
            elseif((Aux->Valor==0)&&(Aux->VerificaDiagnostico()==HAYDIAGNOSTICO))
                return Aux;
            else
                Aux=Aux->Hermano;
        }
        int Estado=Agotados();
        if(Estado==1)
        {
            QuitaAgotamiento();
            Valor=1;
            return this;
        }
    }
}
```

# Apéndice A

```
        }
        else
        {
            Valor=-1;
            return this;
        }
    }
    else if(Valor==1)
        return NULL;
    else
        return NULL;
}

}
short Hipotesis::Agotados(void)//verifica que los nodos por la relación "es similar a" estén agotados
{
    Hipotesis *Aux;
    Aux=Hermano;
    unsigned char Estado=1;
    for(int i=0; i<Papa->RetornaHijos(); i++)
    {
        if(Aux->Valor==2)
        {
        }
        else
        {
            Estado=0;
        }
        Aux=Aux->Hermano;
    }
    return Estado;
}

void Hipotesis::QuitaAgotamiento(void)//quita el agotamiento de los nodos por la relación "es similar a"
{
    Hipotesis *Aux;
    Aux=Hermano;
    for(int i=0; i<Papa->RetornaHijos();i++)
        Aux->Valor=-1;
}

struct Elemento//para uso del examen
{
    Hipotesis *HipotesisconIncertidumbre;
    Elemento *Siguiente;
};

//Aqui hay que definir las variable globales y estáticas del STI
static Hipotesis *ObjetivoPedagogico=NULL;
static short EstadoTutor=NORMAL;//al inicio de una sesión el estado del STI es NORMAL
static int IndiceGlobal;

//Hijos, Identificador,lecciones,indicadores
//el Modelo del Estudiante, solo los conceptos y sus características de numero de
//nodos por la relación elemental para conocer un numero que lo identifica
//dentro del ME, el numero de lecciones que le corresponde e Indicadores

static Hipotesis Red[25]=
{
    Hipotesis(0,0,1,1),
    Hipotesis(0,1,2,3),
    Hipotesis(0,2,3,2),
    Hipotesis(0,3,1,1),
    Hipotesis(4,4,1,1),
    Hipotesis(1,5,1,1),
    Hipotesis(0,6,3,1),
    Hipotesis(0,7,3,1),
    Hipotesis(0,8,3,1)//las centrales generadora
    Hipotesis(0,9,3,1),
    Hipotesis(0,10,3,1),

```

# Apéndice A

```
Hipotesis(0,11,3,1)//las centrales generadora
Hipotesis(6,12,3,1)//las centrales generadoras (la Hipotesis)
Hipotesis(2,13,1,1)//Sistema de generación
Hipotesis(3,14,3,3)//SEE
Hipotesis(0,15,1,1),
Hipotesis(0,16,1,1),
Hipotesis(0,17,1,1),
Hipotesis(3,18,2,1)//Sistema de Transformación
Hipotesis(0,19,1,1),
Hipotesis(0,20,1,2),
Hipotesis(0,21,1,1),
Hipotesis(3,22,2,1)//Sistema de Transmisión
Hipotesis(3,23,2,2),
Hipotesis(0,24,2,2));//Carga demandada

//Red Semántica
```

```
//para el uso del examen
```

```
static Elemento *ListaconIncertidumbre=NULL;
int i;
static int NumeroHipotesisconIncertidumbre=0;
static short ValorAuxiliar;

extern "C"
void __declspec(dllexport) Iniciacion(void)
{
    //Esta función arma las relaciones de cada concepto
    //El método es largo pero necesario, ya que cada red es única
    Hipotesis *Aux=NULL;

    //Para iniciar una "semilla" para la generación de números aleatorios
    srand( (unsigned)time( NULL ) );
    rand();

    //Concepto 0
    Aux=&Red[0];//Apuntamos a uno de los elementos
    Aux->PonPapa(&Red[4]);//uno de sus miembro a Papa
    Aux->PonHijo(NULL);//Uno de sus miembros a uno de sus hijo (el más importante)
    Aux->PonHermano(&Red[1]);//Uno de sus miembros a su hermano

    //Concepto 1
    Aux=&Red[1];
    Aux->PonPapa(&Red[4]);
    Aux->PonHijo(NULL);
    Aux->PonHermano(&Red[2]);

    //Concepto 2
    Aux=&Red[2];
    Aux->PonPapa(&Red[4]);
    Aux->PonHijo(NULL);
    Aux->PonHermano(&Red[3]);

    //Concepto 3
    Aux=&Red[3];
    Aux->PonPapa(&Red[4]);
    Aux->PonHijo(NULL);
    Aux->PonHermano(&Red[0]);

    //Concepto 4
    Aux=&Red[4];
    Aux->PonPapa(&Red[5]);
    Aux->PonHijo(&Red[0]);
    Aux->PonHermano(&Red[4]);

    //Concepto 5
    Aux=&Red[5];
    Aux->PonPapa(&Red[13]);
    Aux->PonHijo(&Red[4]);
    Aux->PonHermano(&Red[12]);
```

# Apéndice A

```
//Concepto 6
Aux=&Red[6];
Aux->PonPapa(&Red[12]);
Aux->PonHijo(NULL);
Aux->PonHermano(&Red[7]);
```

```
//Concepto 7
Aux=&Red[7];
Aux->PonPapa(&Red[12]);
Aux->PonHijo(NULL);
Aux->PonHermano(&Red[8]);
```

```
//Concepto 8
Aux=&Red[8];
Aux->PonPapa(&Red[12]);
Aux->PonHijo(NULL);
Aux->PonHermano(&Red[9]);
```

```
//Concepto 9
Aux=&Red[9];
Aux->PonPapa(&Red[12]);
Aux->PonHijo(NULL);
Aux->PonHermano(&Red[10]);
```

```
//Concepto 10
Aux=&Red[10];
Aux->PonPapa(&Red[12]);
Aux->PonHijo(NULL);
Aux->PonHermano(&Red[11]);
```

```
//Concepto 11
Aux=&Red[11];
Aux->PonPapa(&Red[12]);
Aux->PonHijo(NULL);
Aux->PonHermano(&Red[6]);
```

```
//Concepto 12
Aux=&Red[12];
Aux->PonPapa(&Red[13]);
Aux->PonHijo(&Red[6]);
Aux->PonHermano(&Red[5]);
```

```
//Concepto 13
Aux=&Red[13];
Aux->PonPapa(&Red[14]);
Aux->PonHijo(&Red[5]);
Aux->PonHermano(&Red[23]);
```

```
//Concepto 14
Aux=&Red[14];
Aux->PonPapa(NULL);
Aux->PonHijo(&Red[13]);
Aux->PonHermano(&Red[14]);
```

```
//Concepto 15
Aux=&Red[15];
Aux->PonPapa(&Red[18]);
Aux->PonHijo(NULL);
Aux->PonHermano(&Red[16]);
```

```
//Concepto 16
Aux=&Red[16];
Aux->PonPapa(&Red[18]);
Aux->PonHijo(NULL);
Aux->PonHermano(&Red[17]);
```

```
//Concepto 17
Aux=&Red[17];
Aux->PonPapa(&Red[18]);
```

# Apéndice A

```
Aux->PonHijo(NULL);  
Aux->PonHermano(&Red[15]);
```

```
//Concepto 18  
Aux=&Red[18];  
Aux->PonPapa(&Red[23]);  
Aux->PonHijo(&Red[15]);  
Aux->PonHermano(&Red[22]);
```

```
//Concepto 19  
Aux=&Red[19];  
Aux->PonPapa(&Red[22]);  
Aux->PonHijo(NULL);  
Aux->PonHermano(&Red[20]);
```

```
//Concepto 20  
Aux=&Red[20];  
Aux->PonPapa(&Red[22]);  
Aux->PonHijo(NULL);  
Aux->PonHermano(&Red[21]);
```

```
//Concepto 21  
Aux=&Red[21];  
Aux->PonPapa(&Red[22]);  
Aux->PonHijo(NULL);  
Aux->PonHermano(&Red[19]);
```

```
//Concepto 22  
Aux=&Red[22];  
Aux->PonPapa(&Red[23]);  
Aux->PonHijo(&Red[19]);  
Aux->PonHermano(&Red[18]);
```

```
//Concepto 23  
Aux=&Red[23];  
Aux->PonPapa(&Red[14]);  
Aux->PonHijo(&Red[22]);  
Aux->PonHermano(&Red[24]);
```

```
//Concepto 24  
Aux=&Red[24];  
Aux->PonPapa(&Red[14]);  
Aux->PonHijo(NULL);  
Aux->PonHermano(&Red[13]);
```

```
//relaciona los Indices de los indicadores que van a ser usados por Authorware para cada concepto
```

```
Red[0].PonInd(1,4);  
Red[1].PonInd(1,7);  
Red[1].PonInd(2,6);  
Red[1].PonInd(3,5);  
Red[2].PonInd(1,9);  
Red[2].PonInd(2,8);  
Red[3].PonInd(1,10);  
Red[4].PonInd(1,12);  
Red[5].PonInd(1,13);  
Red[6].PonInd(1,28);  
Red[7].PonInd(1,29);  
Red[8].PonInd(1,30);  
Red[9].PonInd(1,33);  
Red[10].PonInd(1,31);  
Red[11].PonInd(1,32);  
Red[12].PonInd(1,27);  
Red[13].PonInd(1,26);  
Red[14].PonInd(1,3);  
Red[14].PonInd(2,2);  
Red[14].PonInd(3,1);  
Red[15].PonInd(1,23);  
Red[16].PonInd(1,22);  
Red[17].PonInd(1,24);  
Red[18].PonInd(1,21);
```

# Apéndice A

```
Red[19].PonInd(1,19);  
Red[20].PonInd(1,17);  
Red[20].PonInd(2,18);  
Red[21].PonInd(1,20);  
Red[22].PonInd(1,16);  
Red[23].PonInd(1,14);  
Red[23].PonInd(2,15);  
Red[24].PonInd(1,25);  
Red[24].PonInd(2,3);
```

//relaciona los indices de las lecciones que van a ser usados por Authorware para cada concepto

```
Red[0].PonLeccion(1,28);  
Red[1].PonLeccion(1,30);  
Red[1].PonLeccion(2,29);  
Red[2].PonLeccion(1,31);  
Red[2].PonLeccion(2,32);  
Red[2].PonLeccion(3,33);  
Red[3].PonLeccion(1,34);  
Red[4].PonLeccion(1,35);  
Red[5].PonLeccion(1,36);  
Red[6].PonLeccion(1,5);  
Red[6].PonLeccion(2,7);  
Red[6].PonLeccion(3,6);  
Red[7].PonLeccion(1,8);  
Red[7].PonLeccion(2,10);  
Red[7].PonLeccion(3,9);  
Red[8].PonLeccion(1,11);  
Red[8].PonLeccion(2,13);  
Red[8].PonLeccion(3,12);  
Red[9].PonLeccion(1,14);  
Red[9].PonLeccion(2,16);  
Red[9].PonLeccion(3,15);  
Red[10].PonLeccion(1,17);  
Red[10].PonLeccion(2,19);  
Red[10].PonLeccion(3,18);  
Red[11].PonLeccion(1,22);  
Red[11].PonLeccion(2,21);  
Red[11].PonLeccion(3,20);  
Red[12].PonLeccion(1,14);  
Red[12].PonLeccion(2,9);  
Red[12].PonLeccion(3,6);  
Red[13].PonLeccion(1,4);  
Red[14].PonLeccion(1,3);  
Red[14].PonLeccion(2,1);  
Red[14].PonLeccion(3,2);  
Red[15].PonLeccion(1,41);  
Red[16].PonLeccion(1,40);  
Red[17].PonLeccion(1,42);  
Red[18].PonLeccion(1,25);  
Red[18].PonLeccion(2,23);  
Red[19].PonLeccion(1,38);  
Red[20].PonLeccion(1,37);  
Red[21].PonLeccion(1,39);  
Red[22].PonLeccion(1,24);  
Red[22].PonLeccion(2,23);  
Red[23].PonLeccion(1,24);  
Red[23].PonLeccion(2,23);  
Red[24].PonLeccion(1,1);  
Red[24].PonLeccion(2,26);  
}
```

//las siguientes funciones son las que se exportan a modulo de STI implantado en Authorware

```
extern "C"  
void __declspec(dllexport) PropagaVerdadero(void)//solo cuando se examina  
{  
    ObjetivoPedagogico->PropagaVerdadero();  
    //Si contesto bien entonces en la red (ME) propagamos los valores de "SE ASUME CONOCIDO"  
}
```



# Apéndice A

```
extern "C"
void __declspec(dllexport) PropagaFalso(void)//solo cuando se examina
{
    ObjetivoPedagogico->PropagaFalso();
    //Si contesta mal entonces propagamos el valor de "NO HAY INFO"
}

extern "C"
short __declspec(dllexport) GeneraObjetivo(void)
{
    ObjetivoPedagogico=Red[14].GeneraObjetivo();//desde el concepto más complejo
    if(ObjetivoPedagogico==NULL)//Si no se encontró objetivo entonces el estudiante
        //finalizo con éxito su instrucción
        return FINTUTOR;
    else if(ObjetivoPedagogico->Valor==0)//si el objetivo encontrado tiene el Valor NOHAYINFO
        //entonces se diagnostica
        return DIAGNOSTICA;
    else
        return ENSENA; //si el objetivo encontrado tiene el Valor NOCONOCE
        //entonces se enseña
}

extern "C"
void __declspec(dllexport) ModelaEstudiante(void)
{
    ObjetivoPedagogico->ModelaEstudiante();//El Objeto checa la red para buscar que es lo que
    //hace que el estudiante no aprenda
}

extern "C"
void __declspec(dllexport) MarcaHipotesis(int value)
{
    ObjetivoPedagogico->PonValor(value);
    //value == 2 El diagnostico nos dice que el concepto lo conoce el estudiante
    //value == 0 La agenda enseñó un concepto, y hay que diagnosticarlo
    //value == 1 || value == -1 No se aplica, la red marca automáticamente cuando
    //un concepto se asume conocido
    //o no hay información
}

extern "C"
void __declspec(dllexport) IniciaHipotesis(int Index, int Value)//para el uso del Módulo Tutor (administración)
    //inicia el ME con los valores que posee la BD
{
    Value=(char)Value;
    if(Index<0) return;
    else if(Index>NUMERODEHIPOTESIS) return;
    else
    {
        Red[Index].PonValor(Value);
        return;
    }
}

extern "C"
short __declspec(dllexport) RetomaLeccion(void)//retorna el índice correspondiente de la lección
    //para que sea usada por el modulo construido con Authorware
{
    return ObjetivoPedagogico->RetomaLeccion();
}

extern "C"
short __declspec(dllexport) RetornaInd(void) //retorna el índice correspondiente de la pregunta
    //para que sea usada por el modulo construido con Authorware
{
```

# Apéndice A

```
    return ObjetivoPedagogico->RetornaInd(); //Busca la pregunta adecuada para el
}
    //Diagnostico o examen

extern "C"
short __declspec(dllexport) RetornaEstado(void) //retorna el Valor de un concepto
{
    return ObjetivoPedagogico->RetornaEstado();
}

extern "C"
short __declspec(dllexport) ObtieneEstadoTutor(void)
{
    return EstadoTutor; //retorna el estado del STI [MODELA,NORMAL]
}

extern "C"
void __declspec(dllexport) AjusteA(void) //invoca al ajuste del ME
{
    Red[14].Ajuste(); //el ajuste se inicia desde el concepto más complejo o abstracto
}

//la funcione siguiente es para el uso del Módulo Tutor(administración)
//retorna el valor del concepto respecto a su índice con el fin de que sea guardado en la BD
extern "C"
short __declspec(dllexport) RetornaValorHip(short Indice)
{
    return aux=Red[Indice].RetornaValorHip();
}

//las siguientes funciones sirven para realizar el examen inicial
//sus especificaciones y diseño se encuentra en el capítulo de Arquitectura

//esta función genera una lista con los conceptos que poseen incertidumbre
//con el objeto de su posible examinación
extern "C"
short __declspec(dllexport) GeneraListaconIncertidumbre(void)
{
    Elemento *NuevoElemento;
    Elemento *FinLista=NULL;
    NumeroHipotesisconIncertidumbre=0;
    ValorAuxiliar=0;
    for(i=0; i<NUMERODEHIPOTESIS;i++)
    {
        if(Red[i].Valor==0)
        {
            NuevoElemento= new Elemento;
            NuevoElemento->HipotesisconIncertidumbre=&Red[i];
            NuevoElemento->Siguiete=NULL;
            NumeroHipotesisconIncertidumbre++;
            if (FinLista==NULL)//Lista vacia
            {
                FinLista=NuevoElemento;
                ListaconIncertidumbre=NuevoElemento;
            }
            else
            {
                FinLista->Siguiete=NuevoElemento;
                FinLista=NuevoElemento;
            }
        }
    }
    ValorAuxiliar=NumeroHipotesisconIncertidumbre;
    if(NumeroHipotesisconIncertidumbre==0)
    {
        EstadoTutor=NORMAL;
    }
}
```

# Apéndice A

```
        ListaconIncertidumbre=NULL;
    }
    else
    {
        EstadoTutor=EXAMINA;
    }
    return NumeroHipotesisconIncertidumbre;
}

//esta función genera el examen inicial
extern "C"
short __declspec(dllexport) Examen(void)
{
    unsigned char pregunta;
    short indice;
    Elemento *Aux;
    Elemento *Busca;
    if(ListaconIncertidumbre==NULL)
    {
        //No podemos hacer un examen puesto que no tenemos incertidumbre
        return NOHAYEXAMEN;
    }
    else if(NumeroHipotesisconIncertidumbre<10) //si hay menos de 10 conceptos entonces todos se examinan
    {

        ObjetivoPedagogico=ListaconIncertidumbre->HipotesisconIncertidumbre;
        //Es nuestro objetivo pedagógico ahora
        pregunta=ObjetivoPedagogico->RetornaInd();
        //diagnostica con la primera Hipotesis
        ObjetivoPedagogico->Checado=1;
        //Marcamos el concepto como Checado (Evaluado)
        Aux=ListaconIncertidumbre;
        //aux sera borrado
        ListaconIncertidumbre=ListaconIncertidumbre->Siguiente;
        //no hay que perder la cabeza de la lista
        delete Aux;
        return pregunta;
    }
    else
    {
        indice=((int)(((float)rand()/(32767.00))*((float)(ValorAuxiliar))))-1;//generacion de un
//numero aleatorio en funcion del numero de conceptos que son candidatos a su examinación
        if (indice<1) indice=1;
        else if(indice>ValorAuxiliar) indice=ValorAuxiliar;
        else{}
        Aux=ListaconIncertidumbre;
        Busca=ListaconIncertidumbre;
        for(i=1;i<indice;i++)
        {
            Busca=Busca->Siguiente;
        }
        if(Busca==Aux)
        {

            ObjetivoPedagogico=Busca->HipotesisconIncertidumbre;
            //ponemos a esta Hipotesis como objetivo pedagógico
            pregunta=ObjetivoPedagogico->RetornaInd();
            //Buscamos la pregunta adecuada
            ListaconIncertidumbre=ListaconIncertidumbre->Siguiente;
            //como fue seleccionado el primer elemento, solo adelantamos la cabeza de la lista
            delete Aux;
            //eliminamos el primer elemento
        }
        else
        {
            Aux=Busca;
            //nos auxiliamos, este apuntador nos servirá para encadenar el elemento siguiente
            Busca=Busca->Siguiente;
            //este apuntador sirve para diagnosticar
        }
    }
}
```

# Apéndice A

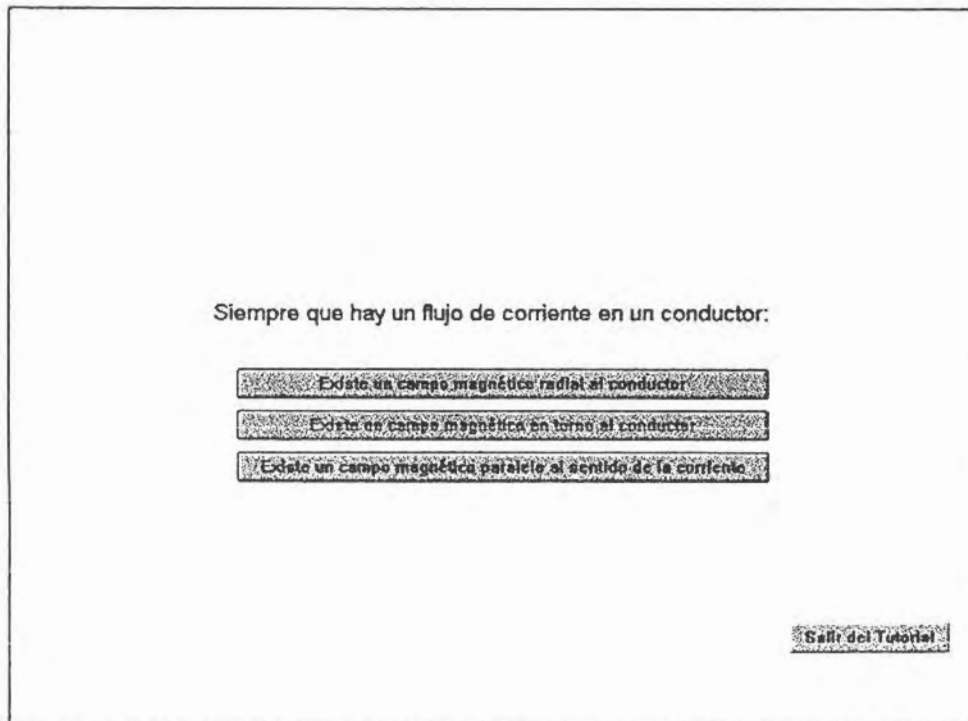
```
ObjetivoPedagogico=Busca->HipotesisconIncertidumbre;
//ponemos a esta Hipotesis como objetivo pedagógico
pregunta=ObjetivoPedagogico->RetornaInd();
//Buscamos la pregunta adecuada
Aux->Siguiente=Busca->Siguiente;
//encadenamos el elemento siguiente
delete Busca;
//borramos el elemento que utilizamos para diagnosticar
}
ValorAuxiliar--;
ObjetivoPedagogico->Checado=1;
return pregunta;
}
}

//elimina la lista restante de conceptos que fueron candidatos a su examinación
extern "C"
void __declspec(dllexport) BorraListaconIncertidumbre(void)
{
    Elemento *Aux;
    if(ListaconIncertidumbre==NULL)
        return;
    else
    {
        for(Aux=ListaconIncertidumbre;ListaconIncertidumbre!=NULL;)
        {
            ListaconIncertidumbre=ListaconIncertidumbre->Siguiente;
            delete Aux;
            Aux=ListaconIncertidumbre;
        }
    }
    EstadoTutor=NORMAL;
    return ;
}
```

## Apéndice B

Las siguientes pantallas son complementarias para la interfaz del prototipo de STI. Las primeras son parte de la interfaz para obtener información del estudiante, las siguientes son mensajes para el usuario, continuando con ejemplos de lecciones y finalizando con información complementaria del prototipo.

Con cada una de ellas se da una descripción breve de su objetivo



Siempre que hay un flujo de corriente en un conductor:

Existe un campo magnético radial al conductor.

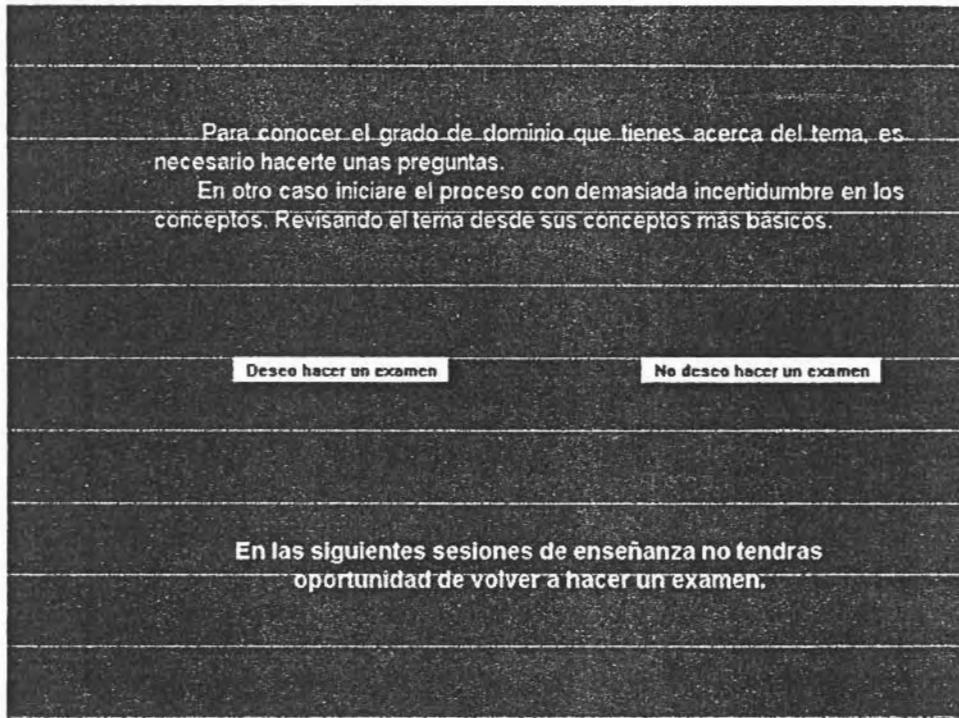
Existe un campo magnético en torno al conductor.

Existe un campo magnético paralelo al sentido de la corriente.

Salir del Tutorial

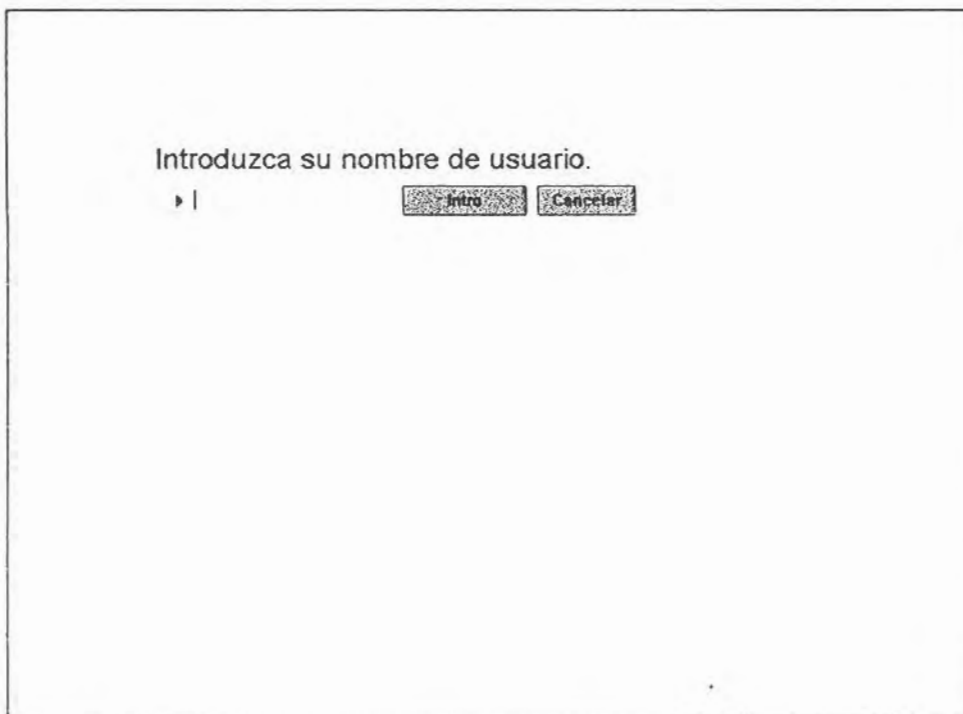
**Figura 56**

Interfaz para capturar la respuesta del estudiante a una pregunta.  
En esta interfaz también se permite la salida del estudiante del Sistema



**Figura 57**

Esta interfaz permite mencionar al estudiante las condiciones del examen y capturar su opción



**Figura 58**

Esta interfaz permite capturar el nombre de usuario del estudiante que tiene en el Sistema.

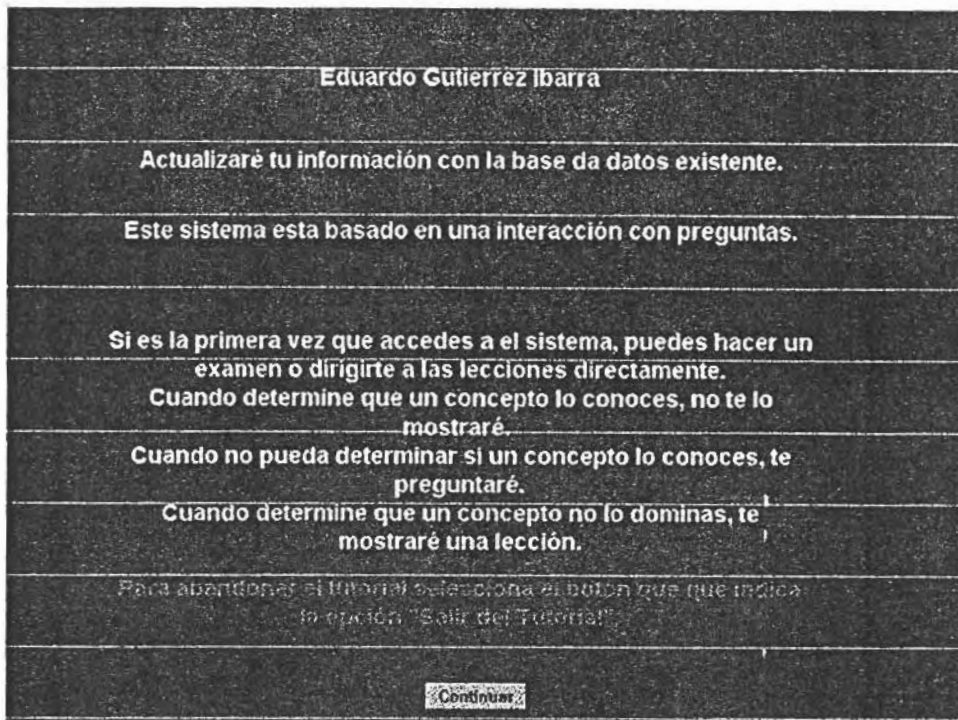


Figura 59

Esta pantalla muestra un mensaje al estudiante, informándole de las características del Sistema.

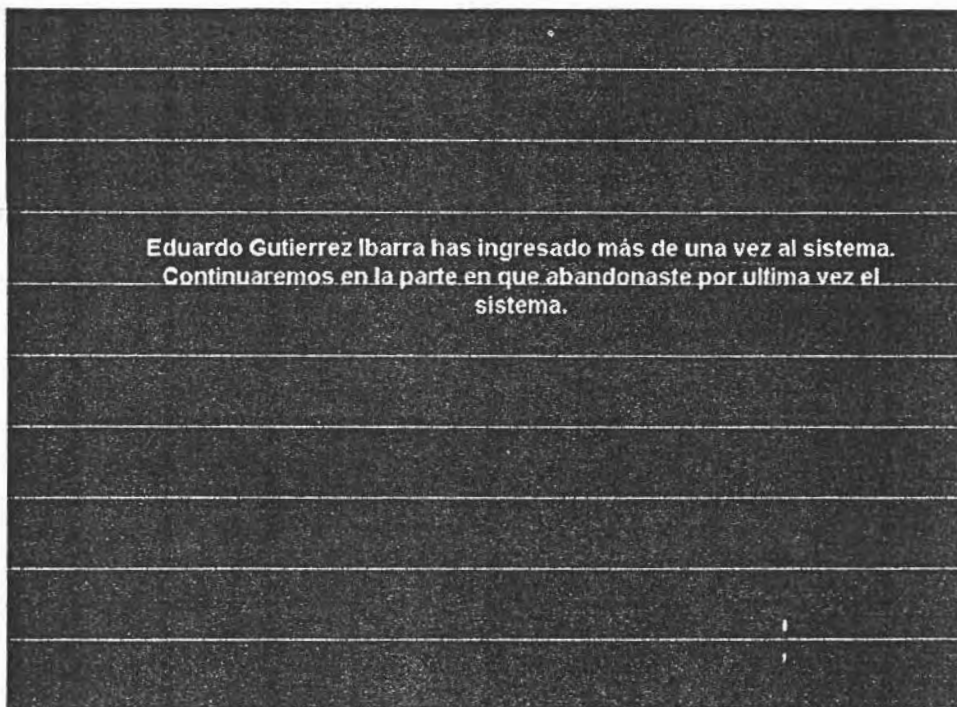
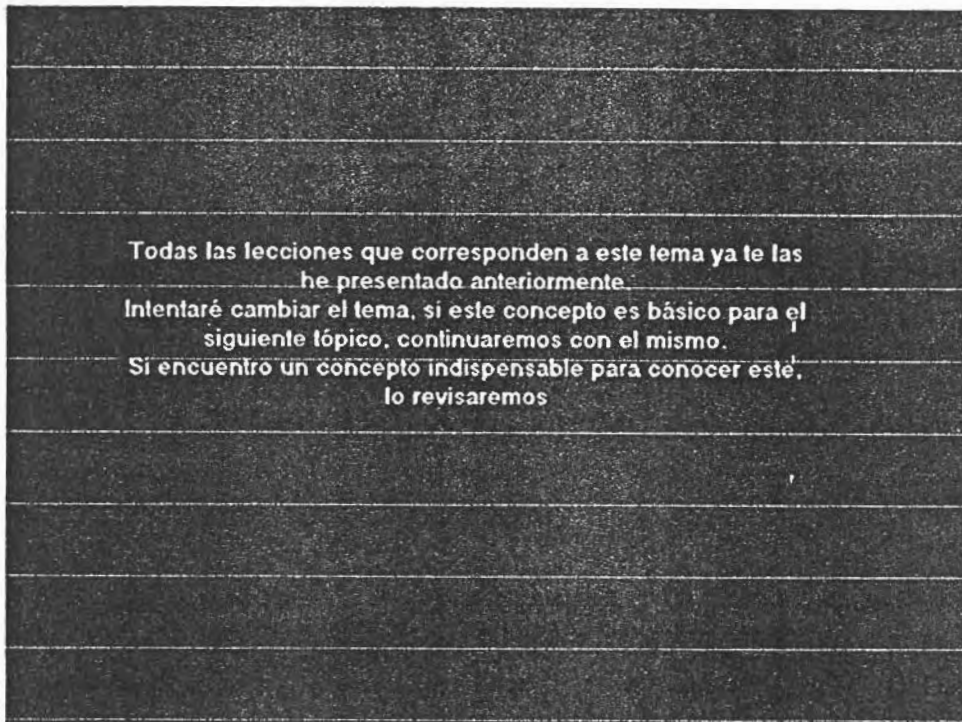


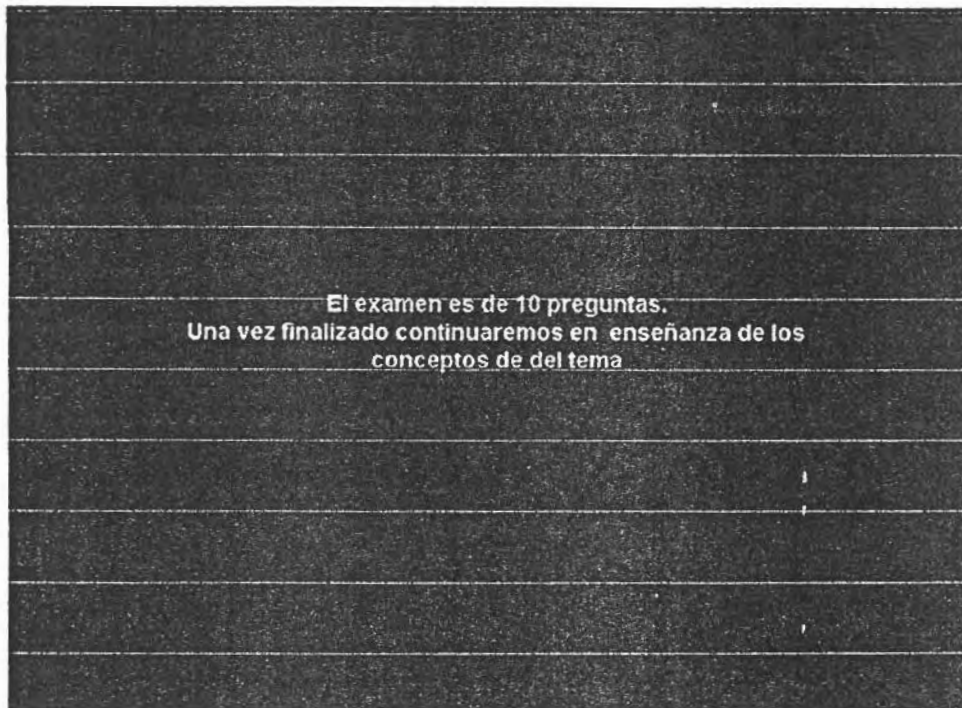
Figura 60

Esta pantalla es un mensaje que informa al estudiante que el STI ya le hizo un examen y por lo tanto lo reconoce y no es necesario repetir el examen.



**Figura 62**

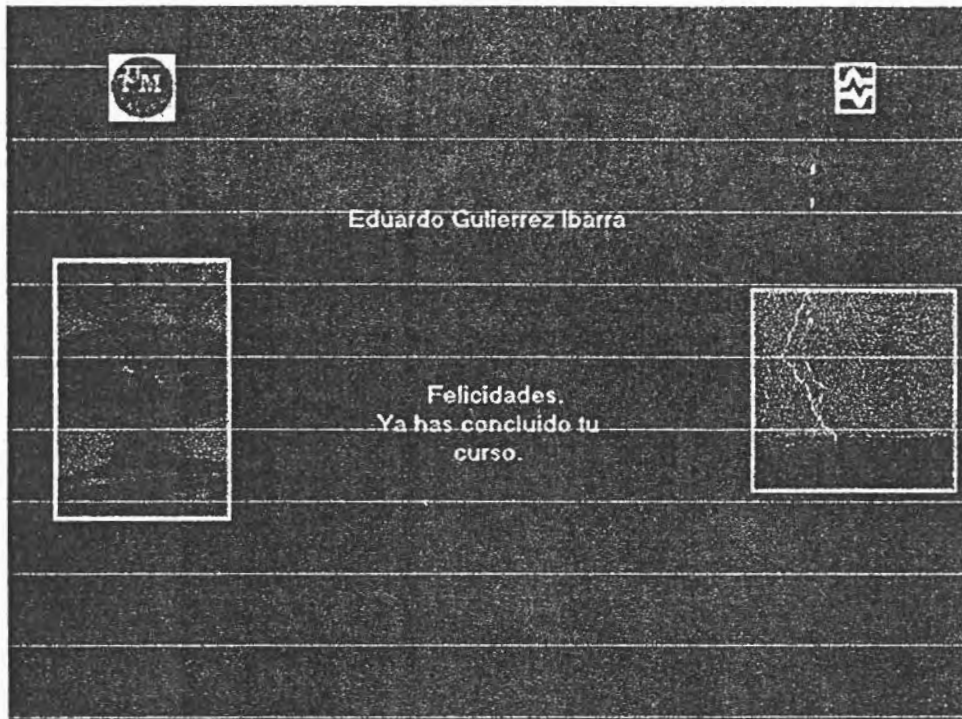
Esta pantalla muestra un mensaje al estudiante informándole que intentara cambiar de tema, dado que el concepto que se enseñaba esta agotado.



**Figura 61**

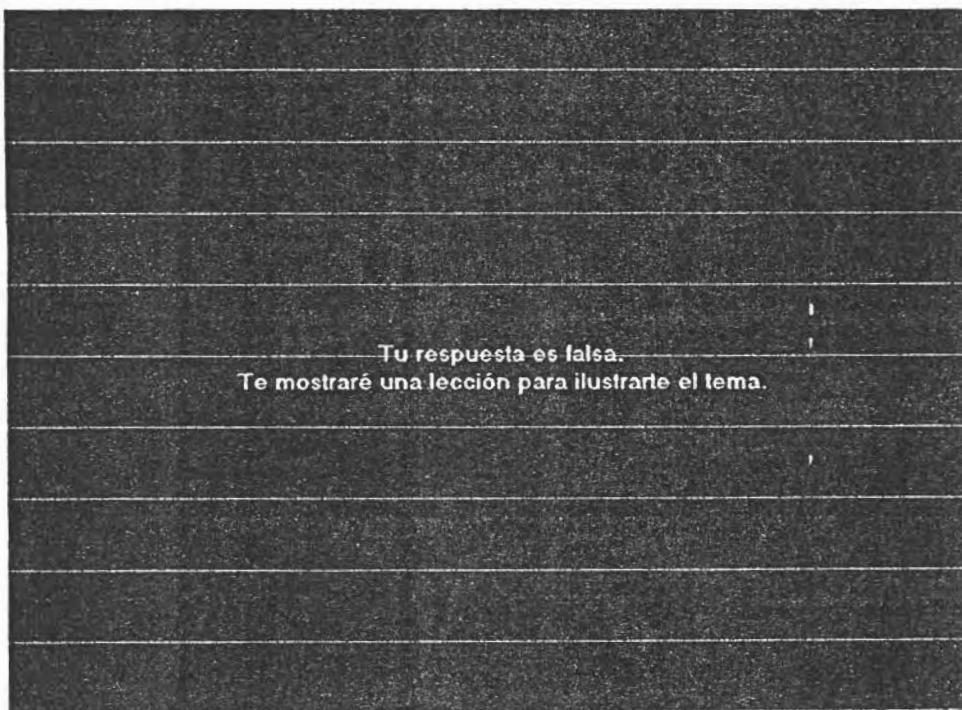
Esta pantalla muestra un mensaje al estudiante informándole el numero de preguntas del examen.





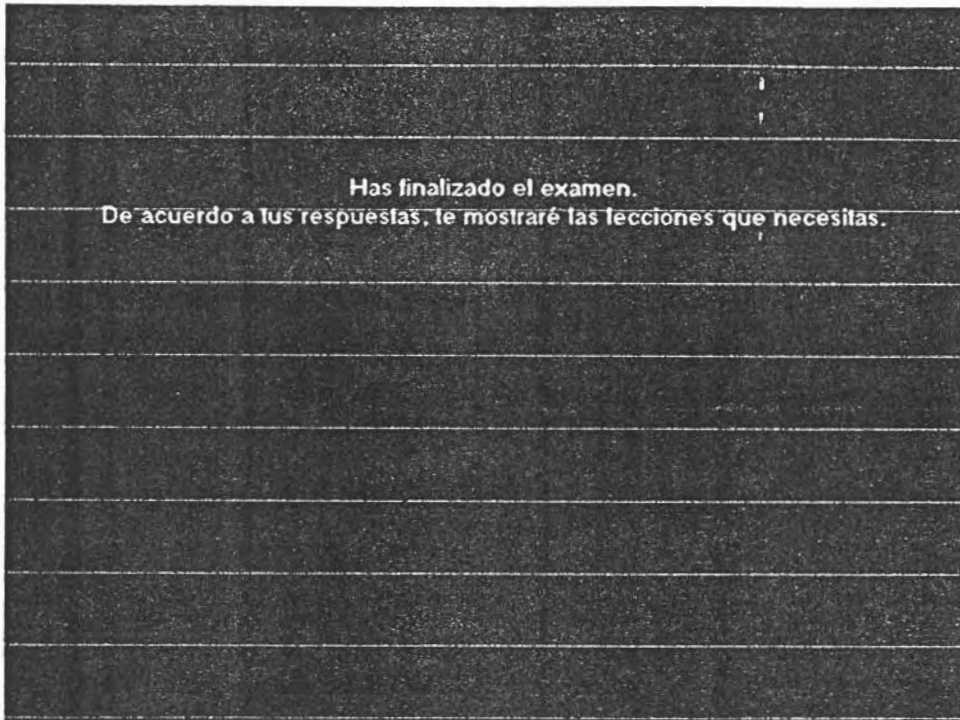
**Figura 63**

Esta pantalla muestra un mensaje al estudiante informándole que el curso ha concluido y lo felicita.



**Figura 64**

Esta pantalla muestra un mensaje al estudiante informándole que la respuesta proporcionada es falsa.



**Figura 65**

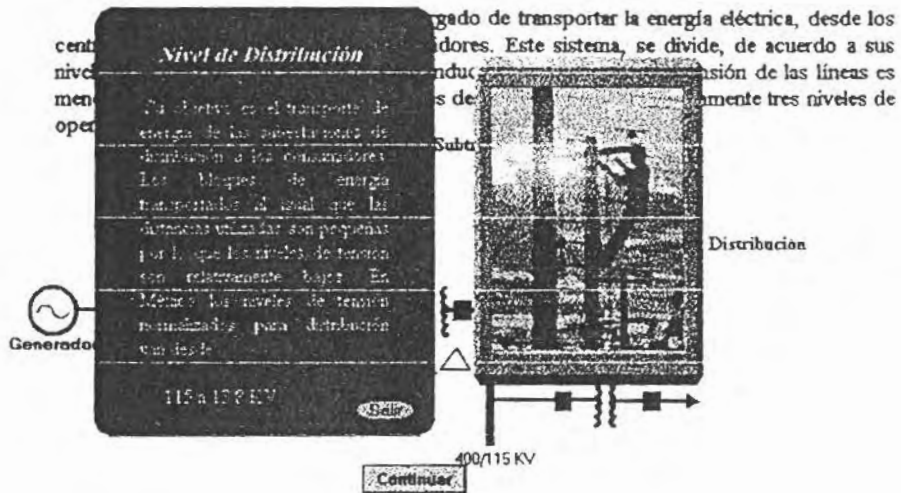
Esta pantalla muestra un mensaje al estudiante informándole que el examen inicial ha finalizado. Así si una nueva pregunta se avista, esta es por un diagnóstico.



**Figura 66**

Esta pantalla muestra un mensaje al estudiante informándole que la respuesta proporcionada fue correcta.

## Sistema de Transporte de Energía

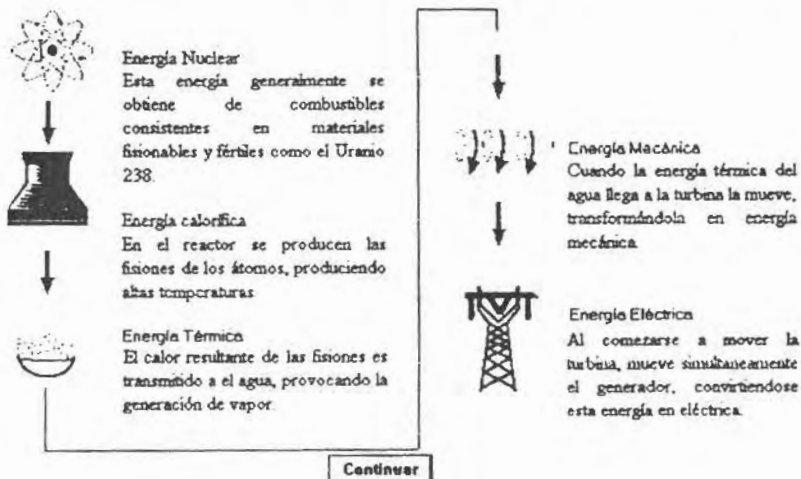


**Figura 67**

Esta pantalla muestra parte de las lecciones que son parte del concepto *Sistema de Transporte de Energía*.

## Transformaciones de Energía

Los procesos de transformación de energía que se realizan en las centrales nucleares, son los siguientes:



**Figura 68**

Esta pantalla muestra parte de las lecciones para el concepto *Central Nucleoeléctrica*.

## Generación del Voltaje

Tomando en cuenta los principios básicos como lo son la:

- Ley de Ampere.
- Ley de Faraday.
- Ley de Lenz.

Supóngase el siguiente sistema:

2 - Si se calienta el agua con tal energía, obtendremos como resultado energía térmica del agua

1 - El fuego contiene energía calorífica fácil de aprovechar.

4 - La energía cinética contenida en el vapor, se convertirá en mecánica moviendo la turbina

6 - Y si el imán comienza a girar dentro de una bobina, se producirá energía eléctrica.

3 - La energía térmica del agua se convertirá en energía cinética manifestada en vapor

5 - Si a la turbina se le acopla un imán

El proceso presentado, muestra de una manera simplificada cómo es que se logra dar movimiento al flujo magnético del imán, para que este pueda inducir voltaje en las espiras de la bobina.

Continuar

Repetir

Figura 69

Esta pantalla muestra parte de las lecciones que son para el concepto *Generación de Voltaje*.



Universidad Tecnológica de la Habana

"Cuerpo de Ingenieros, Liberes"  
Sigue escribiendo, inventando y mejorando



Prototipo de Sistema Tutorial Inteligente para la enseñanza

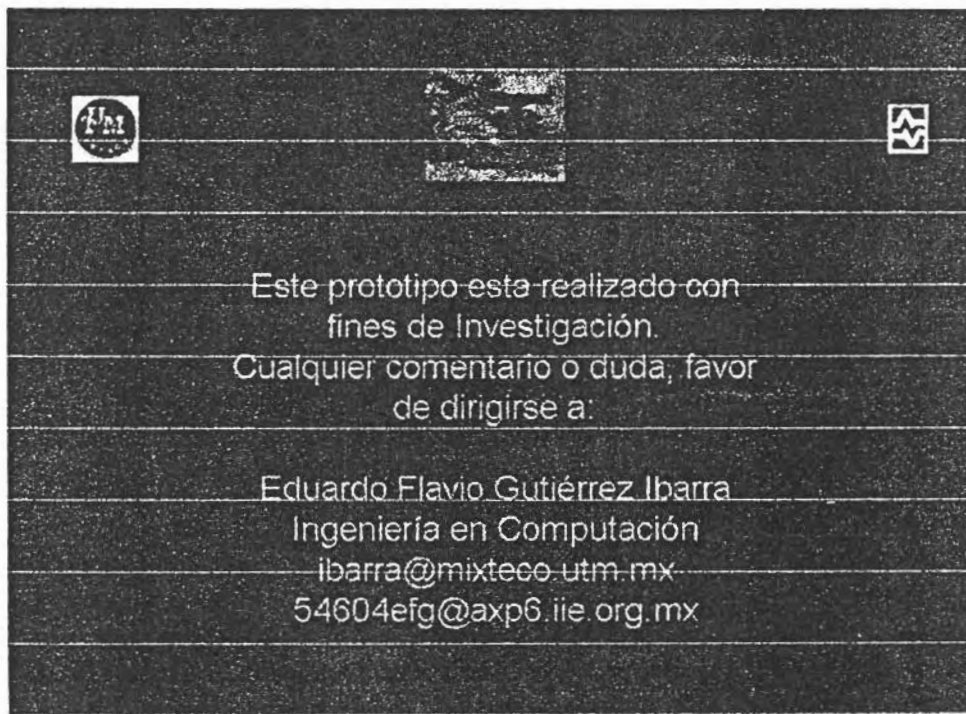
del Sistema Eléctrico de Energía

SIMSIE

Primavera 1998

Figura 70

Pantalla que muestra el inicio de una sesión con el Sistema



**Figura 71**

Pantalla de Información adicional del Sistema.

## Apéndice C

### Instalación del prototipo STI.EXE

#### Requerimientos:

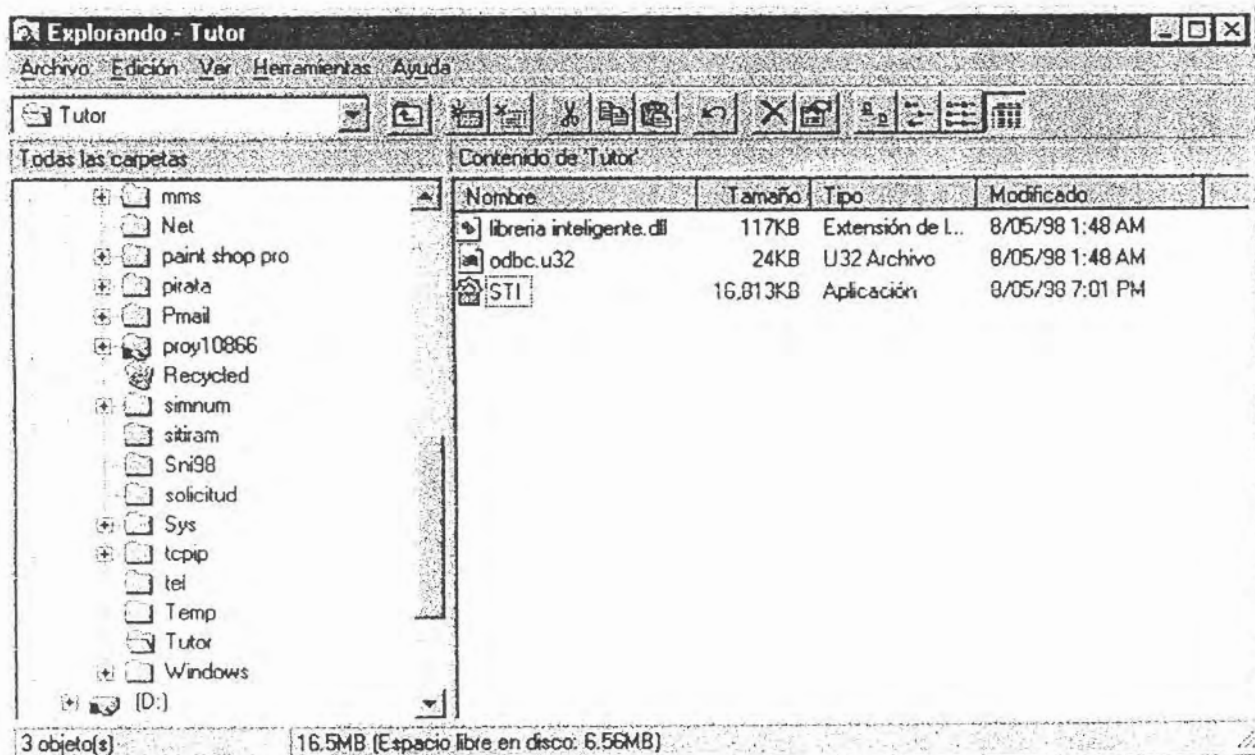
Sistema Operativo: Windows 95 o Windows NT  
Memoria Requerida: 8 MB de RAM  
Espacio en disco: 17 MB  
Programas adicionales: ODBC Administrador de 32 bits  
Hardware: Soporte para aplicaciones Multimedia preferentemente.

#### Instalación del prototipo *STI.EXE*

- Copiar los siguientes archivos a una carpeta (nueva, de preferencia) de su disco duro.

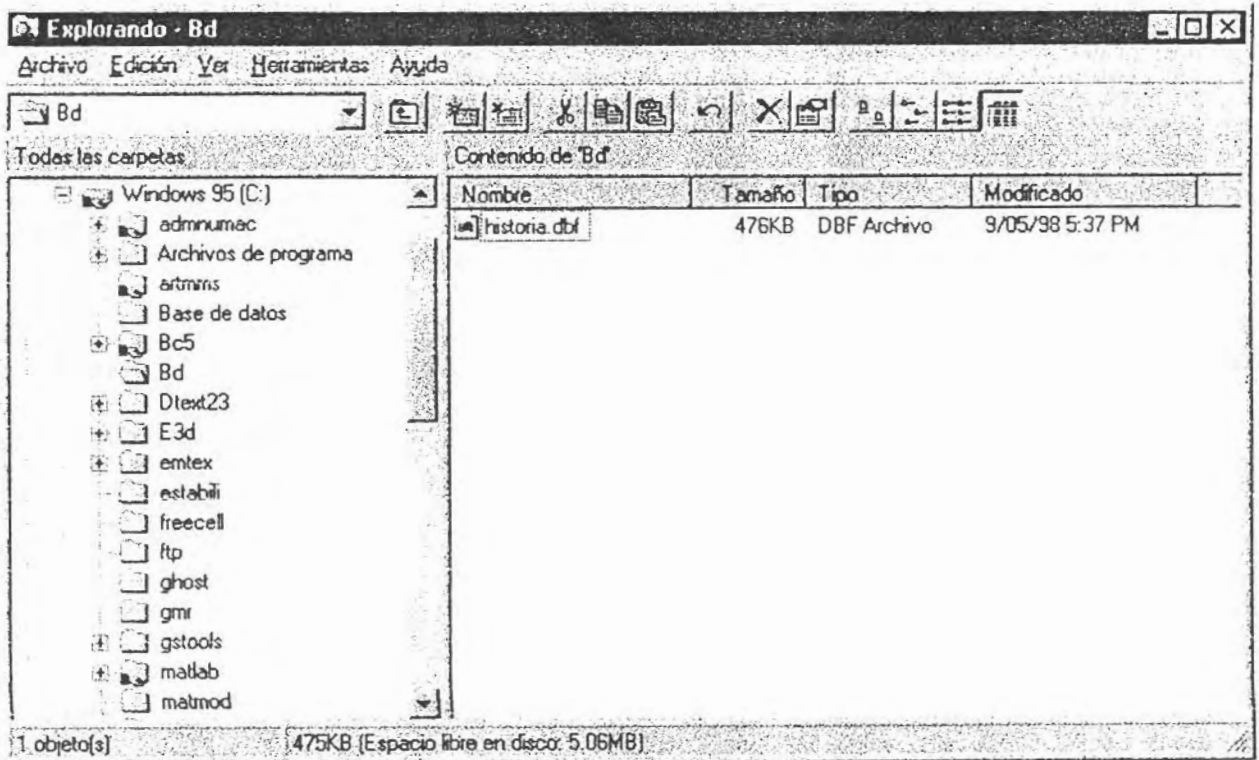
*Librería inteligente.dll*  
*STI.EXE*  
*odbc.U32*

el resultado debe ser como el de la siguiente figura



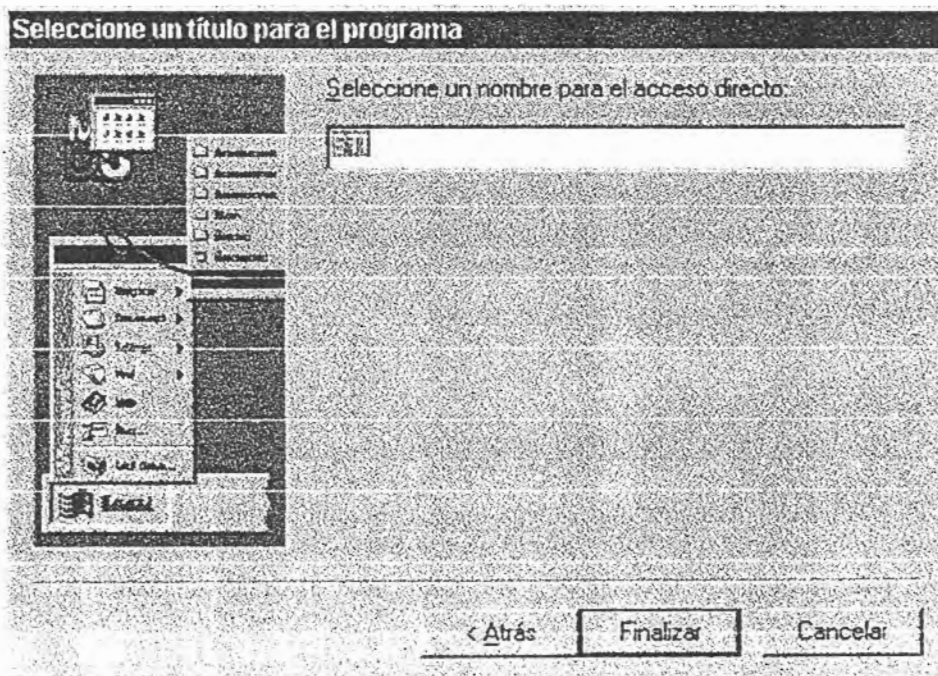
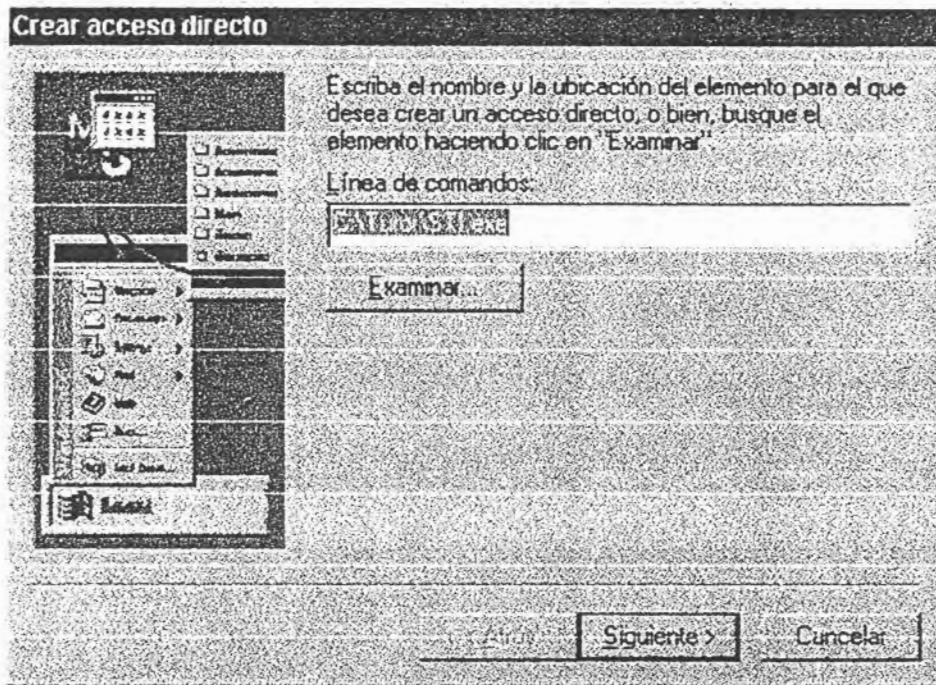
# Apéndice C

- Copiar al archivo *historia.dbf* a una carpeta (nueva de preferencia) de su disco duro. Este archivo en el prototipo solo cuenta con información de personas de la Universidad Tecnológica de la Mixteca y del Departamento de Simulación del Instituto de Investigaciones Eléctricas. Si desea aumentar el número de usuarios o usar una nueva tabla con usuarios diferente, sustituya el archivo correspondiente por el nuevo. El resultado debe ser como el de la siguiente figura.



# Apéndice C

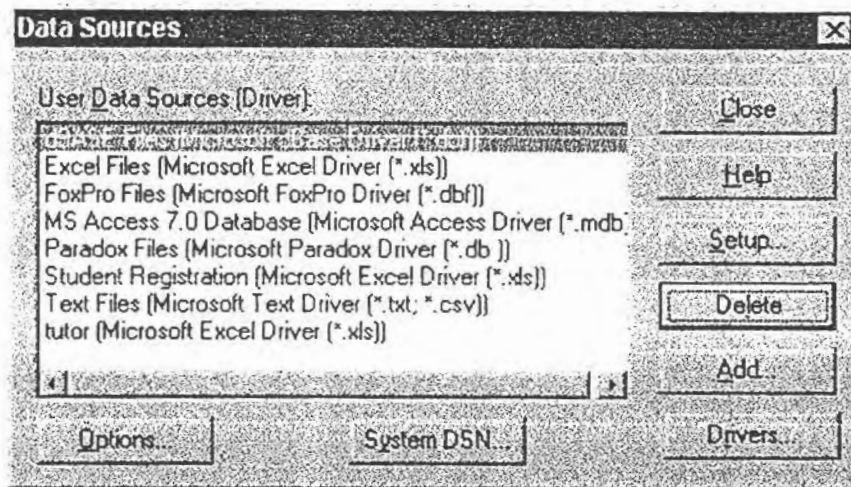
- Genere un acceso directo para *STI.EXE*. Las siguientes figuras muestran una manera de hacerlo



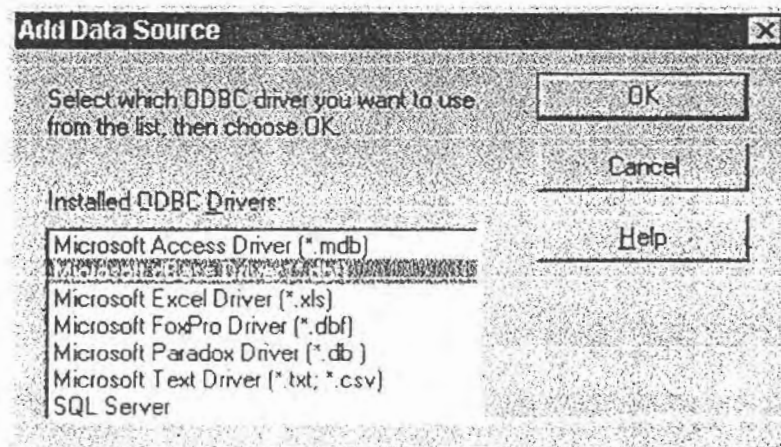


## Instalación de Origen de Datos para ODBC Administrator

- Ejecutar ODBC Administrator, se le presentará la siguiente interfaz.



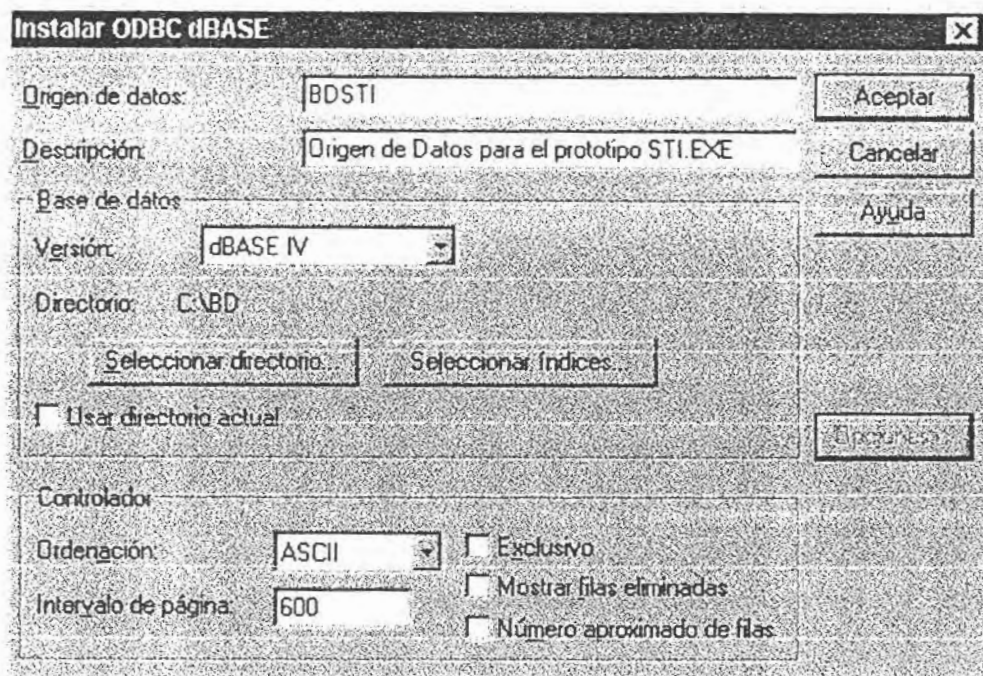
- Pulsar el botón para agregar (*Add* o *Agregar* dependiendo de la versión) un Origen de Datos, escoger un manejador para dBASE



## Apéndice C

- Nombrar al origen de Datos *BDSTI* , agregar una descripción al Origen de Datos, Asignar la versión de la Base de datos a dBASE IV y finalmente al Origen de Datos el directorio donde reside *historia.dbf*.

En las demás opciones usar las de default como se ve en la siguiente figura.



Una vez realizado los pasos anteriores, la instalación es completa. En caso de tener alguna duda, consultar el Manual de Windows 95 ,Windows NT o la ayuda de ODBC Administrator.

Note que las Interfaces de ODBC administrator pueden variar. Sin embargo su contenido es el mismo

## Apéndice D

### Manual de Usuario para STI.EXE

Los siguientes pasos indican la manera de usar el prototipo de STI *STI.exe*. Se requiere que además ya tenga instalado el prototipo en su PC, en caso contrario referirse a la *Guía de Instalación prototipo STI*.

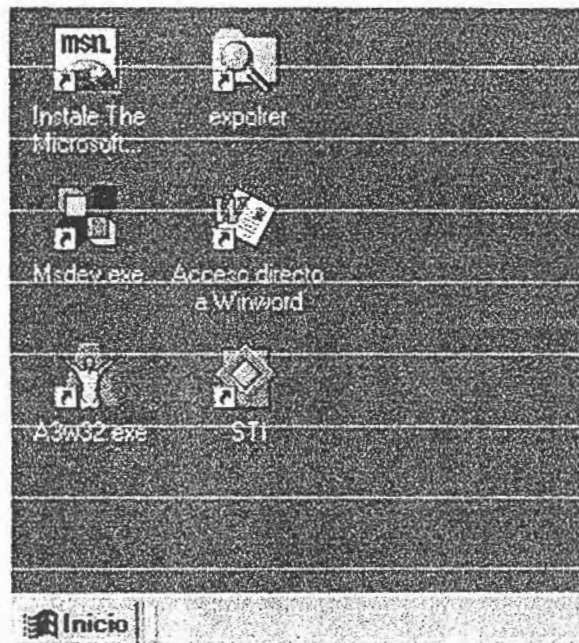
#### Ejecución del programa *STI.EXE*

Busque en su escritorio el icono de acceso directo a *STI* como el que se muestra a continuación.



STI

Y debe encontrarse en su escritorio como se muestra en la siguiente figura



De doble clic sobre el icono correspondiente.

# Apéndice D

Una vez que se ejecuta *STI.EXE* aparecerá la siguiente pantalla de presentación.



*Universidad Tecnológica de la Habana*

*"Labor et Sapientia, Libertas"*  
*Quis quibus, Tuus, sibi, Nullus.*



**Prototipo de Sistema Tutorial Inteligente para la  
enseñanza el Sistema Eléctrico de Energía**

**SITISIE**

Primavera 1998

Para continuar espere unos segundos, presione alguna tecla o un botón del ratón.

## Apéndice D

Después se presenta una interfaz para proporcionar el nombre de usuario. Tecleé su nombre de usuario, después teclee intro o el botón con el título *intro*. Si no desea continuar presione el botón con el título *cancelar*.

Introduzca su nombre de usuario.

▶ carolina|

Intro

Cancelar

Después el Sistema le pedirá su clave de acceso. Asegure de teclear correctamente su clave y de que no este activado el bloque de Mayúsculas. Su clave no aparecerá visible. Cuando termine presione intro.

Introduzca su clave de acceso.

▶ |

Si no es parte de los usuarios registrados o su clave fue mal introducida, el sistema le mostrará el siguiente mensaje.

Nombre de usuario Incorrecto

Si su clave es correcta el Sistema le mostrara la siguiente interfaz

En esta interfaz se explican las características de interacción del sistema..

Para continuar presione alguna tecla o un botón del ratón una vez que el botón con el titulo *continuar* aparezca.

### **Eduardo Gutierrez Ibarra**

**Actualizaré tu información con la base da datos existente.**

**Este sistema esta basado en una interacción con preguntas.**

**Si es la primera vez que accedes a el sistema, puedes hacer un examen o dirigirte a las lecciones directamente.**

**Cuando determine que un concepto lo conoces, no te lo mostraré.**

**Cuando no pueda determinar si un concepto lo conoces, te preguntaré.**

**Cuando determine que un concepto no lo dominas, te mostraré una lección.**

**Continuar**

## Apéndice D

Si es la primera vez que se presenta frente al sistema , se le mostrara la siguiente interfaz  
Aquí seleccione la opción de su preferencia. Note que si decide no hacer examen el sistema no tendrá información suficiente acerca de su dominio de conocimiento en el área, por lo cual se repasaran todas las lecciones.

Es recomendable realizar un examen previo.

Para conocer el grado de dominio que tienes acerca del tema, es necesario hacerte unas preguntas.

En otro caso iniciare el proceso con demasiada incertidumbre en los conceptos. Revisando el tema desde sus conceptos más básicos.

**Deseo hacer un examen**

**No deseo hacer un examen**

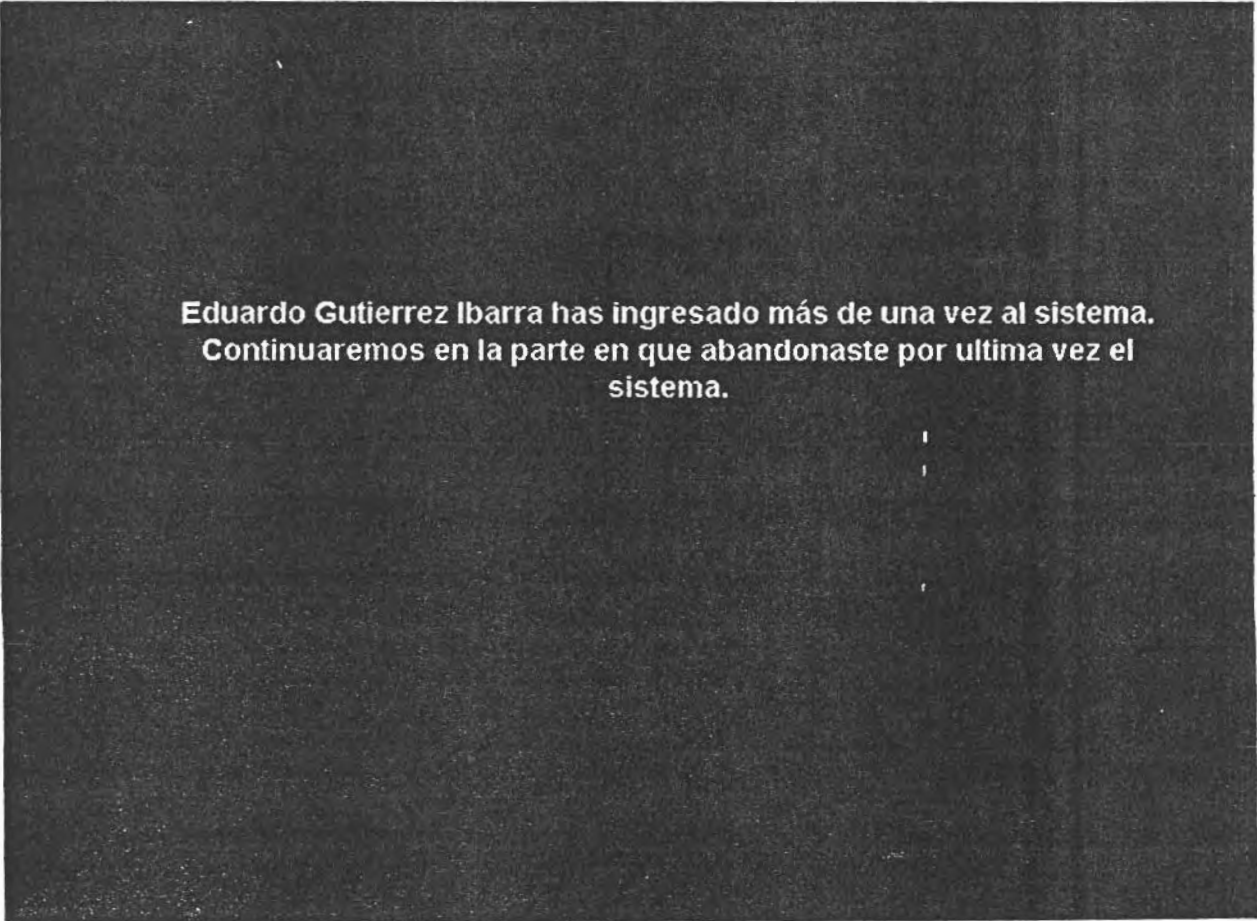
**En las siguientes sesiones de enseñanza no tendras oportunidad de volver a hacer un examen.**



## Apéndice D

Si no es la primera vez que ingresa al sistema, se le presentara la siguiente pantalla.

El sistema ya posee información del grado de dominio que tiene en el tema, por lo cual no es necesario realizar nuevamente un examen



**Eduardo Gutierrez Ibarra has ingresado más de una vez al sistema.  
Continuaremos en la parte en que abandonaste por ultima vez el sistema.**

Si es la primera vez que ingresa al sistema y decide hacer un examen. Conteste las diez (10) pregunta que le hace el sistema. Las preguntas presentan el siguiente formato, se describe en texto azul la pregunta y las posibles respuestas están en el texto que poseen los botones. De click en el botón de la respuesta que considere verdadera. La siguiente figura muestra la interfaz que hace las preguntas.

**Siempre que hay un flujo de corriente en un conductor:**

**Existe un campo magnético radial al conductor**

**Existe un campo magnético en torno al conductor**

**Existe un campo magnético paralelo al sentido de la corriente**

Una vez finalizado su examen se le presenta la siguiente interfaz.

**Has finalizado el examen.  
De acuerdo a tus respuestas, te mostraré las lecciones que necesitas.**

## Apéndice D

Una vez finalizado el examen y si no es su primera vez en el prototipo, puede pasar nuevamente a una pregunta si es que el sistema no conoce información de su dominio acerca del concepto que va a tratar. En otro caso se pasaran a las lecciones.

Para continuar después de cada lección debe esperar a que esta termine, si aparece el botón con el título *continuar* púselo con el ratón, si no aparece entonces pulse cualquier tecla o un botón del ratón.

Para abandonar el prototipo, espere a que se le presente una pregunta y presione con el ratón el botón con el título *Salir del Tutorial*. La siguiente figura muestra la interfaz correspondiente.

**Según la ley de Ampere, el campo magnético esta relacionado linealmente con la corriente que lo produce y/o**

**El número de espiras que contiene una bobina**

**El número de semiconductores asociados**

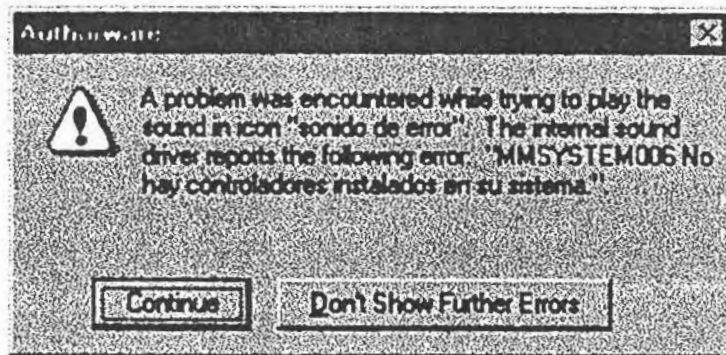
**El número de polos de un conductor**

**Salir del Tutorial**

Cada vez que de una respuesta correcta recibirá un mensaje y un sonido de campana de activará.  
Cada respuesta incorrecta recibirá un mensaje y un sonido de ganso se activará.

## Posible mensaje de error

Es posible que al durante el inicio de una sesión reciba el siguiente mensaje.



El error se debe a que no existen controladores de audio en su PC. Presione con el ratón el botón con el título *Don't Show Further Errors*.

Puede continuar trabajando con el Sistema sin ningún problema. No se reproducirá sonido únicamente