



# UNIVERSIDAD AUTÓNOMA METROPOLITANA

## UNIDAD AZCAPOTZALCO

### **MAESTRIA:**

CIENCIAS DE LA COMPUTACION

### **PROPUESTA DE TESIS:**

“DISEÑO DEL MODULO TUTOR PARA EL  
SISTEMA DE APRENDIZAJE INTELIGENTE  
DE LENGUAJE JAVA BÁSICO”

### **ASESORES DE TESIS:**

DRA. ANA LILIA LAUREANO C.  
M. en C. BLANCA SILVA L.

### **ALUMNA:**

MA. ESTHER PALMERÍN RUIZ

# ÍNDICE

<b>1.- INTRODUCCIÓN.</b> .....	<b>3</b>
<b>2.- ANTECEDENTES.</b> .....	<b>5</b>
<b>3.- ALCANCES</b> .....	<b>6</b>
<b>4.- JUSTIFICACIÓN</b> .....	<b>7</b>
<b>5.- OBJETIVOS:</b> .....	<b>7</b>
5.1 Objetivo General: .....	7
5.2 Objetivos Específicos: .....	7
<b>6.- NUESTRA PROPUESTA</b> .....	<b>7</b>
<b>7.- METODOLOGIA</b> .....	<b>12</b>
<b>8.-CALENDARIZACIÓN</b> .....	<b>12</b>
<b>9.- RECURSOS</b> .....	<b>13</b>
<b>10.- BIBLIOGRAFÍA</b> .....	<b>14</b>

## 1.- INTRODUCCIÓN.

### SISTEMAS TUTORES INTELIGENTES

Los Sistemas Tutores Inteligentes (STI) surgieron en la década de los 70 como una evolución de los Sistemas de Instrucción Asistida por Computador (CAI) en combinación con técnicas de la inteligencia artificial y de los métodos clásicos de enseñanza. El objetivo de los STI es proporcionar una mayor flexibilidad al lograr que éstos permitan una mejor interacción con el estudiante. El propósito es crear un ambiente educativo en el que el sistema tutorial se adapte a las necesidades individuales del usuario, (Laureano-Cruces, 2000 y Laureano-Cruces y de Arriaga, 2000).

El Sistema Tutor Inteligente deberá mantener un modelo del conocimiento del usuario para poder actuar con mayor sensibilidad ante el comportamiento de éste. Además, se hace necesario desarrollar interfaces reactivas, que no sólo permitan el diálogo entre el estudiante y el sistema sino que además potencien el aprendizaje constructivista permitiendo al alumno explorar y obtener sus propias conclusiones

Un STI captura el conocimiento de los expertos, en la enseñanza de un dominio con el fin de crear interacciones en forma dinámica, y personalizada.

### ARQUITECTURA Y FUNCIONAMIENTO

Un Sistema Tutor Inteligente está integrado por los siguientes componentes, mismos que se muestran en la Figura 1.

**Módulo experto:** En este componente se encuentra el conocimiento específico y detallado, obtenido de los expertos humanos que llevan años dedicándose a la solución experta del dominio, motivo de la enseñanza. Este módulo sirve para comparar la solución del estudiante con la de un experto.

**Modelo del estudiante:** Este módulo contiene la información de las características de las intervenciones del estudiante referentes a: tiempo que tarda en la solución, tipo de errores, número de errores, estilo de aprendizaje, temas vistos. Dicha información permite diagnosticar los efectos del proceso de enseñanza y es utilizada para elegir el siguiente tema a enseñar y la metodología o estrategias adecuadas. lo anterior con el fin de guiar el proceso de enseñanza.

**Módulo tutorial:** Este módulo de acuerdo con el modelo del estudiante decide qué enseñar, cómo enseñar y cuándo interrumpir. Lo anterior permite una enseñanza personalizada (tutorial) en este módulo se resuelven los problemas relacionados con el desarrollo del aprendizaje, el programa de contenidos y la forma de enseñar, (Laureano-Cruces-Cruces, Terán-Gilmore, de Arriaga y El Alami, 2003).

Este módulo debe tener las siguientes capacidades: (Laureano-Cruces y de Arriaga, 2001):

- Controlar el aprendizaje y su secuencia.
- Responder a preguntas hechas por el estudiante.
- Detectar el tipo y nivel de ayuda al estudiante.

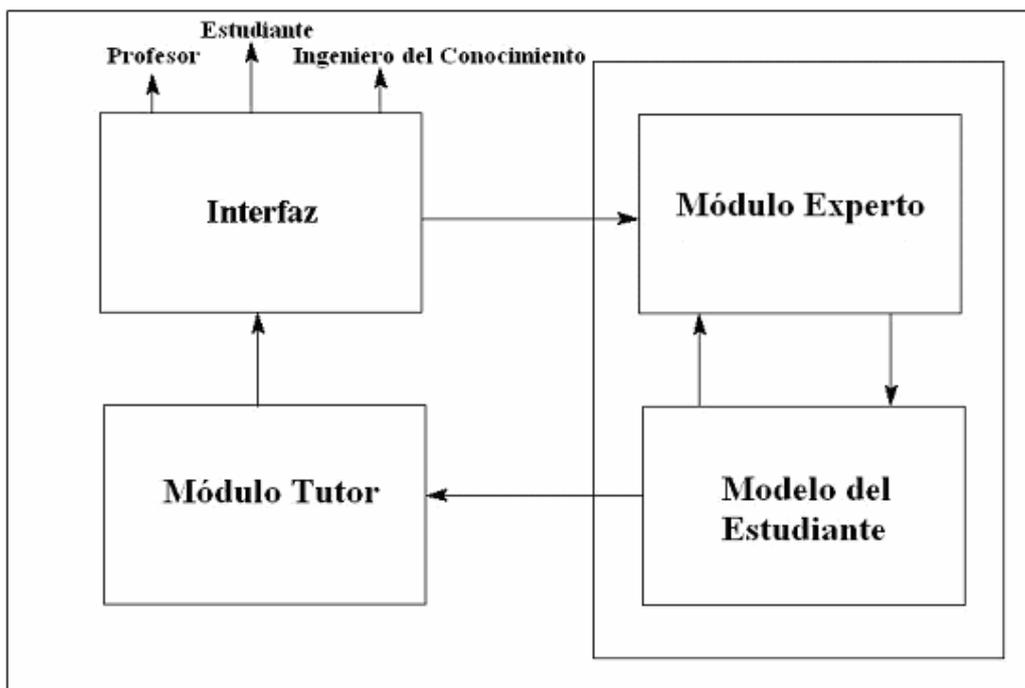
**Interfaz** cubre varias actividades en el funcionamiento global del STI:

Es un conjunto de canales de comunicación entre el estudiante y el sistema. Es el medio físico a través del cual se capta el desarrollo del estudiante, a través suyo el STI percibe el estado cognitivo y emotivo del usuario (Laureano-Cruces, 2004), (Jaques, 2004). Por otro lado tiene el papel de herramienta didáctica por lo que debe ser diseñada con cuidado y poniendo énfasis en la fidelidad del entorno de enseñanza que representa, de aquí que las interfaces deban ser flexibles, dotadas de multimedia, y de fácil acceso.

Con la información: 1) del desarrollo del usuario (clasificada en el modelo del estudiante), 2) la información del módulo del experto, el módulo tutorial obtiene un conjunto de inferencias que le permite dirigir el proceso de enseñanza – aprendizaje, con la suficiente adaptabilidad para implementar estrategias que garanticen un progreso en el proceso de enseñanza llevado a cabo entre el STI y el estudiante.

## **TECNOLOGÍAS EN EDUCACIÓN**

La creación de Sistemas Tutores Inteligentes adquiere significado cuando se piensa en tecnologías para la educación como mediadoras pedagógicas. Estos programas son necesarios si de verdad se quieren soluciones masivas apoyadas en tecnologías pero con calidad pedagógica.



**Figura 1. Arquitectura General de Sistemas Tutores Inteligentes**

## 2.- ANTECEDENTES.

Los sistemas expertos son un campo de aplicación de la inteligencia artificial a mediados de los años sesenta. En el período de nacimiento (1960), se creía que bastaban unas pocas leyes de razonamiento junto con potentes ordenadores para producir resultados brillantes. Un intento en ese sentido fue el llevado a cabo por los investigadores Alan Newell y Herbert Simon que desarrollaron un programa denominado GPS (General Problem Solver; solucionador general de problemas). Podía trabajar con criptoaritmética, con las torres de Hanoi y con otros problemas similares. Lo que no podía hacer el GPS era resolver problemas del mundo real, tales como un diagnóstico médico.

Los sistemas tutores inteligentes comenzaron a desarrollarse con la idea de impartir conocimiento con base en alguna forma de inteligencia para guiar al estudiante en el proceso de aprendizaje (Urretavizcaya, 2001; Sancho, 2002).

Algunos investigadores decidieron entonces cambiar por completo el enfoque del problema restringiendo su ambición a un dominio específico e intentando simular el razonamiento de un experto humano (Laureano y de Arriaga, 2000).

A partir de 1965, un equipo dirigido por Edward Feigenbaum, comenzó a desarrollar sistemas expertos utilizando bases de conocimiento definidas minuciosamente.

En 1967 se construye DENDRAL que se considera como el primer sistema experto. Se utilizaba para identificar estructuras químicas moleculares a partir de su análisis espectrográfico.

Entre 1970 y 1980 se desarrollaron sistemas con control basado en reglas a los cuales pertenecen: MYCIN para consulta y diagnóstico de infecciones de la sangre. Este sistema introdujo nuevas características: utilización de conocimiento impreciso para razonar y posibilidad de explicar el proceso de razonamiento. Lo más importante es que funcionaba de manera correcta, dando conclusiones análogas a las que un ser humano daría tras largos años de experiencia. En MYCIN aparecen claramente diferenciados motor de inferencia y base de conocimientos. Al separar esas dos partes, se puede considerar el motor de inferencias aisladamente. Esto da como resultado un sistema vacío o shell (concha). Así surgió EMYCIN (MYCIN Esencial) con el que se construyó SACON, utilizado para estructuras de ingeniería, PUFF para estudiar la función pulmonar.

Entre los **STI** desarrollados para los diversos campos del conocimiento se pueden destacar:

**Scholar:** (Carbonell y Collins, 1970) es el primer STI desarrollado para la enseñanza de geografía de Sudamérica, utiliza el diálogo socrático, a través de una práctica heurística que consiste en un descubrimiento del alumno de aquello que se le va a enseñar. (Wenger, 1987).

**Steamer:** fue construido para enseñar a los oficiales de la armada norteamericana problemas relativos a la dirección de una central de propulsión a vapor, siendo un ejemplo de interface gráfica que permite mover objetos.

**Guidon:** Clancey (1982) fue desarrollado para la reorganización de Mycin, sistema de diagnóstico médico especializado de enfermedades infecciosas.

**Wusor:** Goldstein (1982) desarrollado como un entrenador para jugar Wumpus, el juego de las cavernas, basado en inferencias. (Sleeman y Brown, 1982).

**Sophie:** (Brown y Burton, 1987) para simulación de circuitos electrónicos a través de generación de hipótesis. Detección de fallos en circuitos electrónicos.

**Proust:** (Soloway et al., 1987) desarrollado para programación en pascal.

**Buggy:** (Brown y Burton, 1987) es un sistema que utiliza la teoría de conceptos erróneos, que cuenta con una base de conocimiento, que incluye los posibles conceptos erróneos para comparar con los que los estudiantes podrían tener al estudiar la materia.

**Coach:** (Selker, 1994) Es un asesor inteligente, orientado a la enseñanza de lenguaje de programación. En este caso, se ofrece un ambiente de ayuda interactiva, donde el usuario proporciona una entrada (que se espera sea un programa en LISP); y un panel en donde, el asesor ofrece información relevante.

**Meno-Tutor:** Este es un Sistema Tutor Inteligente orientado a la enseñanza de la programación, específicamente de Pascal. Cuenta con un conjunto de plantillas de programas, que son soluciones plausibles a problemas que plantea; por ejemplo "realizar un programa que calcule el promedio de una lista de números" , es capaz de reconocer estos errores a través de un análisis del programa que proporciona el estudiante, y su comparación con la plantilla.

Entre algunos de los Sistemas Expertos genéricos se pueden enlistar los siguientes:

**XCON.-** Es un Sistema Experto para configuraciones de ordenadores desarrollado por la Digital Equipment Corporation.

Según los deseos individuales del cliente se configuran redes de ordenadores VAX:

Las respuestas a estas preguntas son muy detalladas. XCON es capaz de comprobar y completar los pedidos entrantes mucho más rápido y mejor que las personas encargadas hasta ahora de esa labor.

**ELIZA** Fue escrito a finales de los sesenta por el profesor Joseph Weizenbaum de Massachusetts Institute of Technology (MIT). El programa fue diseñado como una ayuda al análisis del lenguaje.

**DENDRAL** es un programa experto de análisis químico, soluciona un problema de ingeniería química en tres fases. En primer lugar, el programa infiere cualquier posible restricción sobre la solución basándose en el conocimiento que posee en su base de datos. A continuación permite a los usuarios añadir cualquier otro tipo de restricción y finalmente genera y comprueba una lista de posibles soluciones

**DIM:** sistema experto para el diagnóstico de fallas mecánicas en automóviles a partir de una interacción en tiempo real.

### 3.- ALCANCES

Diseñar el motor de inferencia del modulo tutor, para el lenguaje de programación Java Básico.

Para ello se recopilará la información asociada al proceso de enseñanza – aprendizaje del lenguaje de programación Java básico, y se realizará una investigación sobre los modelos pedagógicos que se pudieran adaptar para la enseñanza de este dominio.

#### **4.- JUSTIFICACIÓN**

En muchos países, la educación ha sido y esta siendo influenciada por la inclusión de la tecnología de la educación. Esto se debe, entre otras, cosas a: la optimización de recursos, la mejora de los procesos de enseñanza aprendizaje, la educación más equitativa, generar una formación continua, llegar a lugares inaccesibles, así como mejorar la relación entre la escuela y la sociedad (Bruner, 2000).

Lo anterior, presenta un escenario de crecientes necesidades de apoyo a la formación de los estudiantes, donde, por una parte, con menores o mayores avances, hay tecnología ya instalada (computadoras personales, software de propósito general y algunas aplicaciones educativas) y por otra los procesos de actualización de profesores.

#### **5.- OBJETIVOS:**

##### ***5.1 Objetivo General:***

Contar con un STI para el aprendizaje del lenguaje de programación Java básico en un ambiente de sistema de aprendizaje individualizado (SAI) en la UAM Azcapotzalco.

Diseñar las reglas de inferencia del Módulo Tutor de un STI que asista al alumno en el aprendizaje del lenguaje de programación Java básico.

##### ***5.2 Objetivos Específicos:***

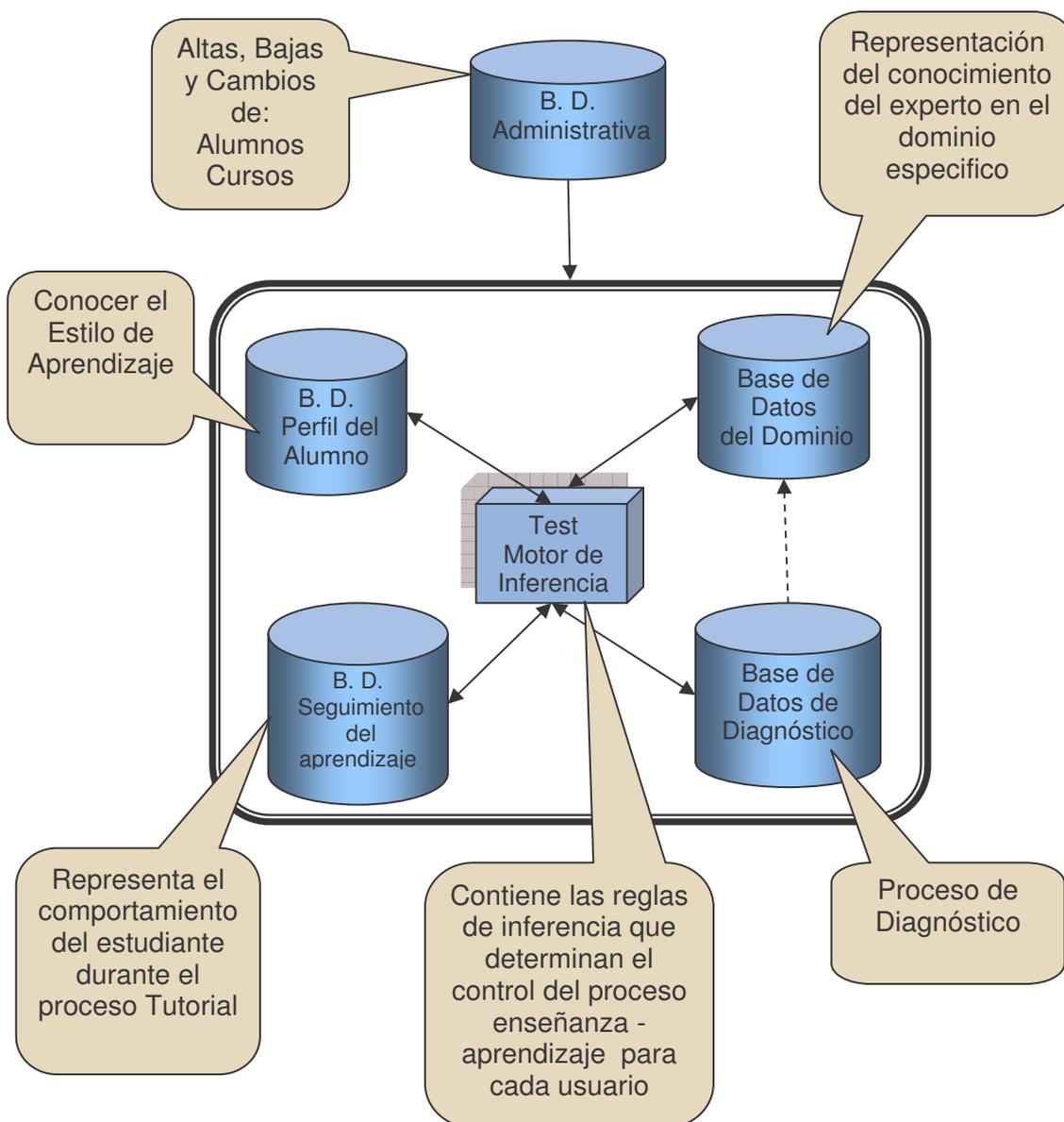
- Elegir el mejor mecanismo de inferencia, para este proyecto (reglas de producción, redes semánticas, lógica de predicados, diagrama de estados pedagógicos).
- Diseñar las reglas de inferencia para el aprendizaje de lenguaje Java básico.
- Diseñar las reglas de inferencia que consideren la planificación y seguimiento de un curso de lenguaje Java básico (simulando un curso tradicional en aula).
- Evaluar las inferencias producidas con fundamento en una base de conocimiento de Java básico.

#### **6.- NUESTRA PROPUESTA**

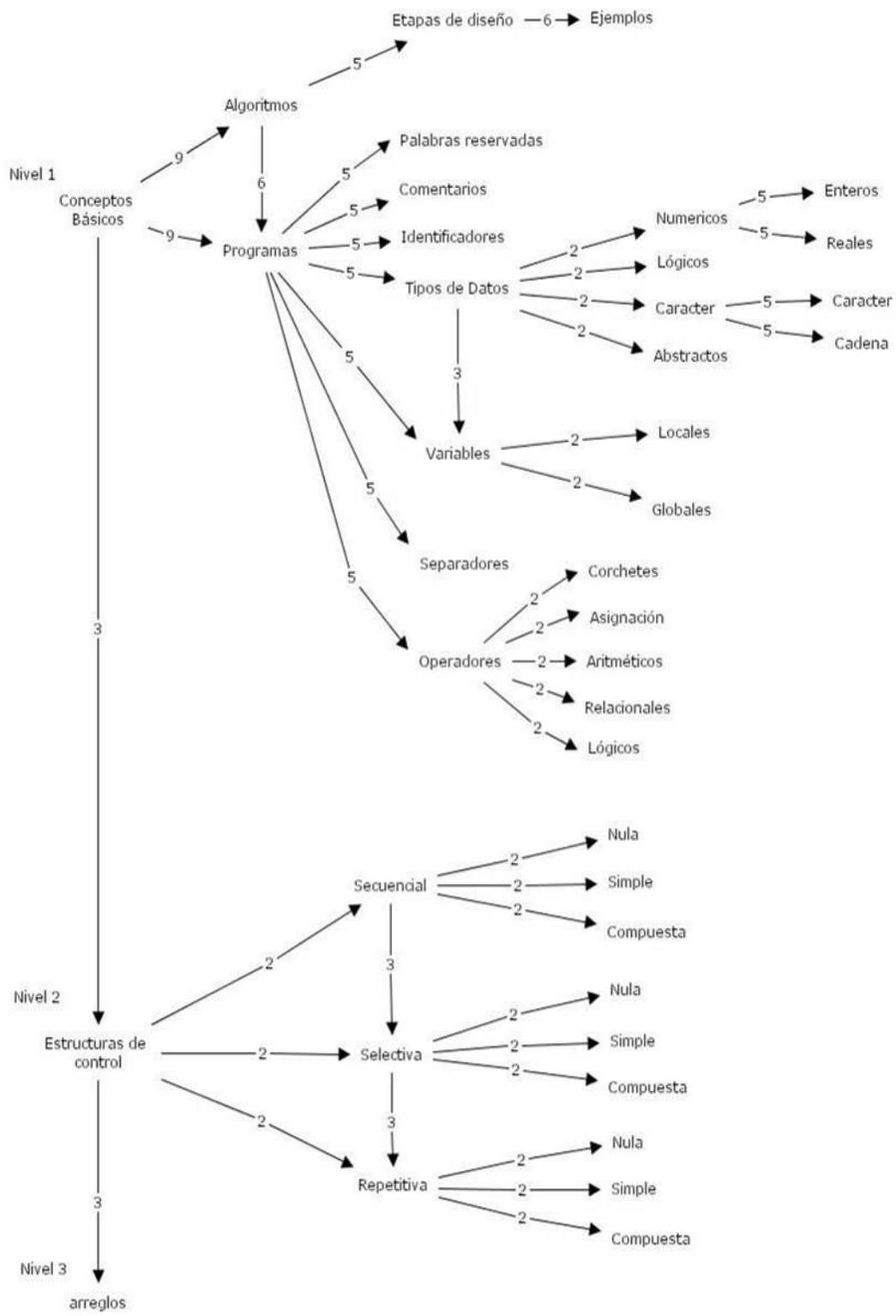
En este caso el proyecto que se propone forma parte de un proyecto incremental en donde la base del conocimiento sobre el lenguaje Java ya fue desarrollada en la tesis "Creación de una Base de Conocimiento para un sistema de enseñanza para el lenguaje Java básico" desarrollado por la alumna Elena Cruz.

A partir de esta base de conocimiento se implementarán las reglas de inferencia. La figura 2 muestra el diagrama de la organización de los módulos que componen la Base de Conocimiento.

Es en esta Base de Conocimiento donde se encuentra almacenada la información asociada al dominio del conocimiento específico, organizado en una serie de conceptos relacionados a temas y unidades a partir del diseño de grafos que muestran la organización y estructura de cada uno de los temas, la secuencia de los mismos así como el tipo de asociación (enlace) que tienen entre si y que forman la gráfica genética figura 3, que es la que se utilizó para la implementación de la Base de Conocimiento del lenguaje de programación Java Básico.



**Figura 2: Descripción de los Componentes de la Base de Conocimiento.**

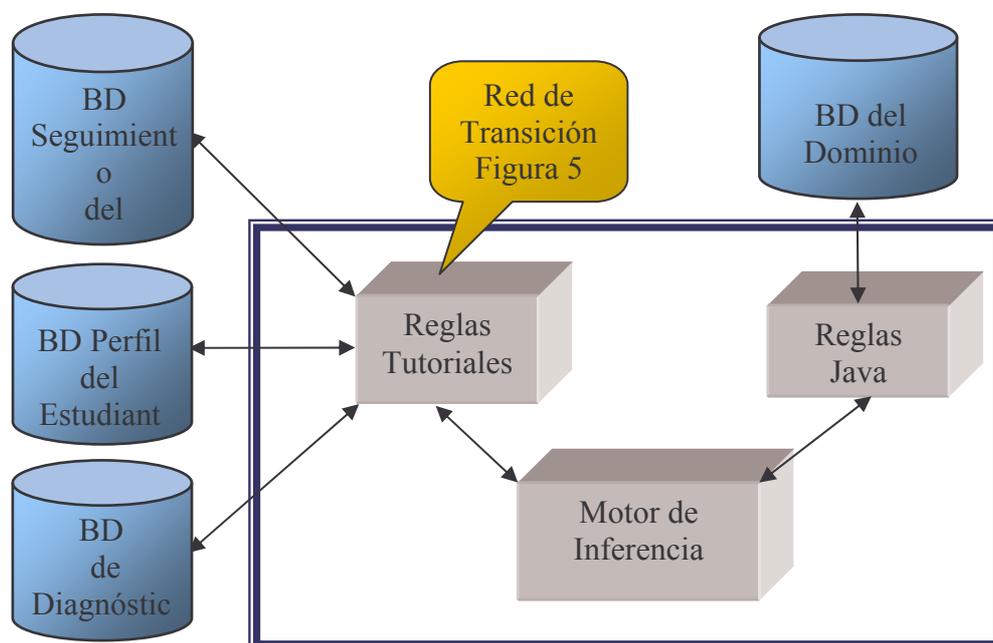


**Figura 3 Grafica Genética de la secuencia de cómo un profesor inicia enseñando conceptos generales de programación estructurada.**

Como ya mencionamos para el diseño de las reglas del Modulo Tutor tenemos que partir de la base de Conocimiento ya creada.

La figura 4 muestra los componentes que forman parte del modulo tutor y la relación con las bases de datos que integran la base de conocimiento ya diseñada.

- Las reglas tutoriales, que serán diseñadas basándonos en la figura 5 y que deciden como asesorar al estudiante sobre un tema específico, y qué técnica utilizar.
- Las reglas de lenguaje Java básico que estarán diseñadas de acuerdo al contenido y la estructura de la base de datos del dominio.
- El motor de inferencia, que procesa las reglas existentes tanto del lenguaje Java, como las reglas tutoriales.



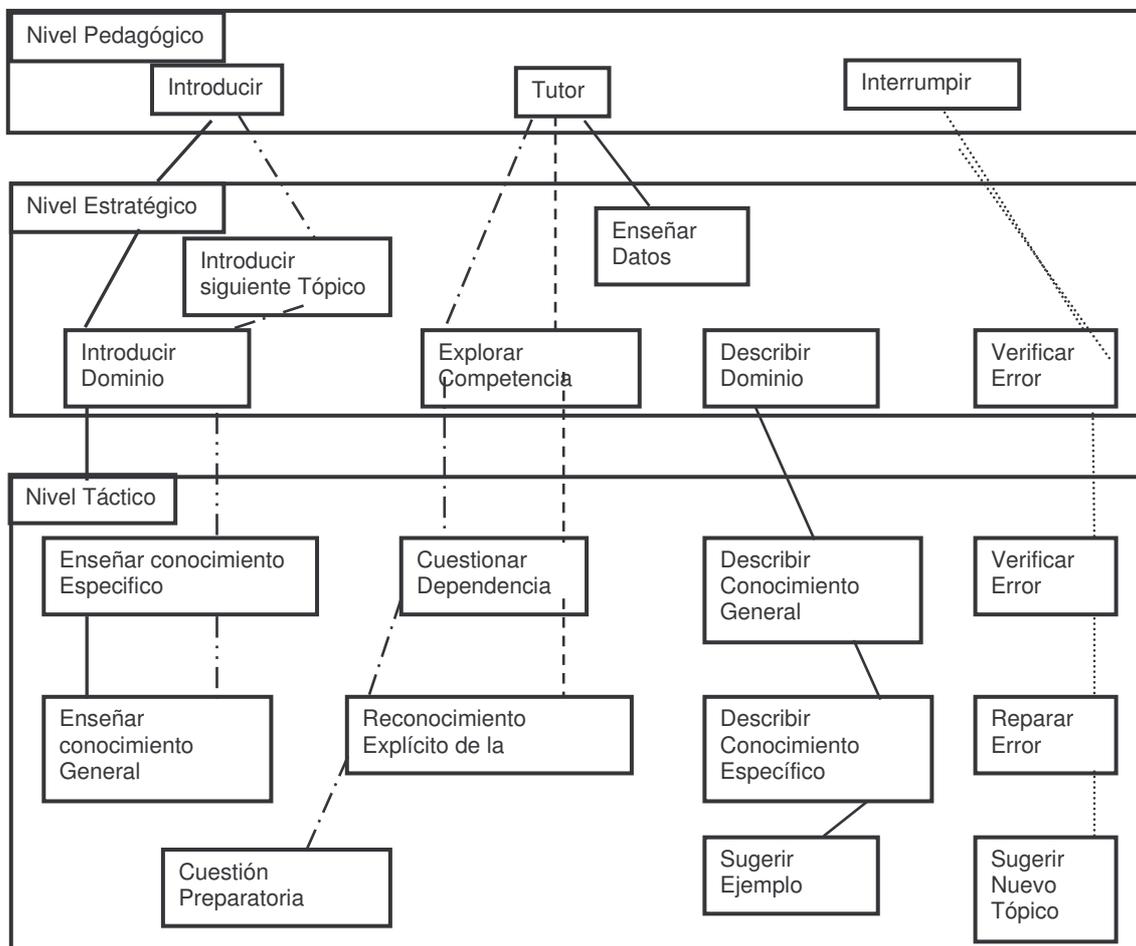
**Figura 4. Diagrama de componentes del modulo Tutor**

El motor de inferencia se diseñara creando un diagrama de estados pedagógicos propuesto por Wolf y McDonald(1984),el componente de tutorización se describe como un conjunto de unidades de decisión organizadas en tres niveles de planificación que pueden definir las acciones del tutor, para poder diseñar estas reglas debemos considerar estos niveles y así poder determinar un resultado adecuado a las preguntas del usuario, estos módulos se observan en la Figura 5.

Otro elemento que se considerará en este diseño para tomar en consideración el proceso de enseñanza – aprendizaje individualizado, es la planificación dinámica propuesta por Fernández (1989).

Cabe mencionar que esta red es una primera aproximación de nuestra propuesta y que conforme vayamos avanzando en el diseño de las reglas tutoriales habrá una variación de los estados que componen esta red.

También debemos mencionar que los estados de esta red están relacionados con las componentes de la base de conocimiento ya diseñada. Esto es, a cada nivel del diagrama de estados pedagógicos se le asociará un tipo de información que permitirá disparar las siguientes acciones, representadas por estados de de un nivel más refinado y así sucesivamente. A continuación se da una explicación más detallada.



**Figura 5 Red de Transición que representa al componente de tutorización.**

Esta red muestra, en el nivel superior (nivel Pedagógico) que se restringe a un tipo de enseñanza específica que determina, con qué frecuencia se interrumpirá al estudiante. El segundo nivel (Nivel Estratégico) refina la decisión pedagógica en estrategia especificando el método que se va a usar (preguntas o descripciones).

El nivel más bajo (Nivel Táctico) se selecciona una táctica para implementar una serie de acciones, en un caso concreto si el estudiante comete un error, el tutor puede elegir entre varias formas de corregirlo. La implementación de este componente es similar a una red de transición de estados, que se atraviesa de estado a estado mediante una rutina iterativa a través de un espacio de caminos.

Los caminos no deben ser fijos ya que existen cambios de estudiante a estudiante y de tema a tema. Esto funcionalmente es una acción que corresponde a las transiciones de alto nivel que se observan en la tutorización humana (Profesor) es en este punto en donde se tomará en cuenta una planificación dinámica.

## 7.- METODOLOGIA

1. Investigación del estado del arte relacionado con el tema.
2. Evaluación y diseño del mecanismo del motor de inferencia.
3. Analizar la base de conocimiento existente para el aprendizaje de Java básico.
4. Diseñar las reglas de inferencia para el aprendizaje de lenguaje Java básico con fundamento en la base de conocimiento existente.
5. Entrevistas y recopilación de información con los expertos en la enseñanza del lenguaje de programación Java, con el fin de esclarecer los mecanismos involucrados en el proceso enseñanza-aprendizaje.
6. Diseñar el diagrama de estados pedagógicos.
7. Evaluar las reglas de inferencia respecto a la base de conocimiento de java básico.
8. implementación del sistema tutorial inteligente.
9. Pruebas.
10. Liberación.
11. Elaboración de informe.

## 8.-CALENDARIZACIÓN

### Proyecto de investigación en Computación I

ACTIVIDAD	SEMANA											
	1	2	3	4	5	6	7	8	9	10	11	
Investigación del estado del arte relacionado con el tema.												
Evaluación y diseño del mecanismo del motor de inferencia.												
Analizar la base de conocimiento existente para el aprendizaje de Java básico.												
Diseñar las reglas de inferencia para el aprendizaje de lenguaje Java básico con fundamento en la base de conocimiento existente.												
Evaluar las reglas de inferencia para el lenguaje Java básico												
Realización de informe de avance												
Presentación de informe de avance												

## Proyecto de Investigación en Computación II

ACTIVIDAD	SEMANA											
	1	2	3	4	5	6	7	8	9	10	11	
Entrevistas y recopilación de información con los expertos en la enseñanza del lenguaje de programación Java, con el fin de esclarecer los mecanismos involucrados en el proceso enseñanza-aprendizaje.												
Diseñar el diagrama de estados pedagógicos.												
Evaluar las reglas de inferencia respecto a la base de conocimiento de java básico.												
Implementación del sistema tutorial inteligente.												
Pruebas												
Liberación												
Realización de informe de avance												
Presentación de informe de avance.												

### 9.- RECURSOS

#### REQUERIDOS:

1. PC Pentium 4
2. Conexión a Internet
3. 512 MB de memoria RAM.
4. Manejador de Base de Datos MySQL.
5. J2SDK1.4.1
6. Base de conocimientos del lenguaje Java básico.
7. Servidor de Aplicaciones Tomcat.

#### DISPONIBLES:

1. PC Pentium 4
2. Conexión a Internet
3. 512 MB de memoria RAM.
4. Manejador de Base de Datos MySQL.
5. J2SDK1.4.1
6. Base de conocimientos del lenguaje Java básico.
7. Servidor de Aplicaciones Tomcat.

## 10.- BIBLIOGRAFÍA

Beverly Woolf And David D. McDonald, "Building a computer Tutor: Design Issues", University of Massachusetts, September, 1994.

Bruce Eckel's , "Thinking in Java", 3rd Edition  
Prentice-Hall, December 2002

Craig Larman, UML Y PATRONES, "Introducción al análisis y diseño orientado a objetos". EDITORIAL Prentice Hall, México, 1999.

Fernández, I. 1989. "Estrategias de Enseñanza en un Sistema Inteligente de Enseñanza Asistida por Ordenador" Tesis Doctoral (tercer Ciclo) de la Universidad del País Vasco, San Sebastián, España.

Froufe Quintas Agustín Y Jorge Cardenes Patricia, "Java 2 Manual de Usuario Y Tutorial", Editorial: RA-MA, México, 2004

Irving M. Copi, "Lógica Simbólica", Editorial Compañía Editorial Continental, S.A de C.V, México, 1998

Jaques, P. 2004. "Using an Animated Pedagogical Agent to Interact Affectively with the Student". Ph.d. Thesis. Universidad Federal do Rio Grande do Sul-Instituto de informática. <http://www.inf.ufrgs.br/~pjaques/>

Laureano-Cruces, A., y de Arriaga, F. 2000. "Reactive Agent Design For Intelligent Tutoring Systems". En la revista Cybernetics and Systems (an International Journal). Vol 31. No. 1, pp. 1-47. ISSN: 0196-9722.

Laureano-Cruces, A., Terán-Gilmore, A, de Arriaga F, y El Alami, M. 2003. "La Importancia De Las Estrategias Cognitivas en el Diseño del Curricula Didáctico". Ana Lilia Laureano Cruces . Vol. I, pp. 35 – 41. En Memorias XVI Congreso Nacional y II Congreso Internacional de Informática y Computación de la ANIEI. Zacatecas, 22-24 de octubre del 2003.

Laureano-Cruces, A., y de Arriaga, F. 2001. "Técnicas de Diseño en Sistemas de Enseñanza Inteligentes". En Revista Digital Universitaria. En año 2001: Vol. 2, Num 1, 31 de marzo de 2001.  
<http://www.revista.unam.mx/>

Laureano-Cruces, A. 2004. "Agentes Pedagógicos" En Memorias del XVII Congreso Nacional y III Congreso Internacional de Informática y Computación de la ANIEI. Tepic, Nayarit, 20-22 de octubre. CD: ISBN 970-36-0155-3.  
<http://delfosis.uam.mx/~ana/>

Laureano-Cruces, A. 2000. "Interacción Dinámica en Sistemas de Enseñanza inteligentes". Tesis Doctoral. Instituto de Investigaciones Biomédicas-UNAM.  
<http://delfosis.uam.mx/~ana/>

Marcellin Jacques Sergio "Inteligencia Artificial, Aprendizaje y Sistemas Expertos", Universidad Nacional Autónoma de México, mayo 1997.

Martha C. Polson & J.J. Richardson. "Foundations of Intelligent Tutoring Systems". LEA. Hillsdale, New Jersey. 1988.

J.A. Vadillo Zorita I. Fernández J. Gutiérrez & J.A. Elarriaga, "Explicaciones en Sistemas Tutores de entrenamiento: Representación del dominio y Estrategias de Explicación" Depto. De Lenguajes y Sistemas Informáticos Universidad del país Vasco España.

Ted Selker, "COACH: A Teaching Agent That Learns ", Communications of the ACM, July 1994.

### **Artículos**

Prof. José Manuel Gutiérrez, Sistemas Expertos Basados en Reglas, Dpto. de Matemáticas Aplicadas. Universidad de Cantabria

N. L. Griffin and F. D. Lewis "A Rule-Based Inference Engine which is Optimal and VLSI Implementable ", Department of Computer Science, University of Kentucky Lexington, Kentucky 40506

Kym Jason Pohl "An Inference Engine-Based Subscription Service", Collaborative Agent Design (CAD) Research Center California Polytechnic University, San Luis Obispo, California, USA

Joseph J. Pfeiffer, Jr. "A Prototype Inference Engine for Rule-Based Geometric Reasoning" Department of Computer Science, New Mexico State University, Las Cruces, NM 88003 USA.

### **REFERENCIAS**

<http://web1.cti.unav.es/asignaturas/ia/tsld011.htm>

<http://www.azc.uam.mx/>

<http://wos3.isiknowledge.com/CIW.cgi>

<http://www.sun.com>

<http://www.BruceEckel.com>.

<http://www.revista.unam.mx>

<http://www.worldscinet.com/>

<http://plato.stanford.edu/entries/logic-fuzzy/>

<http://www.lewismoten.com/LewiesElizaChatbot/default.htm>

<http://aepia.dsic.upv.es/>