

Universidad Central “Marta Abreu” de Las Villas
Facultad de Matemática, Física y Computación



Trabajo de Diploma

HESEI: Herramienta para Sistemas
de Enseñanza/Aprendizaje Inteligentes

Autores:

Eylín Fernández Moré
Maikel León Espinosa

Tutores:

MSc. Natalia Martínez Sánchez
Dra. Zoila Zenaida García Valdivia

2006

Hacemos constar que el presente Trabajo de Diploma ha sido realizado en la facultad de Matemática, Física y Computación de la Universidad Central “Marta Abreu” de Las Villas (UCLV) como parte de la culminación de los estudios de Licenciatura en Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución para los fines que estime conveniente, tanto de forma total como parcial y que además no podrá ser presentado en eventos ni publicado sin la previa autorización de la UCLV.

Firma del Autor

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y que el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma del Tutor
Jefe del Seminario de
Informática Educativa

*"...Enseñar es dejar una huella en las vidas para siempre.
Un estudiante adecuadamente formado es como un vaso que
no admite todo cuanto se quiere verter en él,
sino que discierne y selecciona..."*

*Dedicamos este trabajo a nuestros padres,
quienes han dado todo por ver cumplir
nuestros sueños.*

“La gratitud es la virtud que nos lleva a tomar conciencia de los dones que recibimos cada día, a valorar la generosidad del que nos los da y a mover nuestra voluntad para corresponder a estos dones, aprovecharlos, desarrollarlos y ponerlos al servicio de los demás”.

- Agradecer realmente es una palabra pequeña para expresar todo lo que le debo a mi tutora Zoila Zenaida García Valdivia, no sólo por el tiempo de la tesis, si no por su dedicación durante estos 5 años de carrera, por su ayuda, apoyo, consejos, tiempo y por guiarme. Es para ella mi mayor agradecimiento.
- A mi tutora Natalia Martínez Sánchez, por tanta ayuda, impulso, apoyo e ideas.
- A mi familia en general por tanto ánimo y confianza, en especial a mis padres y hermanas, quienes confiaron siempre en mí.
- A mis compañeros de año por tantos momentos especiales: Norma Elisa, sin ella no hubiese llegado hasta aquí, gracias por todo este tiempo, por tus consejos, por ser mi inspiración. Lisdanay, ejemplo y guía, tantas horas de esfuerzo juntos. A Yadira y Yunia, compañeras de estudio, gracias por soportarme. A Eylín, con quien jamás pensé hiciera tan buena liga.
- A todos los profesores que a lo largo de estos años han influido en mi formación y madurez profesional.
- A los que han colaborado en la programación y escritura del trabajo.
- A todos mis amigos que de una forma u otra han contribuido a que yo alcance esta meta.

Muchas Gracias,

Maikel.

“Agradecer es reconocer, gratificar, apreciar;
agradecer es no olvidar.”

- A mi madre, por su inmenso amor, por su gran ejemplo, su ayuda, su confianza y su dedicación.
- A mi padre, por su amor, su preocupación y su apoyo, a pesar de la distancia.
- A Armando, por sus consejos, su preocupación, sus cuidados y por tanto cariño.
- A Alejandro, mi compañero y amigo, por darme aliento en los momentos más difíciles, por su dedicación, su amor y su permanente compañía.
- A mi fiel amigo Yanko, por su presencia y preocupación, por dedicar parte de su tiempo a escuchar y contribuir con sus ideas a la realización del trabajo.
- A mis tutoras por su ayuda y contribución a la materialización de las ideas; por su dedicación, su valor humano y la confianza depositada en la realización de este trabajo.
- A Maikel, a quien le debo la fuerza y el valor para acometer este trabajo.
- A Dariem, por su inteligencia que no dudó nunca en compartir.
- A Mario, por su gran ayuda en el diseño, imprescindible para la culminación del trabajo.
- A mis queridos amigos: Liana, Norma, Lisdanay, José Jorge, Noeslen, Abdel y Marvin, por estar a mi lado en los momentos de alegría y de tristeza, por su amistad y su ayuda.
- A mis compañeras de cuarto, por su amistad, su apoyo y comprensión.
- A mi familia, por su preocupación y su apoyo espiritual.
- Y a todos mis amigos por los momentos felices compartidos en el transcurso de la carrera.

Gracias a todos,
Eylín.

RESUMEN

El presente trabajo hace uso de las Tecnologías de la Información y las Comunicaciones en el desarrollo de una Herramienta computacional para elaborar Sistemas de Enseñanza/Aprendizaje Inteligentes, denominada HESEI, la cual utiliza técnicas como las de Inteligencia Artificial y Mapas Conceptuales, con el objetivo de adaptar el sistema de enseñanza/aprendizaje, a través de una interfaz visual, a las características del alumno. Permite a expertos no especialistas en computación crear sistemas de enseñanza/aprendizaje inteligentes que aborden contenidos de cualquier especialidad y nivel de enseñanza. En el diseño de HESEI se concibió un módulo de ayuda a la importante y difícil tarea de ingeniería del conocimiento, que utilizando técnicas de Reconocimiento de Patrones, reduce de manera eficiente el número de rasgos con los cuales se describe el modelo cognitivo-afectivo del estudiante. Así, mediante la combinación del Razonamiento Basado en Casos y las técnicas de diagramas conceptuales se logra la adaptabilidad del sistema de enseñanza/aprendizaje inteligente que se genera.

ABSTRACT

This paper centers on the potential uses of computer and communication technologies and the development of a working tool for designing intelligent teaching/learning systems (In spanish HESEI) this tool makes use of artificial intelligence techniques and conceptual maps with the aim of adapting teaching/learning systems to the learners individual needs, by means of a visual interface that permits non-specialists in computers create intelligent teaching/learning systems, related with contents of any field of knowledge or any cognitive level. In the making up of HESEI a helping module for the difficult and important task of learning engineering was devised so that using techniques of pattern recognition, the number of features through which the learners cognitive-affectionate code is described could be effectively reduced. Thus, the combination of a case-based reasoning approach and the techniques of conceptual diagrams result in better and more viable adaptability of the newly generated systems.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I. LAS TÉCNICAS DE INTELIGENCIA ARTIFICIAL Y LOS MAPAS CONCEPTUALES EN LA ENSEÑANZA ASISTIDA POR COMPUTADORA.....	5
1.1 La enseñanza asistida por computadora.....	5
1.2 Uso de las técnicas de Inteligencia Artificial en la enseñanza asistida por computadora	10
1.2.1 Sistemas Expertos.....	12
1.2.2 Sistemas de enseñanza/aprendizaje inteligentes y el Modelo del Estudiante	13
1.2.3 Enseñanza basada en casos, otro enfoque educativo que utiliza técnicas de Inteligencia Artificial	18
1.3 Uso de los Mapas Conceptuales en la enseñanza asistida por computadora como una técnica y estrategia de aprendizaje.....	20
1.3.1 Definición y características de los Mapas Conceptuales.....	21
1.3.2 Tipos de Mapas Conceptuales. Aprendizaje y aplicaciones.....	24
1.4 Conclusiones parciales	28
CAPÍTULO II. IMPLEMENTACIÓN DE HESEI VERSIÓN 1.0	29
2.1 Principales conceptos en el diseño e implementación orientados a objetos	29
2.1.1 Conceptos generales de la Programación Orientada a Objetos.....	29
2.1.2 La metodología de análisis y diseño orientados a objetos.....	30
2.2 Generalidades de HESEI.....	32
2.3 Implementaciones de la interfaz de usuario	38
2.3.1 Editor de preguntas.....	39
2.3.2 Generación de un sistema de enseñanza/aprendizaje inteligente a través de un Mapa Conceptual.....	39
2.3.3 Uso de la componente Vclskin para la selección del estilo de la interfaz.....	40
2.3.4 Autenticación de usuarios mediante LDAP	41
2.4 Uso de las técnicas de Inteligencia Artificial en HESEI.....	43
2.4.1 Reducción del espacio inicial aplicando técnicas de Reconocimiento de Patrones	43

2.4.1.1 LEX. Un nuevo algoritmo para los testores típicos.....	43
2.4.2 Implementación del Modelo del Estudiante usando Razonamiento Basado en Casos	48
2.4.2.1 Arquitectura y aprendizaje de un Sistema Basado en Casos	52
2.4.2.2 La base de casos	53
2.5 Conclusiones parciales	55
CAPÍTULO III. MANUAL DE USUARIOS.....	56
3.1 Inicio de una sesión de trabajo	56
3.2 Interfaz para el usuario Profesor	58
3.3 Interfaz para usuario Estudiante.....	74
CONCLUSIONES	77
RECOMENDACIONES.....	78
REFERENCIAS BIBLIOGRÁFICAS	79
BIBLIOGRAFÍA	83
ANEXOS	84
Anexo 1. Casos de uso del Sistema.....	84
Anexo 2. Expansión de Casos de uso	85
Anexo 3. Diagrama de Transición de Estado.....	86
Anexo 4. Diagrama de Actividades	87

INTRODUCCIÓN

La enseñanza es una actividad sumamente compleja, y a través de la historia el hombre ha experimentado diversos métodos y procedimientos con el propósito de lograr en forma efectiva tanto la enseñanza como el aprendizaje. Por esta razón, la idea de utilizar la computadora como medio de enseñanza es casi tan antigua como la misma, pues prácticamente desde el momento en que éstas aparecieron, surgió el interés de usarlas en el campo educativo.

De acuerdo con Antonio Vaquero “Enseñar es mucho más que dejar aprender. La enseñanza ha de crear los estímulos que activen y aceleren el aprendizaje. El problema radical de la enseñanza es acoplar la mente del alumno a la materia objeto de aprendizaje. Esto implica una enseñanza individualizada de forma que, dada una materia a enseñar, lo ideal es encontrar para cada individuo el transformador adecuado a su nivel de entendimiento y formación, que hiciese el acoplo más adecuado” (Alemán, 1998).

En este sentido, el uso de la computadora en sus diversas modalidades ofrecen, sobre otros métodos de enseñanza, ventajas tales como:

- Participación activa del alumno en la construcción de su propio aprendizaje.
- Interacción entre el alumno y la máquina.
- Posibilidad de dar una atención individual al estudiante.
- Contribución al desarrollo cognitivo del estudiante.
- Control del tiempo y secuencia del aprendizaje por el alumno.
- Retroalimentación inmediata y efectiva, a través de la cual el alumno puede aprender de sus errores.

Los Sistemas de Enseñanza/Aprendizaje Inteligentes (SEI) ayudan en gran medida en el aprendizaje de los estudiantes, es por eso que la mayoría de los SEI que se adaptan bien a las características individuales de los alumnos, en una sesión de aprendizaje, incluyen un módulo denominado Modelo del Estudiante para así poder lograr la adaptación requerida. Estos

sistemas inteligentes se diferencian de los convencionales en que pueden manejar conocimiento estructurado y empírico, no procesan datos, sino conocimiento representado en forma adecuada.

Los SEI constituyen un grupo de aplicaciones de enseñanza que promueven un aprendizaje individual y flexible basado en el conocimiento y comportamiento del usuario. Hasta ahora estos sistemas han demostrado su efectividad en diversos dominios. Sin embargo, su construcción implica un complejo e intenso trabajo de ingeniería del conocimiento, lo que impide un uso más general y aprovechado. Para eliminar estas trabas se plantea el objetivo de construir herramientas de autor que faciliten la construcción de tutores a personas no expertas.

La enseñanza que realiza un SEI suele ser individual y se apoya en un proceso interactivo mixto en el que participan tanto el tutor como el alumno compartiendo el objetivo común de desarrollar un aprendizaje efectivo. En general, conocer su estado de conocimiento para satisfacer sus requerimientos y así poder determinar y aplicar, en cada momento, las estrategias de enseñanza más adecuadas. Estas estrategias deben resolver cuestiones prioritarias que son: qué explicar, con qué nivel de detalle, cuándo y cómo interrumpir al alumno y cómo detectar y corregir sus errores.

Una herramienta para elaborar sistemas de enseñanza/aprendizaje que use técnicas como las de Inteligencia Artificial (IA) y Mapas Conceptuales, logra la adaptación del sistema de enseñanza a las características del alumno con mayor precisión y será de gran utilidad en el aprendizaje del estudiante.

Se pretende con el presente trabajo la elaboración de una herramienta que facilite la creación de SEI, donde la inteligencia se relaciona con el recuerdo de experiencias anteriores, las cuales se sustentan a través del paradigma del Razonamiento Basado en Casos (RBC). Se parte de experiencias docentes donde se han detectado ciertos aspectos que cada vez parecen más importantes para el buen funcionamiento de cualquier asignatura: ¿cómo aprenden los alumnos?, ¿cuándo encuentran problemas en los contenidos?

El SEI es un producto tecnológico diseñado para apoyar el proceso educativo, dentro del cual se concibe como uno de los medios que utilizan quién enseña y quién aprende, para alcanzar determinados propósitos. Además, este software es un medio de presentación y desarrollo de contenidos educativos, como lo puede ser un libro o un video, con su propio sistema de códigos, formato expresivo y secuencia narrativa. De esta manera, el software educativo puede ser visto como un producto y también como un medio.

El diagnóstico es sin duda uno de los procesos más importantes dentro de cualquier SEI, puesto que como ya hemos mencionado, de la calidad del modelo del alumno dependerá la capacidad de adaptación del sistema. No siempre se le presta la atención que merece, dado que el gran esfuerzo que supone desarrollar un SEI hace que a menudo el problema del modelo del alumno se resuelva mediante la aplicación de heurísticas diseñadas a tal fin. Pero la falta de consistencia de dichas heurísticas hace que el comportamiento del sistema sea impredecible, sobre todo en situaciones diferentes a las inicialmente previstas por sus diseñadores. Es por ello que pese al esfuerzo adicional que supone, merece la pena utilizar teorías bien fundamentadas y ampliamente comprobadas que garanticen el funcionamiento óptimo del sistema en todas las situaciones posibles, se propone el uso del paradigma del RBC como marco teórico, el cual proporciona una forma natural y efectiva para el desarrollo de sistemas computacionales de apoyo educacional (Kolodner, 1992).

Objetivo general

Desarrollar una herramienta computacional que facilite la elaboración de sistemas de enseñanza/aprendizaje inteligentes para el apoyo educacional, logrando un puente de enlace entre afectividad y cognición para determinar con la mayor precisión y rapidez posible qué partes del dominio que se pretenden enseñar son las que ya domina el estudiante y cuáles son las que aún desconoce de manera que se decida el estilo de aprendizaje que se adecue al alumno.

Objetivos específicos

- Elaborar una herramienta que permita a expertos no especialistas en computación crear sistemas de enseñanza/aprendizaje inteligentes que aborden contenidos de cualquier especialidad y nivel.
- Implementar un módulo de ingeniería del conocimiento que facilite la selección de rasgos.
- Implementar el Modelo del Estudiante de la herramienta HESEI tomando como marco teórico el Razonamiento Basado en Casos.
- Implementar una interfaz visual para elaborar de manera sencilla un sistema de enseñanza/aprendizaje inteligente con estructura de Mapa Conceptual para una temática determinada.

Preguntas de investigación

- ¿La representación del conocimiento relacionada con el historial del comportamiento del estudiante se facilita a través de casos?
- ¿Qué información se debe almacenar del estudiante?

Este trabajo está estructurado en tres capítulos, en forma abreviada podemos decir que:

En el Capítulo 1 se aborda el tema de la Enseñanza Asistida por Computadoras (EAC), se especifican algunas técnicas para incluir en el diseño de software educativos. Se describen de forma general los software educativos que se adaptan mejor a las características del estudiante. Se tratan temas como la aplicación de la IA en la enseñanza, uso del Modelo del Estudiante para la adaptación del software al alumno y el uso de los Mapas Conceptuales como técnica y estrategia de aprendizaje. En el Capítulo 2 se describe el diseño general de la herramienta HESEI, se explican las diferentes implementaciones de los algoritmos utilizados y se hace mención a la forma en que quedó implementada la herramienta. Por último, en el Capítulo 3 se presenta el manual de usuarios de la herramienta desarrollada.

CAPÍTULO I. LAS TÉCNICAS DE INTELIGENCIA ARTIFICIAL Y LOS MAPAS CONCEPTUALES EN LA ENSEÑANZA ASISTIDA POR COMPUTADORA

En este capítulo se trata la temática de la Enseñanza Asistida por Computadoras, se abordan técnicas para incluir en el diseño de software educativos. Se menciona la aplicación de la IA en la enseñanza, además el uso del Modelo del Estudiante para la adaptación del software educativo a las características del alumno y la aplicabilidad de los Mapas Conceptuales como técnica y estrategia avanzada de enseñanza/aprendizaje.

1.1 La enseñanza asistida por computadora

Desde el comienzo de la historia humana, el aprender ha sido una característica propia de cada individuo que ha contribuido a fundamentar las bases del desarrollo, ya que cuando se aprende se adquiere el conocimiento por medio del estudio, el ejercicio o la experiencia.

A medida que se ha ido evolucionando como civilización, varias personas se han dedicado al estudio del comportamiento humano, esto ha conllevado a la definición de diversos criterios con respecto al aprendizaje; conceptos, métodos, herramientas y técnicas. Así, con el decursar del tiempo, las exigencias de la vida moderna, dinámica, competitiva y llena de información han dirigido a la sociedad a buscar nuevos modelos, técnicas y sistemas que permitan adquirir esos conocimientos de una manera eficaz y eficiente.

En una sociedad impactada por la ciencia y la tecnología, todo ciudadano necesita de una cultura científico-tecnológica para entender, integrarse y actuar en el mundo que lo rodea.

La incorporación de metodologías de enseñanza auxiliadas por computadora incrementa la eficiencia en la transmisión y adquisición del conocimiento, permitiendo la manipulación de

grandes volúmenes de información, la homogenización del conocimiento, la disponibilidad, masificación y la permanente actualización del mismo.

Así, es muy importante interpretar adecuadamente la función que desempeña la computadora como herramienta para representar el conocimiento y su carácter de instrumento pedagógico.

En (Bello, 2002) se ofrecen diferentes dimensiones del uso de la computadora en la educación donde se incluyen:

- **La computadora como objeto de estudio:** Aprender acerca de la computadora (educación acerca de la computación).
 - Alfabetización computacional.
 - Programación de computadoras y la enseñanza de la solución de problemas.
 - Formación de especialistas en Computación e Informática.
- **La computadora como herramienta de trabajo:** Uso de las aplicaciones de la computadora para apoyar procesos educativos (educación complementada con la computadora).
- **La computadora como medio de enseñanza/aprendizaje:** Ambientes de enseñanza/aprendizaje enriquecidos con la computación donde se utiliza la computadora para presentar lecciones completas a los alumnos: Enseñanza Asistida por Computadora.

Los autores del presente trabajo están de acuerdo en que la principal ventaja de la EAC es la interacción, puesto que permite, a través del diálogo instructivo, que el estudiante no asuma una actitud pasiva en el proceso de aprendizaje, logrando captar su atención y contribuyendo a elevar su memoria visual. Además, las explicaciones, preguntas, correcciones y la evaluación se hacen de modo inmediato, con lo que la efectividad pedagógica se acentúa, al individualizarse el trabajo y permitiendo a cada alumno trabajar a su propio ritmo.

Diferentes tipos de técnicas se han desarrollado en correspondencia con estos objetivos. Entre los cuales se encuentra la aplicación de la IA en los sistemas de EAC.

En la actualidad, existe la tendencia a incorporar a la educación tecnologías de la información y comunicación, por ejemplo, multimedia, conexión en red (tanto locales como Internet), así como nuevas posibilidades de navegación a través de la información (hipertexto). Estas tecnologías hacen posible el desarrollo de software educativos cada vez más sofisticados que incorporen interfaces amigables, permitiendo distintos tipos de interacción con los usuarios.

Las expectativas que crea la computadora como medio de enseñanza/aprendizaje se fundamenta tanto en las características técnicas que ofrece la máquina como en las potencialidades de desarrollo que abarca la Informática Educativa.

Una de las características de la computadora es que permite la interacción entre ella y el usuario, de allí que con este medio pueda ofrecerse algo mejor o diferente que con otros medios para promover ciertos aprendizajes. Una buena utilización del medio computacional en la educación depende, en gran medida, de lo interactivo que sea el material.

A esta interacción que la computadora facilita a través de la interfaz usuario-máquina, se añade la capacidad de almacenamiento, procesamiento y transmisión de la información.

El software educativo, por su parte, trata de complementar determinadas limitaciones de otros medios de enseñanza/aprendizaje. De ninguna manera, se debe pretender reemplazar con un software educativo la acción de algunos medios educativos cuya calidad, efectividad y valor práctico están bien demostrados.

Se conoce como software educativo el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto enseñanza/aprendizaje. Ahora bien, el desarrollo de software educativos requiere de grupos multidisciplinarios donde intervengan al menos educadores y especialistas en computación. Estos programas abarcan finalidades muy diversas que pueden ir desde la exposición conceptual hasta el desarrollo de actividades que permitan la adquisición de habilidades básicas, o la resolución de problemas. Algunos autores suelen nombrarlos Software Instructivos, Programas Instructivos, Materiales Educativos Computarizados (MEC), entre otros.

A continuación se manejan diferentes enfoques para la clasificación de tipos de software educativos:

1. Galvis en su obra “Ingeniería de software educativo”, presenta las clasificaciones siguientes de acuerdo a:

- Enfoque educativo:
 - Algorítmico.
 - Heurístico.
- Función educativa:
 - Sistema tutorial.
 - Sistema de ejercitación y práctica o entrenadores.
 - Simulador.
 - Juego educativo.
 - Micromundo exploratorio.
 - Sistemas expertos.
 - Sistemas inteligentes de enseñanza/aprendizaje (Galvis, 1994).

2. Sevillano en su obra “Nuevas Tecnologías, Medios de Comunicación y Educación” expone una clasificación de acuerdo al uso del software educativo:

- Uso instruccional:
 - Programas tutoriales.
 - Programas de ejercitación y práctica.
- Uso demostrativo o conjetural:
 - Simulaciones.
 - Realidad virtual.
 - Los juegos realísticos o juegos de roles.
- Uso con función de entretenimiento:
 - Se aprende jugando.
- Usos especiales:
 - Para alumnos con deficiencias.
- Uso emancipado:

- Laboratorios.
- Telecomunicaciones.
- Procesadores de textos.
- Bases de datos.
- Hojas de cálculo.
- Paquetes estadísticos.
- Diseño de gráficos (Sevillano, 1998).

3. El grupo de Informática Educativa de la Universidad Central “Marta Abreu” de Las Villas (UCLV) comparte el criterio de clasificación de software educativos según:

a) Función educativa:

- Sistema tutorial.
- Sistema de ejercitación y práctica (entrenador).
- Simulaciones.
- Juegos didácticos.

b) Forma de presentación:

- Multimedia.
- Hipermedia (García et al., 2000).
- Sitios web.

c) Uso o no de técnicas de IA:

- Sistemas convencionales.
- Sistemas inteligentes.

Estos sistemas a su vez pueden ser del tipo a) o b).

Estas clasificaciones no son excluyentes, en un mismo software educativo las mismas pueden combinarse.

A continuación se describen los tipos de software educativo fundamentales para una mejor comprensión del tema, expuestos por:

- **Sistema tutorial:** Por lo general este tipo de sistema incluye cuatro fases del proceso de enseñanza/aprendizaje que son la introductoria, en la que se genera la motivación, se centra la atención y se favorece la percepción selectiva de lo que se desea que el alumno aprenda; la fase de orientación en la que se enseña la teoría del tema a tratar; la fase de aplicación, en la que hay evocación y transferencia de lo aprendido y la fase de retroalimentación en la que se demuestra lo aprendido y se ofrece retroinformación y refuerzo. Esto no significa que todos los tutoriales tengan que ser iguales.
- **Simulador y Juego educativo:** Poseen la cualidad de apoyar aprendizaje de tipo experiencial y conjetural, como base para lograr aprendizaje por descubrimiento. La interacción con un micromundo, en forma semejante a la que se tendría en una situación real, es la fuente de conocimiento. En una simulación, aunque el micromundo suele ser una simplificación del mundo real, el alumno resuelve problemas, aprende procedimientos, llega a entender las características de los fenómenos y cómo controlarlos, o aprende qué acciones tomar en diferentes circunstancias. Las simulaciones intentan apoyar el aprendizaje simulando situaciones de la realidad. Por su parte los juegos pueden o no simular la realidad pero sí se caracterizan por proveer situaciones excitantes o entretenidas. Los juegos educativos buscan que dicho entretenimiento sirva de contexto al aprendizaje de alguna temática.
- **Los entrenadores:** Están primordialmente orientados al desarrollo de habilidades, pues no pretenden la dirección total del proceso de instrucción ni llevan a cabo la formación de conceptos nuevos. Sólo supervisan la actividad práctica del estudiante mediante el control de los errores durante la solución de los ejercicios, hacen recomendaciones y controlan la presentación dosificada de problemas y ejercicios.

1.2 Uso de las técnicas de Inteligencia Artificial en la enseñanza asistida por computadora

La IA se ha desarrollado como una de las ramas de la Ciencia de la Computación. El trabajo en la misma comenzó en la década de 1950 y dicho término fue usado a partir de 1956.

“No hay una definición definitiva y única de la Inteligencia Artificial. G.F. y W.A. Stubblefield, en su libro Artificial Intelligence: structures and strategies for complex problem solving, la definen como la rama de la Ciencia de la Computación que se relaciona con la automatización del comportamiento inteligente” (Bello, 2002).

Más que la existencia de una definición formal y acabada de qué se entiende por IA, lo importante es conocer que la misma está dotada de métodos para resolver problemas en los cuales el enfoque algorítmico tradicional de la computación no es suficiente.

La IA es un campo del saber que ha experimentado un rápido desarrollo y se ha diversificado a partir de la década de 1990. Entre las vertientes principales de desarrollo de la IA se encuentran los sistemas expertos y los SEI.

Paralelamente, la IA también ha abordado el desarrollo de sistemas inteligentes de apoyo a la enseñanza. Normalmente las propuestas pedagógicas asumidas para lograr el aprendizaje se basan en distintas tareas que realiza el estudiante, entre las cuales se encuentran comprender, analizar, resolver problemas y evaluar aplicando los conocimientos adquiridos.

Dentro de los sistemas virtuales de aprendizaje se pueden destacar los sistemas tutoriales inteligentes, hipermedias educativas y las simulaciones.

Todos estos sistemas virtuales son programas interactivos que tienen como objetivo facilitar el aprendizaje, adecuando su funcionamiento a las características del alumno. Esta adaptación puede consistir en la selección del contenido de sus presentaciones de acuerdo con las acciones previas del estudiante, su nivel de conocimiento, sus preferencias u otras características. Esta información es procesada y almacenada en el perfil del usuario durante la interacción con el sistema y posteriormente analizada para adaptar nuevamente la información a presentar.

1.2.1 Sistemas Expertos

Autores como (Galvis, 1994) y (Alessi and Trollip, 1985) incluyen a los sistemas expertos como uno de los tipos de sistemas de EAC que utilizan técnicas de IA. Estos constituyen un valioso recurso en el proceso docente.

Un sistema experto es un programa que contiene una representación del conocimiento existente sobre cierto dominio de aplicación, incluye la información conocida de un tópico particular y sobre todo, estrategias para depurar y ofrecer dicho conocimiento. Lo anterior convierte al sistema en un especialista en la materia para la cual está programado (García, 1993).

En (Galvis, 1994) se plantea que desde el punto de vista usuario-aprendiz, un sistema experto además de demostrar gran capacidad de desempeño en términos de velocidad, precisión y exactitud, tiene como contenido un dominio de conocimientos que requiere de gran cantidad de experiencia humana, no solo principios y reglas de alto nivel, si no que es capaz de hallar o juzgar la solución a algo, explicando o justificando lo que halla o juzga, de modo que es capaz de convencer al usuario de que su razonamiento es correcto.

Esta capacidad de razonar como un experto es lo que hace que estos sistemas sean útiles para que los aprendices ganen experiencia en dominios en que es necesario obtenerla y hagan explícito el conocimiento que está detrás de ella.

La complejidad de elaboración de un sistema experto se compensa con la facilidad de su aplicación didáctica permitiendo, según (Laborda, 1986):

- Resolver problemas complicados de manera que su forma de operar sirva de guía para el alumno.
- Organizar el saber disponible sobre la materia, posibilitando su aplicación directa a la solución del problema.
- Preservar el conocimiento para su utilización futura.

- Establecer una comunicación eficiente con el estudiante.
- Captar y presentar en diferentes formas las respuestas que recibe o proporciona.
- Reconocer una extensa gama de errores de razonamiento.
- Proveer conjuntos de problemas distintos y graduar su dificultad relativa.

Otras razones que dan utilidad educativa a un sistema experto son las siguientes:

En la etapa de su diseño, el esfuerzo intelectual que exige formalizar el conocimiento que se va a incluir puede llevar al diseñador a un mayor o más detallado conocimiento del tema que se trata. Por otra parte, la estructura misma de un sistema experto permite ver el conocimiento y el proceso de aprendizaje desde una perspectiva sistémica como una unión de información o conocimiento y manipulación de ese conocimiento. Por último, dado que un sistema experto encierra lo que sabe un experto especialista acerca de un dominio específico, resulta razonable pensar en él como base de un sistema individualizado de aprendizaje apoyado por la computadora en ese dominio.

1.2.2 Sistemas de enseñanza/aprendizaje inteligentes y el Modelo del Estudiante

Los sistemas expertos no constituyen la única vía para el empleo de las técnicas de IA en la enseñanza/aprendizaje. Se puede hacer referencia a un término más general, como son los SEI.

Por su parte, los SEI enfocan una sesión de trabajo como un proceso de cooperación entre el sistema y el alumno, con el objetivo de propiciar el aprendizaje. El sistema debe analizar en cada momento el comportamiento del estudiante para caracterizar su actuación y decidir cuál es la estrategia más adecuada a aplicar; qué explicarle, con qué nivel de detalle, cuándo interrumpirle, cómo corregirle, de forma que culmine con éxito el proceso de aprendizaje. Un aspecto fundamental es que para decidir y aplicar dicha estrategia, es necesario conocer la materia que se imparte y comprender la forma en que se asimila (García, 1993).

En (Bello, 2002) se exponen las variantes metodológicas para los SEI:

- Tutores inteligentes.

- Entrenadores inteligentes.
- Sistemas inteligentes basados en simulación.
- Juegos inteligentes.

Los Sistemas Tutoriales Inteligentes fueron una de las primeras aplicaciones prácticas de las ciencias del conocimiento, resultando ser un éxito desde el punto de vista académico.

Los Sistemas Tutoriales Inteligentes plantean un problema y luego examinan la solución dada por el estudiante y los pasos intermedios si los hubiera. A través de un análisis de las respuestas del estudiante, los Sistemas Tutoriales Inteligentes tratan de inferir lo que el estudiante sabe y no sabe, y en función de ello presentan explicaciones adecuadas y nuevos problemas.

A grandes rasgos y sin entrar en sus particularidades resulta conveniente señalar que todo Sistema Tutorial Inteligente se ajusta a una arquitectura compuesta por:

- **Modelo Pedagógico:** Es el módulo capaz de pronosticar la evolución del alumno, toma las decisiones sobre cómo impartir el conocimiento, es decir, se especifica la forma de presentar el material o dosificación de la enseñanza/aprendizaje.
- **Modelo de Comunicación:** También denominado interfaz, es el encargado de establecer la comunicación entre el estudiante y el sistema.
- **Conocimiento del Dominio:** Módulo que recoge toda la información sobre la asignatura (temario, problemas, ejercicios, esquemas, exámenes, etc.), es donde se representa el contenido temático a enseñar.
- **Modelo Experto en el Dominio de Interés:** Es capaz de comprender las respuestas del alumno hacia el sistema y verificar si son correctas o no.
- **Modelo del Estudiante:** Recoge datos históricos sobre los resultados y características del alumno. En este módulo se almacena toda la información posible que se dispone del alumno, se evalúa su estado de conocimiento y sus características.

El objetivo principal de un Sistema Tutorial Inteligente, es proporcionar ayuda en el proceso de enseñanza/aprendizaje y para que esto se pueda llevar a cabo de una manera efectiva y eficiente es necesario que éste desempeñe algunas funciones importantes:

- Interactuar con el estudiante por medio de un diálogo de iniciativa mixta, donde tanto la computadora como el estudiante puedan iniciar preguntas y esperar respuestas razonables.
- Indicar al estudiante las estrategias apropiadas para atacar un problema en particular y demostrarle las aplicaciones de estas estrategias en problemas concretos.
- Responder al estudiante cualquier pregunta pertinente en términos comprensibles por éste.
- Decir al estudiante no sólo que él está errado, hay que indicarle también el método correcto de solución y hacer hipótesis basadas en la historia de sus errores, que es donde se encuentra la verdadera fuente de dificultades.
- Ser capaz de resolver todos los problemas propuestos, si es posible de diferentes maneras.
- Entender y criticar las soluciones encontradas por los estudiantes.
- Crear un Modelo del Estudiante y modificarlo continuamente basándose en su desempeño y en reglas de enseñanza.
- Tomar acciones que atiendan las deficiencias o los logros del estudiante al resolver problemas que se le presentan.
- Ser capaz de combinar métodos algorítmicos y heurísticos.

La mayoría de las funciones antes mencionadas sólo son posibles si estos sistemas generan y mantienen un Modelo del Estudiante, el cual se usa como base de todo su razonamiento y en consecuencia con su comportamiento.

Uno de los módulos de los Sistemas Tutoriales Inteligentes que más se ha favorecido con la IA es el Modelo del Estudiante pues en su elaboración se utilizan modelos de naturaleza cualitativa que a partir del análisis de la respuesta del estudiante y el proceso de inferencia que

realiza el sistema se logra una mejor adaptación a las características individuales del estudiante.

El Modelo del Estudiante es la imagen que el sistema tiene del estudiante, entendiendo el término imagen como posible interpretación de la realidad.

¿Qué representa el Modelo del Estudiante?

- El conocimiento que el sistema cree que ya adquirió el estudiante.
- El conocimiento estructuralmente relacionado con el adquirido.
- Intereses del estudiante (tendencias en sus preferencias).
- La manera en que el estudiante aprende.
- Errores y conceptos mal interpretados (Ayala, 2002).

Este modelo representa al estudiante, desde dos perspectivas diferentes:

- Conocimiento que éste tiene sobre el dominio del tema a tratar.
- Aspectos psicosociológicos característicos que condicionan el proceso de aprendizaje.

Principales funciones del diseño del Modelo del Estudiante:

- Obtener prototipos de estudiantes a partir de los modelos de estudiantes más típicos.
- Resolver problemas interactivamente con el alumno como táctica pedagógica.
- Interpretar lo que ha hecho un estudiante en particular.
- Conocer lo que sabe un estudiante para predecir lo que es capaz de hacer.

Particularmente, los entrenadores inteligentes están esencialmente orientados al desarrollo de habilidades, pues no pretenden la dirección total del proceso de instrucción ni llevan a cabo la formación de conceptos nuevos. Sólo supervisan la actividad práctica del estudiante mediante el control de los errores durante la solución de los ejercicios, hacen recomendaciones y controlan la presentación dosificada de problemas y ejercicios.

Un buen Sistema Tutorial Inteligente es aquel sistema capaz de ayudar al estudiante en el aprendizaje de diversos conocimientos. Se podría decir también que viene a ser como un sistema semi-presencial de conocimiento, debido a que el profesor está ausente y es el propio sistema el que guía al alumno en la asimilación de los diferentes conceptos.

Un Sistema Tutorial Inteligente debe adaptarse al estudiante en cuanto a:

- Su manera de aprender.
- Sus necesidades.
- Sus oportunidades de aprendizaje en base a la estructura del conocimiento.

Los Sistemas Tutoriales Inteligentes no están libres de limitaciones, en (Wenger, 1987) y (Rosemberg, 1990) se señalan las siguientes deficiencias que aún persisten:

- Alto costo de desarrollo.
- Alto costo del equipamiento requerido para su uso.
- Incapacidad para manipular dominios complejos.
- Carencia de métodos de representación del conocimiento que faciliten el acceso a dicho conocimiento.
- Necesidad de añadir al dominio conocimiento secundario relacionado.

Los Sistemas Tutoriales Inteligentes utilizan técnicas de IA como la representación del conocimiento para modelar el razonamiento humano, enfatizan el aprendizaje por medio de la acción, combinan experiencias de resolución y descubrimiento, son capaces por sí mismos de resolver problemas, formular diagnósticos y proporcionar explicaciones, para lo cual, cuentan con un banco de estrategias de instrucción para poder decidir qué y cómo informar al usuario para lograr una orientación efectiva.

1.2.3 Enseñanza basada en casos, otro enfoque educativo que utiliza técnicas de Inteligencia Artificial

En (Cabrera, 1995) se plantea que una de las técnicas más avanzadas de la EAC es la Enseñanza Basada en Casos.

La Enseñanza Basada en Casos nace como una extensión de las técnicas de IA conocida como RBC y está basada en dos suposiciones:

- El razonamiento humano consiste en el acceso y la combinación de casos o memorias de situaciones experimentadas con anterioridad.
- La mejor forma de enseñar es contar historias reales (por lo menos de apariencia real) de las que el estudiante puede construir los casos necesarios para razonar adecuadamente.

La Enseñanza Basada en Casos se parece a los Sistemas Tutoriales Inteligentes con respecto a que la entrega de la información se adapta en lo posible a las necesidades de cada estudiante, sin embargo, el conocimiento en ambos tipos de sistemas está representado de formas muy distintas.

En la Enseñanza Basada en Casos, el conocimiento se codifica en forma de historias, mientras que el conocimiento del Sistema Tutorial Inteligente se halla dentro de un sistema experto interno. En ambos casos, la introducción de nueva información está reservada a profesionales informáticos de gran experiencia. El reto que ambos sistemas deben solucionar es el desarrollo de interfaces de usuario que permitan a un profesor con mínima experiencia con el sistema incorporar nueva información o modificar la existente.

En los diferentes tipos de sistemas que se pueden desarrollar en el campo de la Informática Educativa, las herramientas basadas en casos se pueden usar para construir, acceder y usar casos como parte de un marco computacional más amplio.

Una componente importante de cualquier proceso de aprendizaje es satisfacer las necesidades de información, acceder a la información que permite responder una pregunta, tomar una decisión, resolver un problema o ayudar a comprenderlo.

Cada caso representa un conocimiento específico y la manera en que debe ser manejado; un caso es por tanto, en la investigación presentada, la situación en la que un estudiante busca cumplir un objetivo de instrucción, dicha situación se maneja bajo ciertas condiciones, de tal forma que, en el momento en que las condiciones varían (cambio de estudiante) se genera un caso nuevo.

El acceso a información es una actividad fundamental en el aprendizaje. La ventaja distintiva del empleo de una biblioteca basada en casos para acceder a la información es la posibilidad de integrar gran cantidad de información en representaciones alternativas y de buscarlas asociativamente.

Cada caso cuenta con un contexto en el cual una porción del conocimiento es aplicable y con índices que permiten su identificación, recuperación y almacenamiento; la estructura interna de los casos, especialmente las diferencias, permiten al razonador enfocar adecuadamente la solución o las características del caso recuperado para solucionar el nuevo caso, o sea, adaptar el proceso de instrucción a un nuevo estudiante.

Resulta fundamental integrar los resultados y experiencias en la elaboración de SEI del tipo tutor o entrenadores con la Enseñanza Basada en Casos, de manera que el empleo del RBC no sea solo para presentar los problemas al estudiante de acuerdo a la actuación de este en las tareas que se le presente (modelo cognitivo del estudiante como una representación dinámica de la evolución del conocimiento) sino que también permita representar los rasgos más comunes que caracterizan al tipo de estudiante (excelente, bueno, regular y malo) ante una materia dada de manera que el sistema sea capaz de clasificar al estudiante y poder adaptarse a sus características en el proceso de tutoría o entrenamiento.

El RBC no es solo importante para estructurar el software educativo sino también para representar los rasgos distintivos de los tipos de estudiantes, de modo que a través de una función de semejanza definida se pueda clasificar al alumno que interactúa con dicho software y así lograr una mayor adaptación de éste a las características del estudiante similar a como lo hace un profesor experimentado.

1.3 Uso de los Mapas Conceptuales en la enseñanza asistida por computadora como una técnica y estrategia de aprendizaje

En obra citada por Vivian Estrada se hace referencia a que la convergencia entre las herramientas y los métodos pedagógicos con las tecnologías de la información y las comunicaciones producen una cualidad nueva que favorece la tarea de generar, transmitir y compartir el conocimiento. Tal es el caso de la fortaleza pedagógica del empleo de los Mapas Conceptuales (Bello, 2002).

Los Mapas Conceptuales iniciaron su desarrollo en el Departamento de Educación de la Universidad de Cornell, EUA, durante la década de 1970 y constituye una respuesta a la teoría del aprendizaje significativo desarrollada por Ausubel, en especial, en lo referente a la evolución de las ideas previas que poseen los estudiantes para lograr un nuevo conocimiento.

Los Mapas Conceptuales han constituido desde entonces, una herramienta de gran utilidad para profesores, investigadores de temas educativos, psicólogos, sociólogos y estudiantes en general, así como para otras áreas sobre todo cuando se necesita tratar con grandes volúmenes de información.

Construir y compartir conocimientos, aprender significativamente, contextualizar el aprendizaje y aprender a aprender, son ideas sobre cuyo significado se ha reflexionado desde hace mucho tiempo y se ha intentado utilizar herramientas que permitan llevar a la práctica estas aspiraciones. Para ello se han empleado diferentes técnicas y estrategias. Precisamente una de estas técnicas que ayudan a aprender son los Mapas Conceptuales. Se trata de

garantizar un proceso de aprendizaje que desarrolle la actitud crítica de los estudiantes y su capacidad para la toma de decisiones, rasgos distintivos del proceso de aprender a aprender.

En los últimos años, los Mapas Conceptuales han alcanzado una gran popularidad y en su integración con las tecnologías de la información y las comunicaciones se han convertido en un elemento muy importante en los planes de perfeccionamiento de los sistemas de enseñanza y han extendido su uso a otras esferas de la actividad humana en las que la gestión y uso del conocimiento ocupe un lugar preponderante.

1.3.1 Definición y características de los Mapas Conceptuales

Si definir un Mapa Conceptual es relativamente sencillo, más sencillo es comprender el significado de la definición. El Mapa Conceptual es descendiente del cuadro sinóptico que el profesor de la enseñanza media escribía en la pizarra para que se entendiese mejor lo que estaba explicando, fue definido por Novak, su creador, como una técnica que representa, simultáneamente, una estrategia de aprendizaje, un método para captar lo más significativo de un tema y un recurso esquemático para representar un conjunto de significados conceptuales incluidos en una estructura de proposiciones (Novak and Gowin, 1988).

Elementos básicos de los Mapas Conceptuales:

- **Los conceptos:** También llamados nodos, son regularidades en los acontecimientos o en los objetos que se designan mediante un término. Los conceptos hacen referencia a acontecimientos que son cualquier evento que sucede o puede provocarse y a objetos que son cualquier elemento que existe y se puede observar. Los conceptos son, según (Novak and Gowin, 1988), desde la perspectiva del individuo, las imágenes mentales que provocan las palabras o signos con las que se expresan regularidades. Estas imágenes tienen elementos comunes en todos los individuos y matices personales de quien las percibe.
- **Las palabras-enlace:** Son palabras que unen los conceptos y señalan los tipos de relación existente entre ambos. Siempre que sea posible debe garantizarse que una sola

palabra describa la relación entre dos conceptos. El enlace debe quedar suficientemente claro.

- **Las proposiciones:** Están constituidas por conceptos y palabras-enlace. Es la unidad semántica más pequeña que tiene valor de verdad. Su lectura debe ser fácil de comprender.

En el Mapa Conceptual se organizan estos elementos, relacionándose gráficamente y formando cadenas semánticas, es decir, cadenas que posean un significado. De esta manera el Mapa Conceptual más pequeño que se puede construir, será aquel que esté compuesto por dos conceptos, una palabra-enlace y la proposición que surge de esta relación.

Por ejemplo, los conceptos Árbol y Hojas pueden ser relacionados mediante la palabra-enlace “Posee” y se construye un mapa como el que se muestra en la figura 1.1:

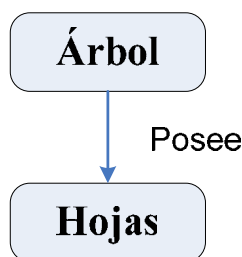


Figura 1.1 Ejemplo de relación entre los conceptos Árbol y Hojas

En este ejemplo “Posee” actúa como palabra-enlace y se forma una proposición que se lee, árbol posee hojas. Este Mapa Conceptual es muy simple, pero pueden alcanzar cualquier complejidad y estar conformado por cientos de conceptos y palabras-enlace, en correspondencia con el tema que está representando.

Se debe aclarar, que no hay un Mapa Conceptual correcto único, pueden existir varios. Lo importante son las relaciones que se establecen entre los conceptos a través de las palabras-enlace para formar proposiciones que configuran un valor de verdad sobre el objeto estudiado. Por tal razón, en torno a un concepto pueden señalarse diversidad de valores de verdad. De

hecho, resulta muy difícil encontrar dos Mapas Conceptuales exactamente iguales, debido al carácter individual del conocimiento.

Las características básicas de un Mapa Conceptual son:

- Organización del conocimiento en unidades o agrupaciones holísticas, es decir, que cuando se activa una de éstas, también se activa el resto.
- Segmentación de las representaciones holísticas en subunidades interrelacionadas.
- Estructuración serial y jerárquica de las representaciones.

Según (Gaines and Shaw, 1995), los Mapas Conceptuales pueden ser descritos bajo diversas perspectivas de acuerdo con el nivel de análisis considerado:

- **Perspectiva abstracta:** Los Mapas Conceptuales constituidos por nodos ligados por arcos pueden ser vistos como hipergrafos ordenados. Cada nodo tiene un identificador único y un contenido y los enlaces entre nodos pueden ser direccionados o no direccionados, representados visualmente por líneas entre los nodos con o sin flechas en las extremidades.
- **Perspectiva de visualización:** Los Mapas Conceptuales pueden ser vistos como diagramas, contruidos mediante el uso de signos. Cada tipo de nodo puede determinar (o ser determinado) por la forma, por el color externo o por el llenado interior. Los enlaces pueden ser identificados por el grosor de la línea, el color u otras formas de representación.
- **Perspectiva de conversación:** Los Mapas Conceptuales pueden ser considerados como una forma de representación y comunicación del conocimiento a través de lenguajes visuales, porque están sujetos a interpretación por alguna comunidad de referencia. Esta interpretación permite el establecimiento de un paralelo entre el lenguaje natural y el lenguaje visual.

1.3.2 Tipos de Mapas Conceptuales. Aprendizaje y aplicaciones

Por la propia definición y la razón de ser de los Mapas Conceptuales, los de tipo jerárquico son los más usados y difundidos, pero éstos pueden diseñarse siguiendo otro criterio de organización.

Existen cuatro tipos principales de Mapa Conceptual:

- **Mapas Conceptuales en forma de araña:** El mapa es estructurado de manera que el término que representa al tema principal es ubicado en el centro del gráfico y el resto de los conceptos llegan mediante la correspondiente flecha.
- **Mapas Conceptuales jerárquicos:** La información se representa en orden descendente de importancia. El concepto más importante es situado en la parte superior del mapa.
- **Mapas Conceptuales secuenciales:** En este tipo de mapa los conceptos son colocados uno detrás del otro en forma lineal.
- **Mapas Conceptuales en sistemas:** En este tipo de mapa la información se organiza también de forma secuencial pero se le adicionan entradas y salidas que alimentan los diferentes conceptos incluidos en el mapa.

En (Ausubel, 1989) y (Novak and Gowin, 1988) se formula una teoría de aprendizaje que ha resultado ser un gran aporte para el perfeccionamiento de la educación. La idea principal en la teoría de Ausubel es que el aprendizaje de nuevos conocimientos depende de la medida de lo que ya se conoce. En otras palabras, la construcción del conocimiento comienza con nuestra observación y reconocimiento de eventos y objetos a través de conceptos que ya poseemos. En general, nosotros aprendemos construyendo una red de conceptos.

Otro elemento importante en la teoría de Ausubel, es el enfoque de aprendizaje significativo. Para aprender significativamente los individuos deben relacionar nuevos conocimientos con conceptos relevantes que ellos ya conocen (a diferencia del aprendizaje por memorización). El nuevo conocimiento debe interactuar con la estructura del conocimiento ya aprendida. El

aprendizaje significativo puede ser contrastado con aprendizaje por memorización el que también puede incorporar nueva información a la estructura del conocimiento, pero sin interacción. El aprendizaje memorístico es débil para recordar secuencias de objetos, pero no ayuda con lo aprendido a entender las relaciones entre los objetos. El aprendizaje significativo, entonces, es personal, idiosincrásico, e involucra el reconocimiento de relaciones entre conceptos.

La principal aplicación de los Mapas Conceptuales ha sido en la enseñanza. Ese es su propósito básico, pero las aplicaciones de un Mapa Conceptual son ilimitadas. En el caso de las ciencias de la educación, debe destacarse que los Mapas Conceptuales dirigen la atención, tanto del estudiante como del profesor, sobre el reducido número de ideas importantes en las que deben concentrarse en cualquier tarea específica de aprendizaje.

Un Mapa Conceptual también puede hacer las veces de mapa de ruta, donde se muestran algunos de los caminos que se pueden seguir para conectar los significados de los conceptos de forma que resultan proposiciones. Una vez que se ha completado una tarea de aprendizaje, los Mapas Conceptuales proporcionan un resumen esquemático de todo lo que se ha aprendido.

Dado que se alcanza más fácilmente un aprendizaje significativo cuando los nuevos conceptos o significados conceptuales se engloban bajo otros conceptos más amplios, más inclusivos, los Mapas Conceptuales más empleados son de tipo jerárquicos, es decir, los conceptos más generales e inclusivos se sitúan en la parte superior del mapa, y los conceptos progresivamente más específicos y menos inclusivos, en la inferior. Las relaciones subordinadas entre conceptos pueden cambiar en diferentes segmentos de aprendizaje, por lo que en un Mapa Conceptual, cualquier concepto puede elevarse a la posición superior, y seguir manteniendo todavía una relación proposicional significativa con otros conceptos del mapa.

Confeccionar un Mapa Conceptual constituye una técnica dirigida a revelar conceptos y proposiciones que reflejen el criterio de verdad sobre un determinado segmento de la realidad circundante. Hasta este momento, sólo se pueden hacer conjeturas sobre el grado de acierto

con que los Mapas Conceptuales representan los conceptos que se poseen, o la gama de relaciones entre conceptos que se conocen (y que se puede expresar como proposiciones). En el proceso de elaboración de los Mapas Conceptuales se pueden desarrollar nuevas relaciones conceptuales, en especial si, de una manera activa, se intenta construir relaciones proposicionales entre conceptos que previamente no se consideran relacionados.

El uso de Mapas Conceptuales diseñados por el profesor, incrementa tanto el aprendizaje como la retención de información científica. Los estudiantes producen mapas como herramientas de aprendizaje.

Teniendo en cuenta que los Mapas Conceptuales constituyen una representación explícita y manifiesta de los conceptos y proposiciones que posee una persona, ellos permiten que profesores y alumnos intercambien puntos de vista sobre la validez de un vínculo proposicional determinado, o darse cuenta de las conexiones que faltan entre los conceptos y que sugieren la necesidad de un nuevo aprendizaje. Por esta razón, esta técnica ha articulado tan favorablemente con los preceptos de la educación a distancia que supone que estudiantes y profesores no estén físicamente en el mismo lugar y al mismo tiempo.

Utilizando tanto la vía sincrónica como la asincrónica, se puede intercambiar desde lejos la interpretación que se tiene sobre un concepto, su definición, alcance y sus relaciones con otros conceptos en un contexto determinado. Esto ha estimulado el desarrollo de múltiples herramientas computacionales dirigidas a construir y compartir conocimiento, que tienen como soporte básico los Mapas Conceptuales y como vehículo por excelencia la red de redes.

La riqueza del conocimiento puede ser incrementada por el uso de Mapas Conceptuales. Diversos autores sugieren que los estudiantes que hacen o analizan Mapas Conceptuales tendrán un conocimiento básico amplio y, por lo tanto, estarán más preparados para resolver problemas en comparación con aquellos estudiantes que han aprendido por memorización.

Los Mapas Conceptuales se han convertido en un instrumento útil para la preparación de los docentes y para la comprensión por parte de los alumnos de temas de diferentes asignaturas.

Indiscutiblemente, este instrumento de aprendizaje ofrece ventajas en el desarrollo mismo del aprendizaje del estudiante. A continuación se exponen algunas de éstas:

- Ilustran la estructura cognoscitiva o de significados que tienen los individuos a partir de sus percepciones y experiencias personales.
- Permiten trabajar y corregir los errores conceptuales del estudiante y facilitar la conexión de la información con otros conceptos relevantes de la persona.
- Facilitan la organización lógica y estructurada de los contenidos de aprendizaje, ya que son útiles para separar la información significativa de la información trivial.
- Fomentan la cooperación entre los estudiantes.
- Contribuyen a planificar la instrucción y a la vez ayuda a los estudiantes a aprender a aprender.
- Favorecen la creatividad y la autonomía de los estudiantes.
- Facilitan la comprensión integral de los fenómenos al interrelacionar los conceptos, las ideas de los alumnos, y la estructura de la asignatura. En el caso de la Historia, por ejemplo, relacionan hechos pasados, presentes y futuros.
- Fomentan la negociación, al compartir y discutir significados. La confección de los MC en forma grupal, por ejemplo, desempeña una útil función social en el desarrollo del aprendizaje.
- Permiten relacionar las partes (el todo) unos con otros.

Entre los cuidados que se deben tener en cuenta a la hora de confeccionar un Mapa Conceptual, están los siguientes:

- Confundir un Mapa Conceptual con un esquema o diagrama de flujo y en lugar de presentar relaciones supraordenadas y combinatorias entre conceptos, se presenten simples secuencias lineales de acontecimientos.
- Evitar establecer relaciones excesivamente confusas entre conceptos.
- Garantizar que no se constituya en la única herramienta o técnica para construir aprendizaje, sino que sea parte de una secuencia más amplia, ordenada y sobre todo, significativa.

- Tener presente que la elaboración de los Mapas Conceptuales es un proceso que requiere tiempo. Los estudiantes necesitan practicar el pensamiento reflexivo.

En lo fundamental, los éxitos de los Mapas Conceptuales, en nuestra opinión, se basan en un instrumento que combinan el rigor científico con la sencillez y flexibilidad. Es una ayuda para el que genera, transmite, almacena, divulga información y conocimiento y constituyen una importante herramienta para lograr un valor práctico, sobre todo en los sistemas propios de la IA.

1.4 Conclusiones parciales

A partir de la consulta realizada a la bibliografía se pueden extraer las conclusiones fundamentales siguientes:

- Enseñar es algo que ha evolucionado junto al hombre a través de los años. La Enseñanza Asistida por Computadoras ha repercutido de forma extraordinaria en la sociedad, revolucionando los métodos tradicionales de enseñanza y elevando la calidad del proceso de formación del hombre.
- El uso de las técnicas de Inteligencia Artificial han facilitado el desarrollo cualitativo de los software educativos añadiéndole a los mismos la capacidad de adaptación al estudiante en una sesión de aprendizaje.
- El uso de los Mapas Conceptuales es sin duda una técnica efectiva para mejorar los métodos de enseñanza, además su aplicabilidad en la Informática Educativa ayuda a complementar la relación que debe existir entre tecnología y pedagogía.

CAPÍTULO II. IMPLEMENTACIÓN DE HESEI VERSIÓN 1.0

En este capítulo se abordan conceptos generales de la Programación Orientada a Objetos. Se explican elementos de la implementación de HESEI. Se describen algoritmos usados, además los modelos y técnicas utilizadas en la elaboración de la herramienta.

2.1 Principales conceptos en el diseño e implementación orientados a objetos

Es importante tratar algunos conceptos fundamentales de la Programación Orientada a Objetos (POO), así como aspectos sobre la metodología de análisis y diseño orientada a objetos UML utilizada durante las etapas de desarrollo del sistema (Rumbaugh et al., 1997a), (Rumbaugh et al., 1997b).

2.1.1 Conceptos generales de la Programación Orientada a Objetos

Los módulos que en este capítulo se describen son realizados utilizando el paradigma de POO. Es necesario precisar cuáles son las características particulares y las propiedades que son utilizadas en este diseño para identificar los objetos.

En (Rodríguez, 1999) se presentan las siguientes propiedades de los objetos:

- **Estado:** Se define a partir de los valores que en un momento dado tienen los atributos del objeto. La estructura del objeto se define como el conjunto de todos los atributos o propiedades. Además, un objeto puede conocer o contener a otros objetos, estas relaciones son también parte de su estado.
- **Comportamiento:** Define cómo actúan los objetos frente a estímulos externos en términos de cambio de estados.
- **Identidad:** Es la propiedad de un objeto que lo distingue del conjunto de todos los demás objetos del universo al que pertenece. Los modelos de POO son representaciones abstractas de este tipo.

El marco de referencia conceptual en un sistema orientado a objeto es el modelo de objetos que incluye cuatro conceptos fundamentales:

- **Abstracción:** Denota las características esenciales de un objeto que lo distinguen de todos los demás tipos de objeto y proporciona así fronteras conceptuales nítidamente definidas respecto a la perspectiva del observador.
- **Encapsulamiento:** Es uno de los principios más importantes de la POO, ha permitido la reusabilidad de objetos. Constituye el proceso de almacenar en un mismo compartimiento los elementos de una abstracción que constituyen su estructura y su comportamiento. El cliente se interesa por lo que hace el objeto y no cómo lo hace.
- **Modularidad:** Es la propiedad que tiene un sistema que ha sido descompuesto en un conjunto de módulos cohesivos y débilmente acoplados.
- **Jerarquía:** Es una clasificación u ordenamiento de abstracciones.

Otros conceptos importantes a tener en cuenta son:

- **Concurrencia:** Es la propiedad que distingue un objeto activo de uno que no está activo.
- **Persistencia:** Es la propiedad de un objeto por la que su existencia trasciende el tiempo, el espacio, o ambos.

En (Rodríguez, 1999) se definen los conceptos de clase y tipo:

“La clase no es más que una representación abstracta que define la estructura y el comportamiento que le son comunes a un grupo de objetos. Mientras que el tipo es un protocolo usado en los mecanismos de comunicación e interacción entre objetos. Tiene identidad y generalmente está más relacionado a los mecanismos de comunicación que a la propia naturaleza de los objetos.”

2.1.2 La metodología de análisis y diseño orientados a objetos

Las metodologías de análisis y diseño orientadas a objetos, como el resto de las metodologías de la ingeniería de software tratan de establecer pautas para el desarrollo de sistemas; indican

los pasos a seguir durante las diferentes etapas por las que transita un software en su construcción. Generalmente en las metodologías existen tres etapas bien definidas que son: la etapa de análisis, la etapa de diseño y la de implementación (Kendall and Kendall, 1997).

Según Grady Booch (Booch et al., 1999), (Booch, 1991):

“El análisis orientado a objetos es un método de análisis que examina los requisitos desde la perspectiva de las clases y objetos que se encuentran en el vocabulario del dominio del problema”.

“El diseño orientado a objetos es un método de diseño que abarca el proceso de descomposición orientado a objetos y una notación para describir los modelos lógico y físico, así como los modelos estático y dinámico del sistema que se diseña”.

“En el análisis de un problema se deben examinar los roles que un objeto puede jugar. Durante la etapa de diseño se refinan estos roles inventando las operaciones que llevan a cabo las responsabilidades de cada rol”.

Al definir las clases es necesario tener en cuentas las operaciones que permiten representar el comportamiento de una clase y los atributos que representan su estructura.

No sólo es necesario conocer como se definen correctamente los objetos, sino qué relaciones se pueden establecer entre ellos. En (Booch et al., 1999) se definen tres tipos de relaciones entre objetos:

- **Asociación:** Es una conexión bidireccional entre clases. Se muestra en el diagrama una línea conectando las clases relacionadas.
- **Agregación:** Es una relación existente entre una pieza y sus partes. Una agregación se representa a través de una línea que conecta las clases relacionadas con un diamante en el extremo de la clase que representa a la clase agregada.

- **Dependencia:** Es una forma débil de relación entre un cliente y un proveedor cuando el cliente no tiene un conocimiento semántico del proveedor. Una dependencia se representa como una línea discontinua desde el cliente hasta el suministrador.

La herencia se ha usado en el diseño de la herramienta y no es más que la relación entre una superclase y sus subclases. Existen dos formas de encontrar herencia: generalización y especialización. Así, los atributos comunes, operaciones y relaciones son mostrados en las clases de mayor nivel en la jerarquía.

Un elemento que facilita la organización del diseño en UML son los paquetes. Un paquete es un agrupamiento de elementos del modelo. Los paquetes en sí mismos pueden estar compuestos por otros paquetes. Un paquete puede contener paquetes subordinados y elementos ordinarios del modelo.

2.2 Generalidades de HESEI

HESEI utiliza técnicas de IA y Mapas Conceptuales, con el objetivo de adaptar con mayor precisión el sistema de enseñanza/aprendizaje a las características del alumno. Esta herramienta facilita el diseño e implementación de software educativos destinados a cualquier nivel de enseñanza.

Sus fundamentos teóricos se apoyan en la ingeniería del conocimiento, las técnicas de IA (véase figura 2.1), así como la ciencia pedagógica. Por medio de entrevistas y sesiones realizadas con los profesores más experimentados se recopilan conocimientos valiosos que se pueden formalizar y codificar.

Los sistemas (software educativos) que se crean con HESEI se denominan sistemas de enseñanza/aprendizaje inteligentes, donde el término “inteligente” está dado por la capacidad de adaptación a las características cognitivas del estudiante (Modelo del Estudiante) y su implementación se logra haciendo uso del paradigma del RBC.

Esta técnica intenta llegar a la solución de nuevos problemas de forma similar a como lo hacen los seres humanos, cuando un individuo enfrenta un nuevo problema comienza por buscar en su memoria experiencias anteriores similar al actual y a partir de ese momento establece semejanzas y diferencias y combina las soluciones dadas con anterioridad para obtener una nueva solución.

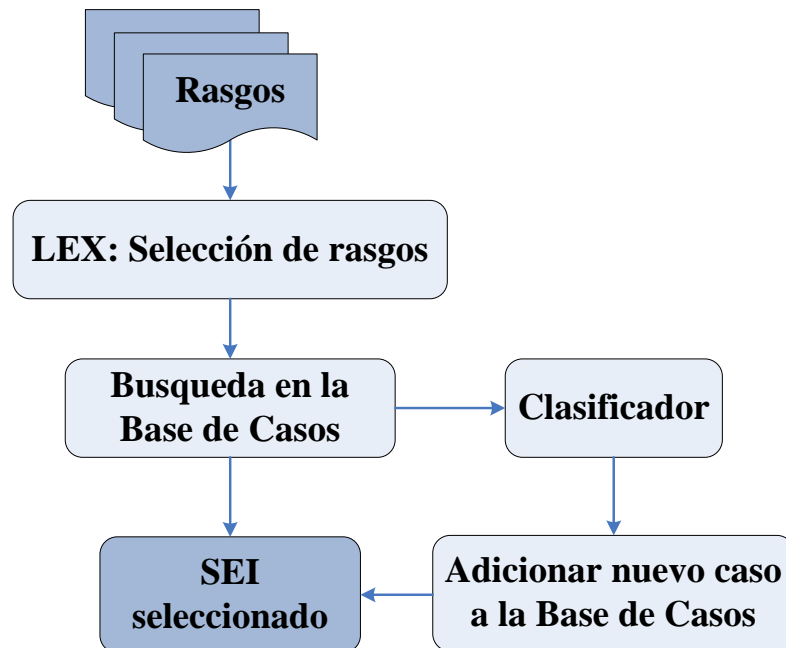


Figura 2.1. Inteligencia Artificial en HESEI

El SEI es un producto tecnológico diseñado para apoyar el proceso educativo, dentro del cual se tiene en cuenta qué se enseña y quién aprende, para alcanzar determinados propósitos. Además, este software es un medio de presentación y desarrollo de contenidos educativos, como lo puede ser un libro o un video, con su propio sistema de códigos, formato expresivo y secuencia narrativa. De esta manera, puede ser visto como un producto y también como un medio.

En la figura 2.2 se muestra la arquitectura de HESEI, que incluye RBC, y un algoritmo de Reconocimiento de Patrones para lograr la implementación del Modelo del Estudiante con una previa selección de rasgos.

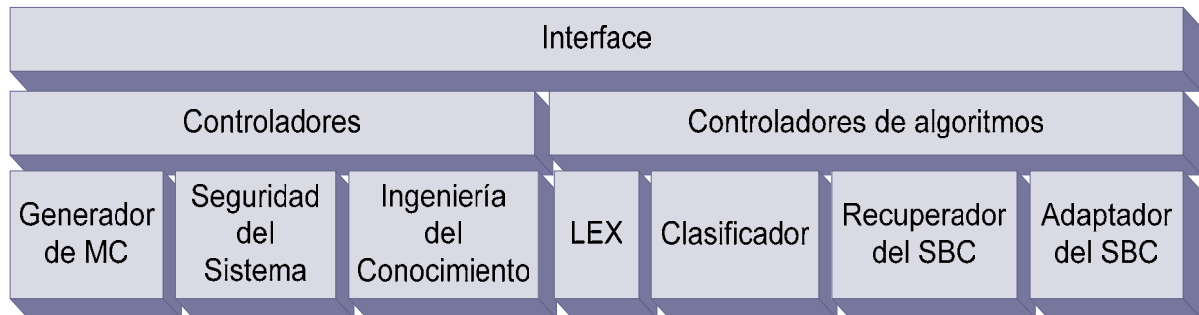


Figura 2.2. Arquitectura de HESEI

Breve descripción de algunas componentes que conforman la arquitectura del sistema:

- **Interfaz:** La interfaz cuenta con un editor, el cual le permite al profesor introducir toda la información necesaria para preparar los sistemas de enseñanza/aprendizaje y editar las preguntas. Además a través de esta componente el estudiante podrá interactuar con el SEI elaborado en forma de Mapa Conceptual adaptado a él según sus características.
- **Seguridad del sistema:** Es el encargado de identificar el tipo de usuario que interactúa con la herramienta.
- **Ingeniería del Conocimiento:** Esta componente capta todo el conocimiento que el profesor desea tener en cuenta en su SEI.
- **LEX:** Está estrechamente relacionada con la componente Ingeniería del Conocimiento, se utiliza un algoritmo (LEX) de Reconocimiento de Patrones para reducir el espacio de representación inicial (rasgos que conforman el Modelo del Estudiante).
- **Clasificador:** Es una estructura de datos utilizada para asignarle a cada estudiante el entrenador adecuado en cada tópico para conformar su SEI.
- **Recuperador y adaptador del SBC:** En esta componente se implementó un Sistema Basado en Casos compuesto por la base de casos y los algoritmos de recuperación y adaptación de casos.

Gran parte del éxito de cualquier Sistema Basado en el Conocimiento (Bello, 2002b) es que se realice una buena ingeniería del conocimiento teniendo en cuenta que el usuario no es experto en informática. La herramienta utiliza técnicas de Reconocimiento de Patrones para ayudar al usuario profesor en esta tarea.

HESEI permite al profesor estructurar el sistema de enseñanza/aprendizaje o tema, en tópicos. Estos a su vez estarán divididos por preguntas, cada tópico tiene tantas preguntas como se desee, véase figura 2.3.

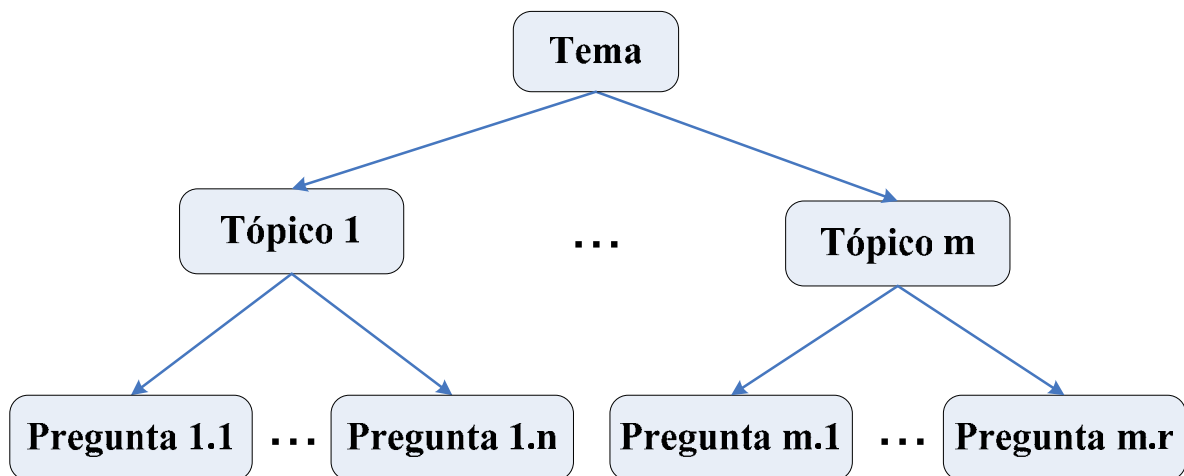


Figura 2.3. Estructura de un Sistema de Enseñanza/Aprendizaje elaborado con HESEI

Si el número de preguntas es n la cantidad de combinaciones para la evaluación de cada tópico es 2^n ya que cada pregunta se evalúa de bien o mal (dos valores).

Cuando el profesor introduce el número de preguntas para evaluar el tópico a través de la interfaz de HESEI, se muestra la tabla de combinaciones para que él indique la evaluación que decidió (véase tabla 2.1).

Por ejemplo, si se tiene que el Tópico 1 está conformado por 3 preguntas, se tendrán 8 combinaciones, ya que 2^3 es 8. Si el profesor decide evaluar de bien sólo las combinaciones que tienen evaluaciones de bien en las preguntas 1 y 3, según muestra la tabla 2.1, se está en

presencia de un clásico ejemplo en el cual la pregunta 2 no está teniendo peso en la evaluación.

P1	P2	P3	Evaluación
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	<u>0</u>	1	<u>1</u>
1	1	0	0
1	<u>1</u>	1	<u>1</u>

Tabla 2.1. Ejemplo de una pregunta (P2) sin peso en la evaluación

Tomando las preguntas como rasgos y la evaluación como clase, la herramienta haciendo uso de un algoritmo de reconocimiento de patrones, LEX (Pons and Santiesteban, 2003), calcula los testores típicos, es decir, las preguntas que están tomando peso en la evaluación, el profesor tiene la posibilidad de eliminar a aquellas que no toman parte en la evaluación.

Este proceso se realiza con cada uno de los tópicos. Seguidamente la herramienta muestra la tabla resultante y en una columna adicional se introduce, teniendo en cuenta la evaluación del tópico, el entrenador que se utilizará.

Como mínimo pueden ser dos entrenadores, uno para cuando el tópico es evaluado de bien y otro en el caso de mal.

Terminado este paso, el sistema ofrece al profesor un editor con todas las facilidades para introducir el cuestionario y los entrenadores correspondientes a cada tópico.

Con toda esta información y haciendo uso del RBC, un estudiante al responder las preguntas de cada tópico tendrá asociado, según su comportamiento y conocimiento, un SEI apropiado a él.

Los rasgos predictores son las repuestas del cuestionario que realiza el estudiante, conformándose de esta forma un caso.

El algoritmo de recuperación, que se propone, hace una búsqueda exhaustiva, en la base de caso, comparando el nuevo caso con todos los almacenados en la misma, usando la función de similitud propuesta por (Gutiérrez, 2003), si la búsqueda no es exitosa, se evalúa en el Clasificador (véase figura 2.4) tomando los valores de los rasgos predictores del nuevo caso como entradas y como salida se obtiene el estereotipo de entrenador (rasgo objetivo), almacenando el nuevo caso a la base de casos.

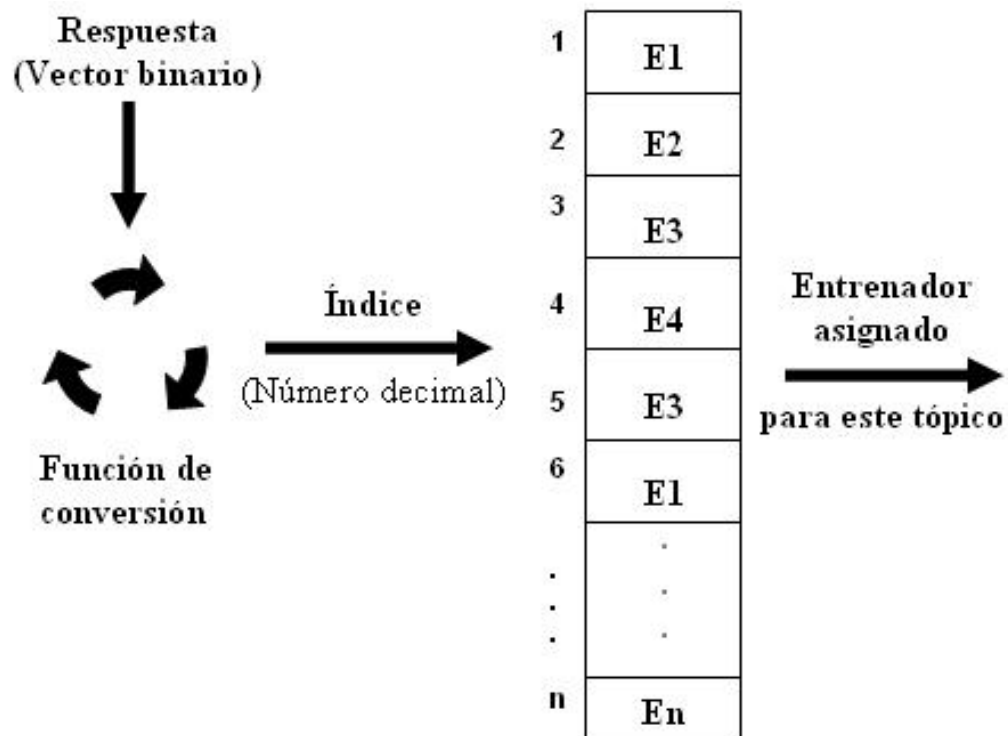


Figura 2.4. Clasificador

Los usuarios de HESEI son Estudiante y Profesor, actores fundamentales en el proceso de enseñanza/aprendizaje. Éstos presentan los casos de uso propios según la función que les toca representar en ese proceso (véase Anexos 1 y 2). A través del diagrama de transición de estado mostrado en el Anexo 3 se representan los diferentes estados por los que el estudiante transita en la aplicación.

En el diagrama de actividades presentado en el Anexo 4 se observa la secuencia de acciones realizadas por los actores del sistema en la creación e interacción con un sistema de enseñanza/aprendizaje inteligente.

2.3 Implementaciones de la interfaz de usuario

Delphi 7 es la versión 7.0 del popular compilador desarrollado por Borland Software Corporation y ofrece, a los desarrolladores de aplicaciones, la posibilidad de crear software basándose principalmente en la POO y con elementos de programación estructurada, que finalmente forman parte del lenguaje Object Pascal. Cuenta con herramientas muy útiles en la edición de recursos para aplicaciones visuales, todas incluidas dentro de su Visual Component Library (VCL).

Este compilador posee además gran variedad de estructuras de datos definidas que pueden ser usadas concisa y racionalmente para resolver cualquier tipo de problema de programación de una manera más rápida y eficiente.

Todos los módulos de HESEI se han implementado en esta potente herramienta de desarrollo de software haciendo uso y aprovechando todas las facilidades de este ambiente de programación.

2.3.1 Editor de preguntas

Se cuenta con un editor de preguntas, es una herramienta de fácil manejo elaborar, este editor facilita la confección de preguntas de forma dinámica, además permite editarlas después de creadas y resueltas.

El profesor puede elaborar preguntas pertenecientes a un tópico determinado, antes de editar se conocerá la cantidad de tópicos y la cantidad de preguntas de cada tópico. Posteriormente se elige el tipo de pregunta. Toda esa información debe ser almacenada ya que el estudiante deberá responder las preguntas que el profesor editó al cargar el tema seleccionado.

El editor permite el trabajo sobre tres tipos de preguntas:

- Verdadero o falso.
- Marcar la correcta.
- Relacionar columnas.

Luego que el profesor edite las preguntas, éstas queden almacenadas, se guardará la respuesta para cuando el estudiante responda, se chequea si la respuesta es correcta (1) o es incorrecta (0), primeramente el profesor debe introducir las respuestas para que luego el sistema pueda verificar si coinciden con las del estudiante.

2.3.2 Generación de un sistema de enseñanza/aprendizaje inteligente a través de un Mapa Conceptual

Por cada SEI se generará un Mapa Conceptual; el nodo inicial será el nombre de la temática, luego tendrá tantos hijos como tópicos hayan en el tema y cada tópico tendrá tantos hijos como entrenadores o materiales de estudio tenga el tópico. Al mostrarse el Mapa Conceptual deberán estar habilitados los entrenadores destinados a ese caso, así como los de un nivel de complejidad inferior, haciendo distinción en el entrenador que le corresponde y en los que

están accesibles. De esta forma quedarán desactivados aquellos entrenadores de complejidades superiores.

Datos necesarios para generar un Mapa Conceptual:

- Nombre del sistema de enseñanza/aprendizaje.
- Cantidad de tópicos y su nombre.
- Cantidad de preguntas de cada tópico.
- Entrenadores de cada tópico y su objetivo fundamental.

2.3.3 Uso de la componente Vclskin para la selección del estilo de la interfaz

A través de la componente Vclskin, disponible para Delphi, se implementó la opción de cambiar el estilo de ventana de la aplicación, de manera tal que el usuario use la configuración que desee.

A menudo se interactúa con software que tienen interfaces estáticas, totalmente predeterminadas y se desea, en múltiples ocasiones lograr una mayor afectividad en la interfaz.

Si se está diseñando un software educativo para niños de escuelas primarias será de suma importancia llamar su atención a través de colores vivos, botones y componentes grandes, que resalten a la vista, sin embargo si el software educativo es para niveles superiores de la educación, tales como la enseñanza preuniversitaria o la propia universidad, una interfaz discreta, sencilla y agradable será la apropiada, es por eso que la herramienta permite cambiar el estilo de configuración del ambiente del programa.

Se cuenta con una colección de ficheros de configuración de estilos llamados skins, disponibles gratuitamente en Internet, de los cuales se selecciona el más adecuado según el interés del software y a quién esté destinado.

Si el usuario desea incorporar algún nuevo estilo, solamente debe copiarlo hacia la carpeta que contiene los demás estilos y posteriormente seleccionarlo para que la herramienta tome esa configuración en su interfaz visual.

Los investigadores de la rama Informática Educativa están llamados a lograr niveles superiores en el proceso enseñanza/aprendizaje, tanto por los profesores como por los estudiantes que interactúan con el software, pero no sólo se puede depender del nivel cognitivo, si no también interactuar con procesos afectivos que despierten mayores motivaciones, ya que un SEI emocionalmente activo debe ser capaz de dar sentido a la información afectiva y poseer la capacidad de responder a esta información en términos emocionalmente apropiados. Afectividad y cognición deben ir de la mano en el proceso de enseñar y aprender.

2.3.4 Autenticación de usuarios mediante LDAP

LDAP (“Lightweight Directory Access Protocol”, “Protocolo Ligerero de Acceso a Directorios”) es un protocolo de red que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse, aunque es posible almacenar otra información como:

- Datos de contacto del usuario.
- Ubicación de diversos recursos de la red.
- Permisos y certificados.

El servicio de directorio LDAP se basa en un modelo cliente-servidor. Uno o más servidores LDAP contienen los datos que conforman el árbol del directorio LDAP. El cliente LDAP se conecta con el servidor LDAP y le hace una consulta.

El servidor contesta con la respuesta correspondiente, o bien con una indicación de dónde puede el cliente hallar más información (normalmente otro servidor LDAP).

No importa con qué servidor LDAP se conecte el cliente, siempre observará la misma vista del directorio; el nombre que se le presenta a un servidor LDAP hace referencia a la misma entrada a la que haría referencia en otro servidor LDAP (Pinheiro, 2000).

El servidor al cual se conecta es Active Directory, este nombre es utilizado por Microsoft para referirse a su implementación de seguridad en una red distribuida de computadoras. Utiliza distintos protocolos, principalmente: protocolo LDAP, DNS y DHCP.

Como cualquier otro LDAP, en Active Directory este protocolo viene usado como una base de datos que almacena de forma centralizada toda la información de un dominio de administración, con la ventaja de mantener esta información sincronizada entre distintos servidores de autenticación de acceso a la red.

Se utilizan cuatro funciones de API de Windows para lograr la autenticación mediante LDAP:

- **Ldap_init:** Esta función inicializa una sesión con el servidor LDAP. Se le pasa como parámetros el servidor y el puerto. Al ponérsele nil en el valor para el servidor, se toma el servidor por defecto del dominio al que pertenece la máquina. El puerto por defecto es el 389.
- **Ldap_connect:** Esta función establece una conexión con el servidor LDAP.
- **Ldap_bind_s:** Esta función autentifica un cliente al servidor LDAP. Entre otros parámetros, se le pasan: la estructura de autenticación, la cual contiene los datos del usuario y el tipo de la autenticación, el cual está definido por varias banderas, ejemplo de ellas son: LDAP_AUTH_OTHERKIND y LDAP_AUTH_NTLM (esta última es la autenticación para Microsoft Windows NT y sus versiones posteriores).
- **Ldap_unbind:** Esta función libera todos los recursos asociados con la sesión LDAP.

2.4 Uso de las técnicas de Inteligencia Artificial en HESEI

No existe aún ninguna definición única y rigurosa que cubra adecuadamente todos los aspectos que el término “Inteligencia Artificial” representa. Se enuncian dos definiciones:

Según Luger y Stubblefield, IA es:

“La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente” (Bello, 2002).

Según Kurzweil, IA es:

“El arte de crear máquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia” (Bello, 2002).

Utilizar algunas técnicas de IA en la herramienta añade potencialidades que permiten considerar los sistemas de enseñanza/aprendizaje elaborados con HESEI como inteligentes.

2.4.1 Reducción del espacio inicial aplicando técnicas de Reconocimiento de Patrones

El Reconocimiento de Patrones se preocupa por detectar regularidades entre conjuntos de datos, que son difíciles de hallar con algoritmos tradicionales. Es un área con una larga historia a lo largo de la cuál los investigadores han propuesto modelos desde enfoques muy variados. Sigue siendo un problema de investigación muy activo y con aplicaciones en infinidad de problemas.

2.4.1.1 LEX. Un nuevo algoritmo para los testores típicos

Uno de los problemas del Reconocimiento de Patrones es la selección de variables, que se utiliza para reducir de modo eficiente el número de variables (atributos o rasgos) con los cuales se deben describir los objetos y para encontrar los rasgos que inciden de manera

determinante en un problema. Una alternativa de solución al problema de la selección de variables es a partir de la utilización del conjunto de testores típicos.

Un testor es un conjunto de características (rasgos) que diferencia a elementos (objetos) de clases distintas. Un testor típico es un testor al que si se le elimina cualquiera de sus rasgos pierde la propiedad de ser testor (Pons and Santiesteban, 2003). Dicho de otro modo, un testor típico es un subconjunto minimal de rasgos que permite diferenciar objetos de clases distintas, así si t es testor, para todo t' subconjunto de t y t' no testor, t es típico (Shulcloper et al., 1995).

Para el cálculo de los testores típicos se han desarrollado varios algoritmos que por su estrategia de cómputo pueden clasificarse en algoritmos de escala exterior y de escala interior. Los algoritmos de escala exterior son aquellos que realizan el cálculo de los testores típicos generando los elementos del conjunto potencia del conjunto de columnas de la matriz básica en un determinado orden, de tal forma que, usando determinados criterios, el algoritmo trata de evitar el análisis de todos los subconjuntos. Por el contrario, los algoritmos de escala interior realizan el cálculo de los testores típicos basados en el estudio de la estructura interna de la matriz, encontrando condiciones que garantizan la característica de testor y la tipicidad en las columnas asociadas a las posiciones unitarias de la matriz.

Ejemplos de algoritmos de escala interior son CC (Águila and Shulcloper, 1984) y CT (Bravo, 1983) y de escala exterior BT (Shulcloper, 1982), TB (Shulcloper, 1982), CER (Ayaquica, 1997) y LEX (Pons and Santiesteban, 2003).

El algoritmo LEX tiene como idea fundamental ir construyendo listas de rasgos que posean la propiedad de tipicidad y luego, comprobar si este conjunto de rasgos constituye un testor. Cuando se obtiene un testor típico se producen saltos.

El algoritmo comienza analizando el primer rasgo de la matriz básica. Un nuevo rasgo se puede incorporar a la lista si se cumple que no es excluyente con la lista (puede coexistir con los rasgos de la lista para formar un testor típico y tiene filas típicas con respecto a la lista). Esta condición es la que garantiza la propiedad de tipicidad.

Luego, se comprueba si el conjunto de rasgos contenido en la lista junto con el nuevo rasgo forma un testor, verificando si no existe fila en la matriz básica (considerando sólo esos rasgos) completa de ceros. Si se cumple que es testor, entonces estamos en presencia de un testor típico y se almacena.

Si se encontró un testor típico y éste contiene al último rasgo de la matriz básica, entonces se saltan todos sus subconjuntos consecutivos. Si no contiene al último rasgo de la matriz básica, para saltar todos los supraconjuntos del testor típico encontrado se elimina el último rasgo de la lista y se analiza si se puede incluir el próximo rasgo de la matriz básica.

Cuando un rasgo es aceptado como nueva componente de la lista actual l se actualizan adecuadamente los cr_i^l para todas las filas de la matriz básica y los conjuntos de filas típicas de todos los rasgos de l .

Si el rasgo analizado no se puede incorporar a la lista se prosigue el análisis con el siguiente rasgo de la matriz básica.

Con el objetivo de lograr un mejor desempeño del algoritmo se realiza previamente un ordenamiento de la matriz básica que consiste en colocar como primera fila aquella que tenga la menor cantidad de unos y, en esta fila encontrada, situar todos los unos a la izquierda (lo cual implica, por supuesto, un reordenamiento de las columnas correspondientes a cada posición unitaria).

Si existiera más de una fila con mínima cantidad de unos se toma cualquiera de ellas. De esta forma, el algoritmo puede terminar cuando, durante el proceso de retroceso, la lista actual esté vacía y el rasgo que le corresponda analizarse posea un cero en la primera fila, ya que ninguno de los posibles subconjuntos de rasgos que se pueden formar con los rasgos que restan pueden formar un testor típico por no cumplir la propiedad de ser testor (la primera fila de la matriz básica es completa de ceros en estas columnas).

Algoritmo LEX (Pons and Santiesteban, 2003):

Entrada: Matriz básica.

Salida: El conjunto de todos los testores típicos de la matriz básica.

1. Ordenamiento de la matriz básica

- a. Encontrar la fila con cantidad mínima de unos; de existir más de una, escoger cualquiera de ellas. Ponerla como primera fila en la matriz básica.
- b. Ordenar las columnas poniendo indistintamente como primeras las que tengan un 1 en la primera fila.

2. Inicialización

- a. $l = []$, $X = X_1$ (X es el primer candidato a elemento de l).

3. Evaluación del candidato X

- a. Si $l = []$ y la columna correspondiente a X tiene cero en la primera fila de la matriz básica entonces FIN.
- b. Si X es excluyente con l entonces ir al paso 4; no se acepta el candidato.
- c. Si para toda fila i de la matriz básica $cr_i^{l+[X]} > 0$, entonces guardar $l + [X]$; es un testor típico. Ir al paso 4.
- d. Si $X = X_n$, entonces ir al paso 4.
- e. Hacer $l = l + [X]$; se acepta el candidato. Actualizar los valores de cr_i^l para todas las filas de la matriz básica y F_x^l para todos los elementos X_i en l .

4. Selección del nuevo candidato

- a. Si $X \neq X_n$, entonces sea j el índice de X en la matriz básica, hacer $X = X_{j+1}$ e ir al paso 3.
- b. Si $l = []$ entonces FIN.

c. Si $l + [X]$ fue un testor típico o X no fue excluyente con l , entonces si existe el hueco X_p de $l + [X]$, entonces hacer $X = X_{p+1}$ y eliminar de l todos los elementos desde X_p hasta X_{j_s} , actualizando cr_i^l para todas las filas de la matriz básica y $F_{X_t}^l$ para cada X_t de l . Ir al paso 3.

De no existir X_p entonces FIN.

d. Hacer $X = X_{j_s}$, donde X_{j_s} es el último rasgo de l y $l = l \setminus [X_{j_s}]$. Actualizar cr_i^l para todas las filas de la matriz básica y $F_{X_t}^l$ para cada X_t de l .

Retornar al paso 4.

Con el número de rasgos predictores se forma una tabla de 2^n combinaciones, donde n es un número que puede ser considerablemente grande. La solución de este problema implica reducir el espacio de representación inicial, de forma tal que si existen rasgos predictores superfluos, se analice si se mantienen o no, según su importancia desde el punto de vista metodológico. Este proceso se aplica a cada uno de los tópicos del tema.

Para el cálculo de los testores típicos se requiere de una matriz dividida en clases. Tomamos como matriz la tabla de 2^n combinaciones del tópico. Y como clases los entrenadores concebidos según la evaluación del tópico.

Aplicando el algoritmo LEX (Pons and Santiesteban, 2003), se obtiene la tabla de casos cuyos rasgos predictores coinciden con los testores típicos y el rasgo objetivo con el entrenador asociado a los valores de los rasgos predictores.

2.4.2 Implementación del Modelo del Estudiante usando Razonamiento Basado en Casos

Dentro de las múltiples herramientas con que hoy cuenta la IA para su desempeño se encuentra el RBC, el cual es útil, dada su flexibilidad y la semejanza de su funcionamiento con el de los humanos.

El RBC representa un nuevo método para resolver problemas no estructurados en el cual el razonamiento se realiza a partir de una memoria asociativa que usa un algoritmo para determinar una medida de semejanza entre dos objetos. En este paradigma la base del comportamiento inteligente de un sistema radica en recordar situaciones similares existentes en el pasado. Debe destacarse que es una técnica en la cual la memoria se sitúa como fundamento de la IA y más concretamente de los sistemas basados en el conocimiento.

El RBC denota un método en el cual la solución de un nuevo problema se realiza a partir de las soluciones conocidas para un conjunto de problemas previamente resueltos (o no resueltos) del dominio de aplicación. Este método se distingue por el hecho de utilizar directamente la información almacenada en la memoria del sistema sobre los problemas o casos. Los sistemas que emplean el RBC usan una memoria permanente en lugar de alguna forma de base de conocimientos en la cual se almacene de forma explícita el conocimiento sobre el dominio de aplicación en forma de estructuras conceptuales, reglas de producción u otra forma de representación del conocimiento.

RBC significa razonar en base a experiencias o “casos” previos. Es una alternativa entre otras metodologías para construir sistemas basados en el conocimiento. Al razonar basado en casos el solucionador de problemas recuerda situaciones previas similares a la actual y las usa para ayudar a resolver el nuevo problema.

La principal idea del RBC tomada de (Aamodt and Plaza, 1994), es la de resolver un nuevo problema recuperando situaciones previas similares y reutilizando la información y el

conocimiento de dichas situaciones. Encontrando un caso pasado similar y reutilizando su solución en la situación del nuevo problema se soluciona un nuevo caso.

Según (Bello, 1996a) hacer cualquier inferencia basada en casos incluye los pasos siguientes:

- Presentar como entrada al sistema una descripción del problema a resolver o problema actual (Presentación).
- Encontrar aquellos casos en memoria que resuelven problemas similares al actual (Recuperación).
- Determinar las partes del viejo caso que se deben focalizar para resolver el subproblema actual.
- Derivar una solución adaptando la solución previa a las restricciones del nuevo problema (Adaptación).
- Chequear la consistencia del valor derivado con la descripción del problema a resolver, y aceptar o rechazar este (Validación).
- Añadir a la base de casos, si se considera apropiada la solución validada, para ser usada en el futuro si se considera conveniente (Actualización).

Encontrar los casos relevantes comprende según (Bello, 1996a):

- Caracterizar el problema de entrada asignando los rasgos apropiados a este.
- Recuperar los casos desde la memoria con esos rasgos.
- Seleccionar el caso o los casos que más se parecen a la descripción dada.

El módulo de recuperación tiene gran importancia en el Sistema Basado en Casos, ya que asegura que los casos más relevantes para un problema dado sean recuperados. Para esto el sistema recupera desde la memoria de los problemas ya resueltos almacenados en la base de casos los que son relevantes (muy parecidos) para el nuevo caso, obteniéndose como resultado la secuencia de operaciones y pasos. La selección de los casos se logra mediante una función de semejanza, la cual permite que no tenga que partir desde el inicio para crear una nueva tecnología (Morell, 2002).

La función de semejanza implementada es:

$$\beta(O, Ot) = \frac{\sum_{i=1}^n w_{ji} * \delta_i(O, Ot) * |1 - |\varepsilon_i(O) - \varepsilon_i(Ot)||}{\sum_{i=1}^n w_{ji}}$$

$$\delta_i(O, Ot) = \begin{cases} 1 & \text{si son iguales} \\ 0 & \text{en otro caso} \end{cases}$$

Donde:

w_{ji} , peso del rasgo j del tema i,

e_i , grado de certeza de la evaluación de la función de boole.

En la actualidad se desarrollan los sistemas híbridos en los cuales se combinan los sistemas basados en el conocimiento con:

- Hipertextos.
- Redes neuronales.
- Algoritmos genéticos.
- Multimedia.
- Realidad virtual.

El RBC permite al sistema híbrido manipular casos y realizar un razonamiento ancho y poco profundo. Según Steve Mott, presidente de la compañía Cognitive Systems, mediante las memorias o casos se pueden modelar dominios completos si se agrupan los casos suficientes para cubrir todos los problemas del dominio.

Para Ralph Barletta, vicepresidente de tecnología de software de esa compañía, un buen enfoque es modelar el dominio con reglas tanto como sea posible y utilizar el RBC para manipular las excepciones o problemas en el límite de decisión del sistema basado en el conocimiento que se desarrolla (Liebowitz, 1993).

El RBC constituye un nuevo método de solución de problemas para el desarrollo de sistemas basados en el conocimiento, el cual tiene las ventajas y desventajas siguientes según (Bello, 1996a):

Ventajas:

- El esfuerzo en la solución de problemas puede ser capturado para ahorrar trabajo en el futuro.
- Experiencias previas que hayan sido exitosas pueden ser utilizadas para justificar nuevas soluciones.
- Experiencias previas que no hayan sido exitosas pueden ser utilizar para anticipar problemas.
- La comunicación entre el sistema y los expertos se realiza en base a ejemplos concretos, es decir, el sistema explica sus decisiones citando precedentes.
- El RBC trabaja a partir de bases de datos existentes, no se requieren entrevistas con los expertos, simplificándose la adquisición del conocimiento.
- El RBC es un algoritmo de aprendizaje incremental, el aprendizaje tiene lugar tan pronto como un nuevo ejemplo está disponible, sin excesivo costo computacional.
- El RBC permite proponer soluciones para los problemas rápidamente, evitando el tiempo necesario para derivar respuestas desde el estado inicial de un proceso de búsqueda de soluciones.
- El RBC permite proponer soluciones en dominios que no se comprenden completamente.
- Los casos ayudan a focalizar el razonamiento sobre las partes importantes de un problema señalando qué rasgos del problema son importantes.
- El RBC es aplicable a un amplio rango de problemas.

Desventajas:

- El sistema no explora todo el espacio de soluciones, por lo que no puede encontrar soluciones óptimas.
- Requiere de una base de datos considerablemente grande y bien seleccionada.
- La consistencia entre varios casos es difícil de mantener.
- El RBC depende de una adecuada función de semejanza la cual no es fácil de encontrar para cada aplicación.

2.4.2.1 Arquitectura y aprendizaje de un Sistema Basado en Casos

En (Bello, 1996b) se plantea que un Sistema Basado en Casos tiene dos componentes principales: una base de casos y un resolutor de problemas. La base de casos contiene las descripciones de los problemas resueltos o no previamente. Cada caso puede describir un episodio particular o una generalización de un conjunto de episodios relacionados.

El resolutor de problemas tiene a su vez dos componentes: un recuperador de casos y un razonador sobre el problema; la principal diferencia entre la arquitectura de los dos tipos de Sistemas Basado en Casos radica precisamente en esta última componente.

En el tipo solucionador de problemas el proceso que se ejecuta es “recordar un caso y adaptar su solución” y en el interpretativo el proceso es “recordar un caso y evaluar el problema nuevo basado en su solución”; por lo que en dependencia del tipo el razonador sobre el problema contendrá un fuerte algoritmo de adaptación o un procedimiento de justificación. Luego de este paso intermedio de adaptación o justificación, se realiza un proceso de crítica y evaluación de la solución sobre la base del resultado, lo cual producirá la solución final o desencadenará una reiteración del proceso (Bello, 1996b).

Las posibilidades de aprendizaje de los Sistemas Basado en Casos descansan en la acumulación de nuevos casos y el perfeccionamiento de la organización de la base de casos.

Las alternativas para lograr esto son las siguientes:

- **Acumulación de nuevos casos:** Convirtiendo base de datos en parte de la base de casos. Almacenando los nuevos problemas que resulten de interés como casos de la base. Tomando problema resueltos por otros sistemas como nuevos casos.
- **Generación automática de casos:** Mediante el razonamiento por analogía, el aprendizaje basado en explicación y usando algoritmos genéticos para construir casos hipotéticos.
- **Reordenamiento de la base de casos:** El reordenamiento de la base de casos permite un mejor acceso y recuperación de los casos, disminuyendo el tiempo de acceso y posiblemente reduciendo la cantidad de información duplicada. Esto se puede realizar mediante el perfeccionamiento del sistema de índices y realizando generalizaciones.

2.4.2.2 La base de casos

El conjunto de problemas resueltos (que pudieron ser resueltos con éxito o no) forma la base de casos o memoria permanente del sistema.

En la memoria permanente se registran los problemas (casos) resueltos y debe poseer una organización que le permita mostrar cualidades similares a la memoria humana, es decir:

- Ser ilimitada.
- En la medida que la memoria crezca no se puede hacer más lenta.
- Debe permitir buscar directamente los elementos de memoria que sean relevantes a un problema.

El diseño de la base de casos incluye la compilación del vocabulario de términos usados para describir los rasgos del problema, la selección de los rasgos apropiados para indexar los casos y la especificación de los esquemas usados para almacenar los casos.

Según (Bello, 1996c) un caso consiste de la descripción de un problema, la solución dada, el resultado de esa solución y si la solución dada no fue exitosa debe añadirse una explicación de

las anomalías, las estrategias de reparación y una referencia al próximo resultado. Otra alternativa es que el caso contenga toda la secuencia del proceso de solución al problema, representando los episodios ocurridos en este proceso.

Los casos según (Bello, 1996c) se caracterizan por:

- Contener conocimiento a nivel operacional.
- Representar situaciones experimentadas previamente.
- Constituir contenidos y el contexto en el cual se aplican.
- Abarcar los objetivos y subobjetivos que surjan en el razonamiento y deben incluir tanto los intentos exitosos como los fallidos para obtener esos objetivos. Los intentos exitosos serán usados para proponer las soluciones a los nuevos problemas y los fallos para prevenir errores potenciales.
- Representar una solución completa la cual puede ser recuperada, adaptada y rehusada posteriormente, eliminando la necesidad de descomponer y recomponer.
- Almacenar la información adicional, la cual se adquiere durante el uso, modificación o almacenamiento del caso.
- Almacenar estructuras monolíticas usadas individualmente o como un conjunto conectado de eventos que son reconstruidos en el momento de la recuperación.

Los casos se pueden representar como unidades de información de complejidad variable pero bien delimitada, mediante conjuntos de nodos de una red que se pueden compartir por varios casos, o como subpartes de una gran jerarquía entre otras representaciones.

Lo esencial para el trabajo de un Sistema Basado en Casos es que todos los casos relevantes al nuevo problema puedan ser recuperados eficientemente desde la base de casos. En esto intervienen dos aspectos principales del sistema: cómo están organizados los casos en la base y cómo se realiza la comparación entre las descripciones del caso y el problema a resolver. Por eso, se dice que la recuperación tiene dos momentos principales: el acceso a los casos y la selección del caso o de los casos similares (Bello, 1996c).

Se han propuesto diversos enfoques para modelar la base de casos, tales como:

- Organización de los casos en una estructura plana.
- Organización de los casos en una jerarquía.
- Organización de los casos según algún criterio de clasificación.
- Organización de los casos utilizando las ventajas de una arquitectura de computadora paralela.

2.5 Conclusiones parciales

El uso de las técnicas de Inteligencia Artificial y los Mapas Conceptuales han facilitado el desarrollo cualitativo de los sistemas de enseñanza/aprendizaje que se elaboran con HESEI, incorporándole a los mismos la capacidad de adaptación al estudiante en una sesión de trabajo y complementando la relación que debe existir entre tecnología y pedagogía.

A través del desarrollo de software educativos de este tipo y la aplicación de las Tecnologías de la Información y las Comunicaciones se propicia el aprendizaje de contenidos.

HESEI utiliza un modelo con arquitectura híbrida que sirve de base para elaborar sistemas de enseñanza/aprendizaje inteligentes, utilizando las técnicas de Reconocimiento de Patrones y Razonamiento Basado en Casos, la cual representa una opción para la capacitación asistida por computadora.

CAPÍTULO III. MANUAL DE USUARIOS

A través de este capítulo se describen las facilidades de uso para los usuarios Estudiante y Profesor de HESEI, el cual necesita requerimientos mínimos para su funcionamiento:

- Procesador Intel 486/100 MHz o superior.
- 16 Mb de memoria RAM.
- Sistema operativo Windows 95 o superior.
- Monitor SVGA.

3.1 Inicio de una sesión de trabajo

La sesión de trabajo de HESEI comienza mostrando su ventana de presentación, la cual se usa para autenticarse en la aplicación (véase figura 3.1).

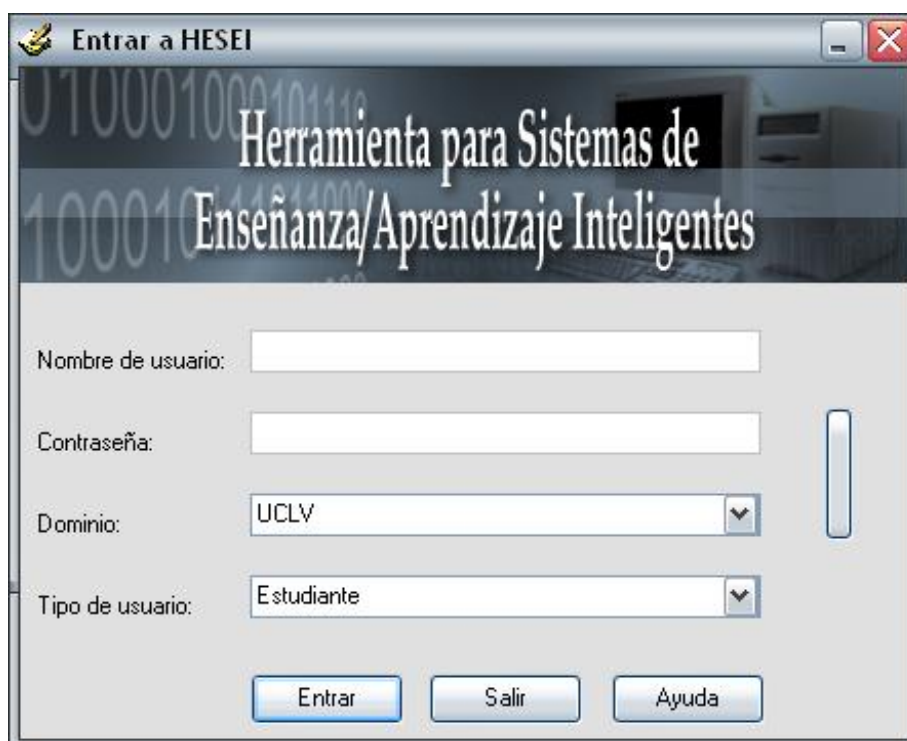


Figura 3.1 Ventana de presentación

Para entrar a HESEI el usuario necesita identificarse:

- Escribir nombre de usuario.
- Escribir contraseña.
- Seleccionar dominio.
- Seleccionar tipo de usuario.

Si el dominio es “UCLV” la herramienta comprueba que la contraseña es válida, sin embargo al utilizar HESEI en otra institución no habrá chequeo de contraseña.

Si selecciona “Profesor” en tipo de usuario se muestra la ventana donde éste podrá elaborar su sistema de enseñanza/aprendizaje.

Al seleccionar “Estudiante” en tipo de usuario se muestra la ventana Estudiante donde el alumno responderá las preguntas del sistema de enseñanza/aprendizaje elegido.

Lo anteriormente presentado se explica en la ventana Ayuda (véase figura 3.2), la cual aparece al pulsar el botón ayuda de la ventana presentación.

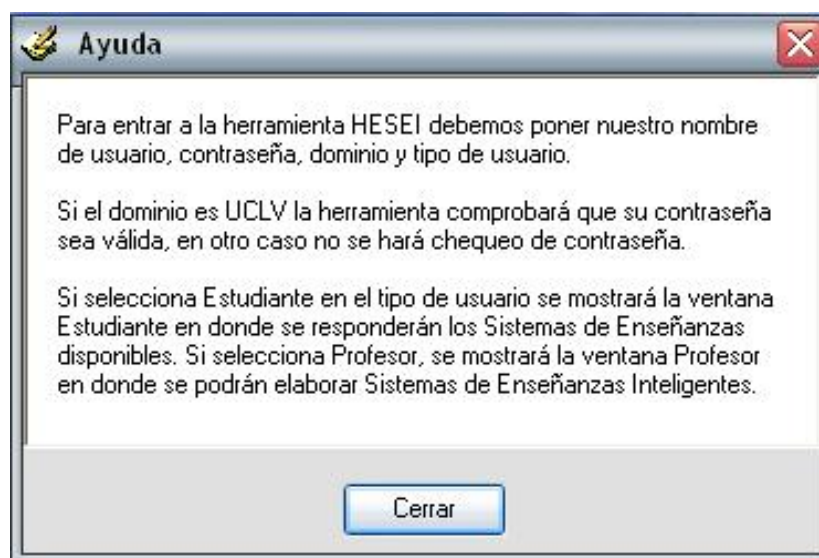


Figura 3.2 Ventana de ayuda de la presentación

Algo característico del software es que ofrece la posibilidad de escoger entre varios estilos de configuración de apariencia, existen estilos predeterminados que se pueden usar para tener una interfaz más atractiva según el gusto del usuario: ventanas coloreadas y diferentes formas de los componentes están disponibles. El usuario seleccionará de todos los estilos el más apropiado (véase figura 3.3).

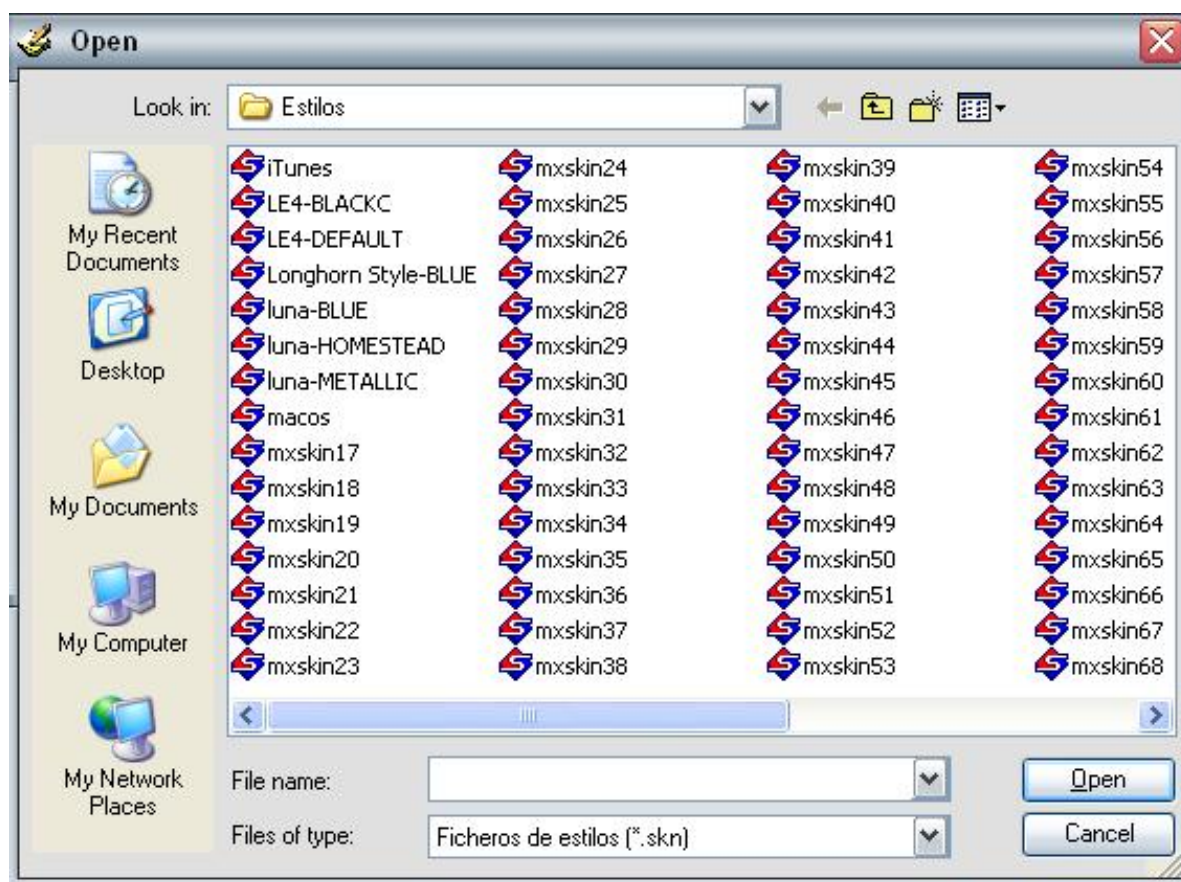


Figura 3.3 Ventana para seleccionar el estilo deseado

3.2 Interfaz para el usuario Profesor

En la ventana Profesor (véase figura 3.4) el menú principal permite ejecutar las siguientes opciones del usuario “profesor”:



Figura 3.4 Ventana Profesor

En Archivo (véase figura 3.5) se tienen las opciones:

- **Crear nuevo Sistema de Enseñanza:** Permite crear un nuevo Sistema de Enseñanza diseñado por el “profesor”, se pondrá el título y seguidamente se define la cantidad de tópicos que lo conforman.
- **Guardar Sistema de Enseñanza:** Al terminar la creación de un Sistema de Enseñanza el “profesor” tiene la opción de guardarlo, y así el “estudiante” podrá responder posteriormente, también el “profesor” puede modificarlo cuando lo desee.
- **Abrir Sistema de Enseñanza:** Esta opción permite abrir un Sistema de Enseñanza anteriormente creado y guardado con el fin de poder modificarlo.
- **Salir:** Salir de la aplicación.



Figura 3.5 Menú Archivo de la ventana Profesor

En Ayuda (véase figura 3.6) se tienen las opciones:

- **Ayuda:** Esta opción ofrece ayuda al usuario de cómo usar la herramienta.
- **Acerca de:** Muestra el título de la aplicación y los autores.



Figura 3.6 Menú Ayuda de la ventana Profesor

Para la creación de un nuevo Sistema de Enseñanza, se debe proporcionar el nombre del tema y la cantidad de tópicos que lo conforman (véase figura 3.7).

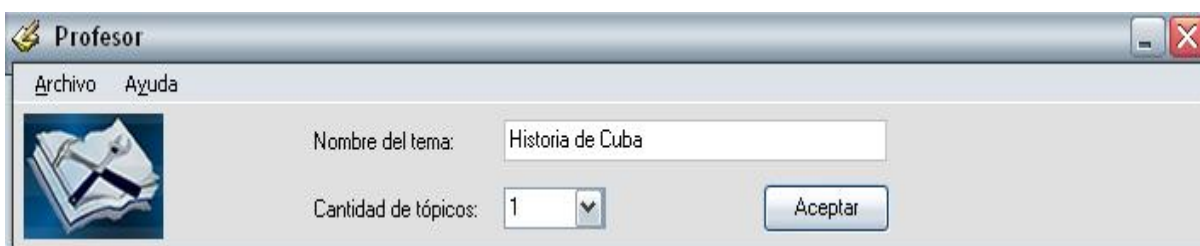


Figura 3.7 Ventana para crear un nuevo SEI

Después de almacenada esta información se debe elegir el tópico que se desea configurar (véase figura 3.8).

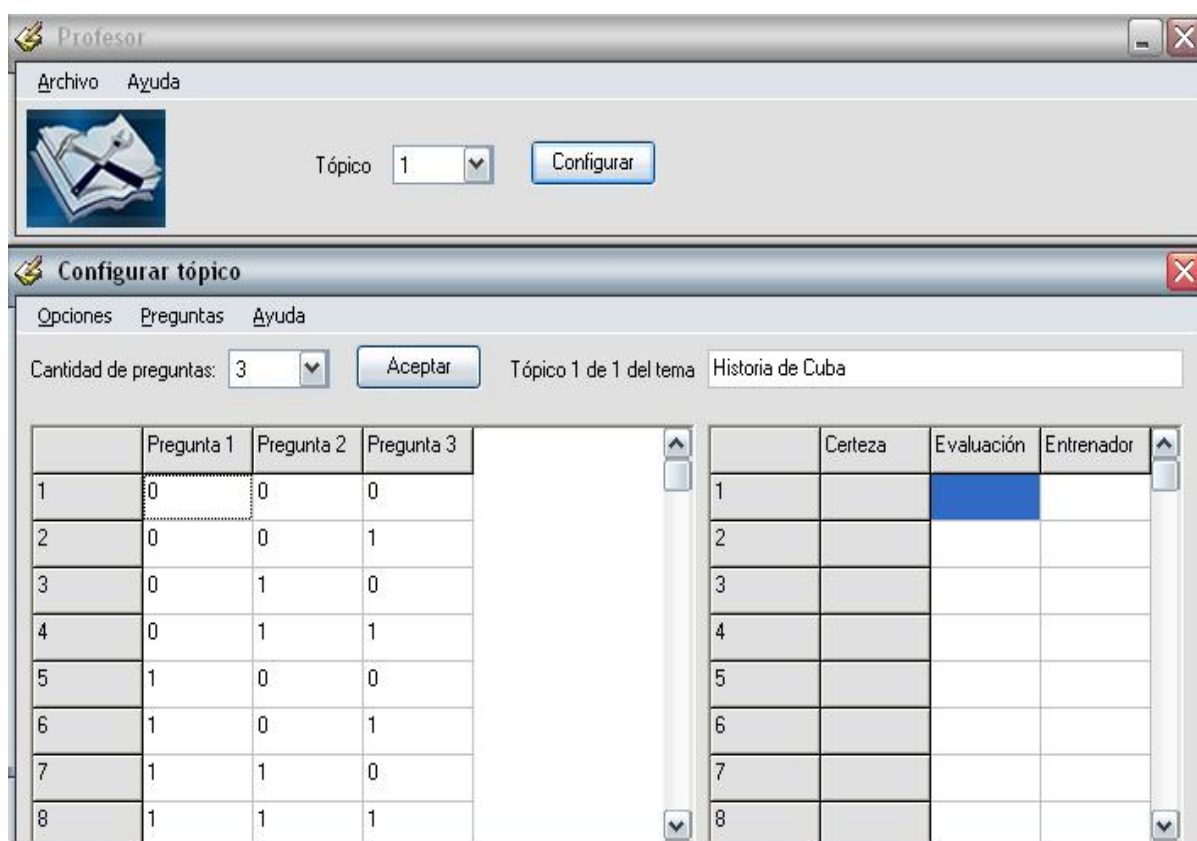


Figura 3.8 Ventana para seleccionar el tópico a configurar

Al presionar el botón Configurar se mostrará la ventana Configurar tópico (véase figura 3.9) donde el “profesor” deberá introducir el número de preguntas que lo conforman.

Seguidamente al presionar el botón Aceptar, se generan todas las posibles combinaciones de preguntas bien o mal respondidas por un alumno, es decir, si el número de preguntas es n el número de combinaciones es 2^n .

Es responsabilidad del “profesor” colocar la evaluación que él entienda conveniente según su apreciación. Esta tarea puede realizarse manualmente analizando cada una de las combinaciones o auxiliarse del Asistente para la evaluación.



3.9 Ventana para configurar un tópico

En el menú Opciones (véase figura 3.10) se encuentran todas las operaciones a realizar con el tópico que se está configurando.

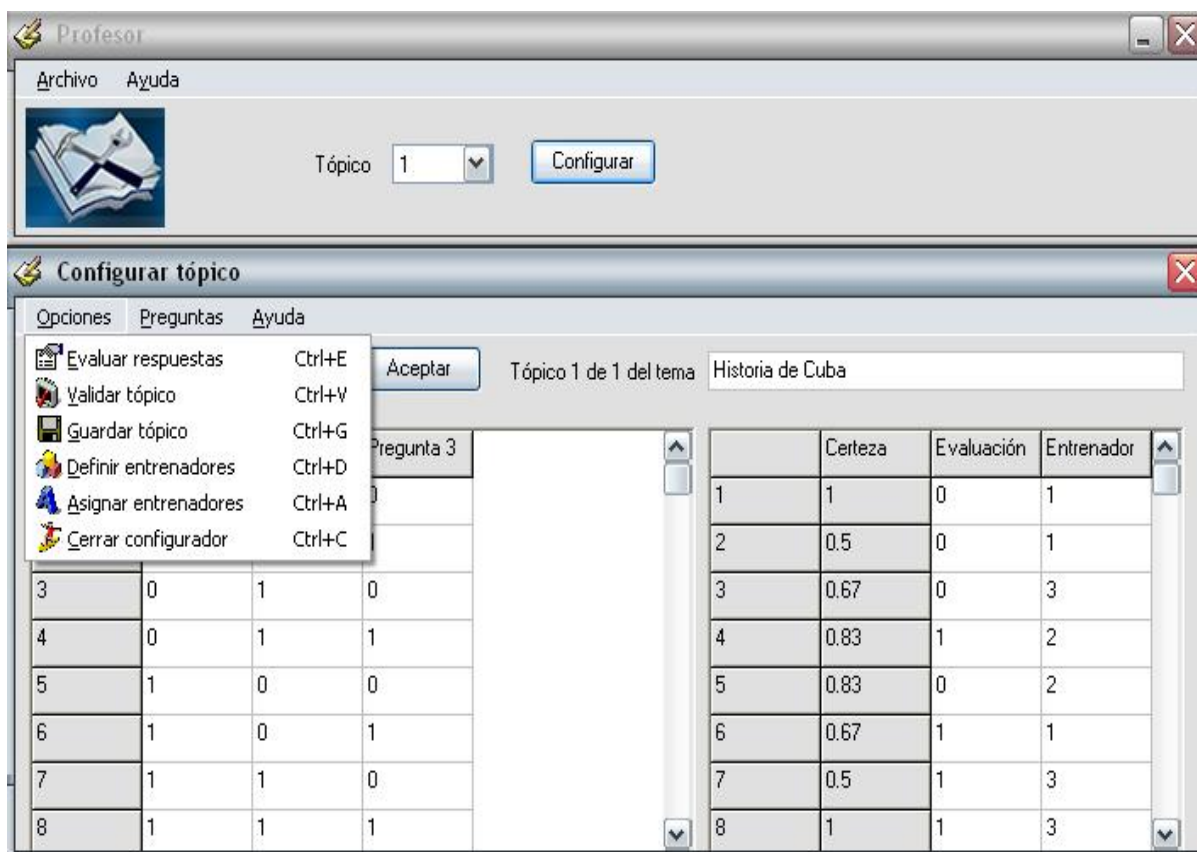


Figura 3.10 Menú Opciones de la ventana para configurar un tópico

En la opción Asistente para la evaluación (véase figura 3.11), se muestra una ventana donde se elige el por ciento de preguntas correctamente respondidas para que el tópico sea aprobado, además se debe seleccionar la posición que las preguntas ocupan: Al principio, al final o desordenadas. Al presionar el botón Aceptar se le coloca la evaluación a todas las combinaciones que cumplen esa condición.

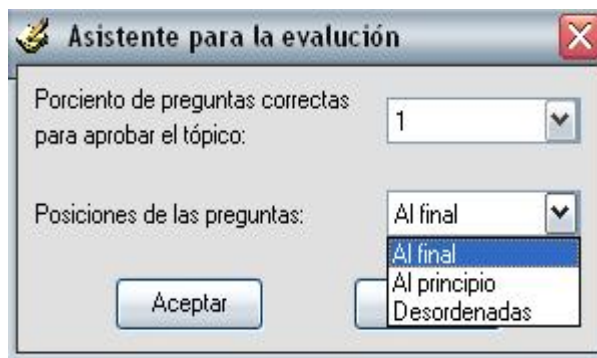


Figura 3.11 Ventana del asistente para la evaluación

En la opción Validar tópico, la herramienta calcula la certeza de la evaluación teniendo en cuenta que la complejidad de las preguntas depende del orden que ellas tengan en el tópico, es decir, la pregunta $k+1$ debe tener un nivel de dificultad superior al de la pregunta k , y así sucesivamente, por lo que el cálculo de la certeza de la evaluación ayudará al “profesor” a colocar el entrenador adecuado a cada tipo de respuesta.

Seguidamente la herramienta revisa si alguna pregunta no está teniendo peso en la evaluación, es decir que no esté influyendo en la evaluación y de esta forma, si el profesor lo desea, tiene la posibilidad de eliminarla del tópico que está confeccionando. Para esto se muestra la ventana Selección (véase figura 3.12) que permite elegir la combinación de preguntas suficientes para la evaluación dada. Las evaluaciones se darán de 0 ó 1, es decir “bien” o “mal”.

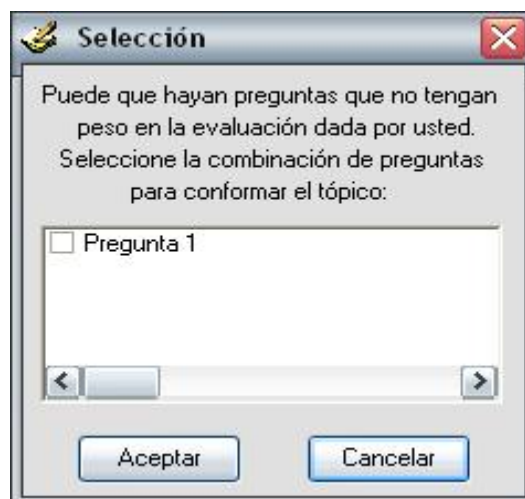


Figura 3.12 Ventana Selección

Otra opción es Guardar tópico (véase figura 3.13), permite guardar la configuración del tópico que se ha realizado, para esto el tópico no tiene que estar totalmente terminado.

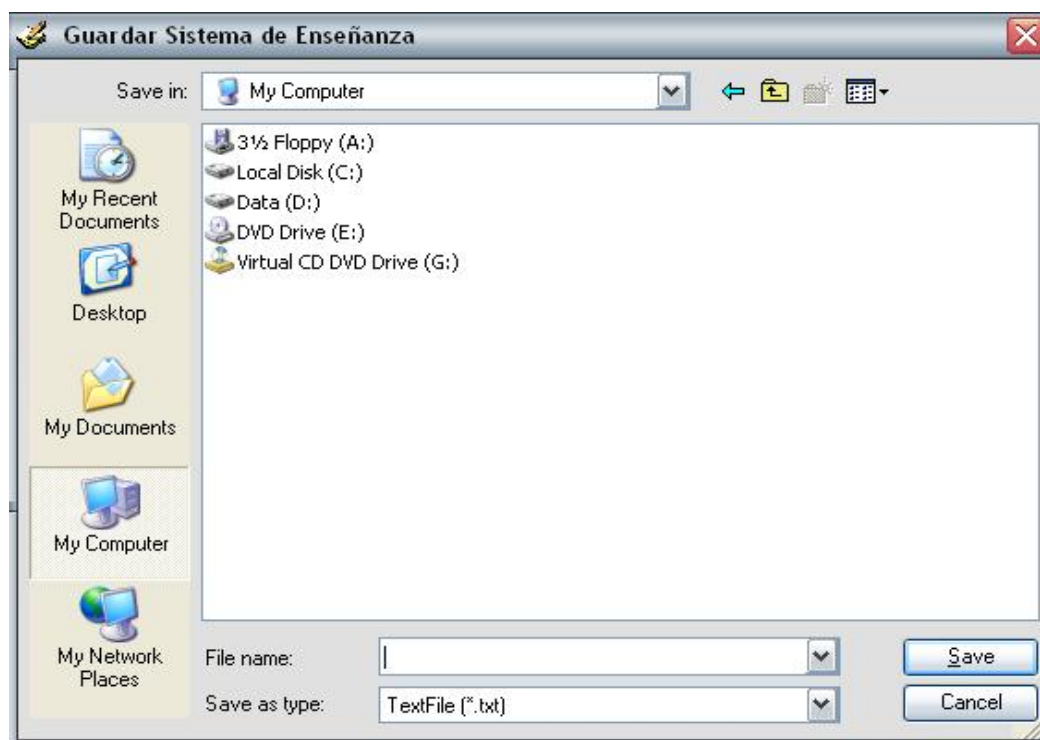


Figura 3.13 Ventana para guardar SEI

Definir entrenadores (véase figura 3.14) ofrece la posibilidad de obtener información sobre los entrenadores que el profesor va a colocar en la herramienta, los cuales están destinados a cada tipo de estudiante. De una manera sencilla y cómoda al indicar donde está el entrenador y hacia donde lo queremos copiar, en un estilo similar al explorador de Windows permite buscar el entrenador que va a ser añadido a través del botón Adicionar a la tabla de entrenadores, el botón Eliminar permite quitar de la tabla un entrenador. El entrenador asociado a cada evaluación será el número que ocupa en la tabla de entrenadores. Se aconseja crear una carpeta para almacenar todos los entrenadores según su tópico.

Esta ventana incluye un botón que muestra la ventana Ayuda (véase figura 3.15) en la cual se explica todo lo planteado anteriormente.



Figura 3.14 Ventana para definir entrenadores

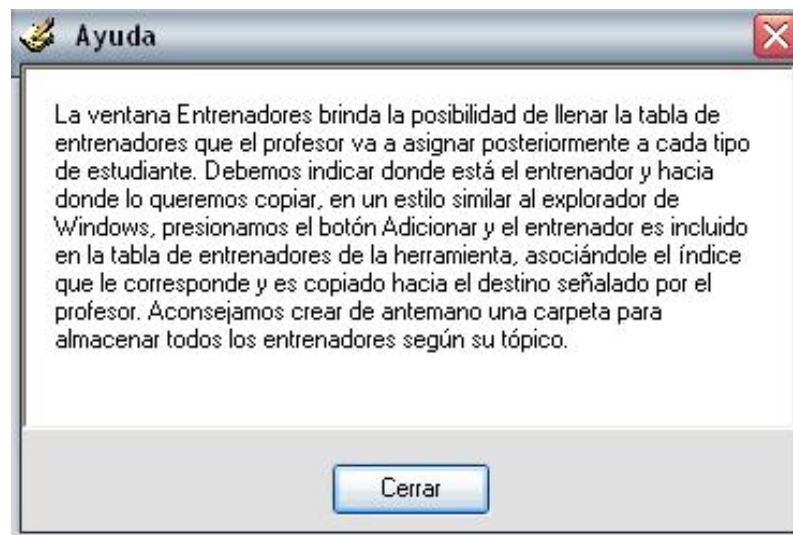


Figura 3.15 Ayuda de la ventana para definir entrenadores

Asignar entrenadores permite asignar a cada combinación de respuestas el entrenador que estime conveniente el profesor, es por ello que esta tarea puede ser realizada manualmente o a través del Asistente para asignar entrenador (véase figura 3.16).

Éste ofrece la opción de evaluar de dos formas: clásica y avanzada. Se selecciona primeramente la forma de evaluación que se utilizará y seguidamente el entrenador a asignar. Si es de forma clásica se especifica desde y hasta que combinación se asigna el entrenador, si es de forma avanzada se especifica qué evaluación y con qué certeza se asocia el entrenador seleccionado. Al presionar el botón Aceptar la acción será ejecutada.

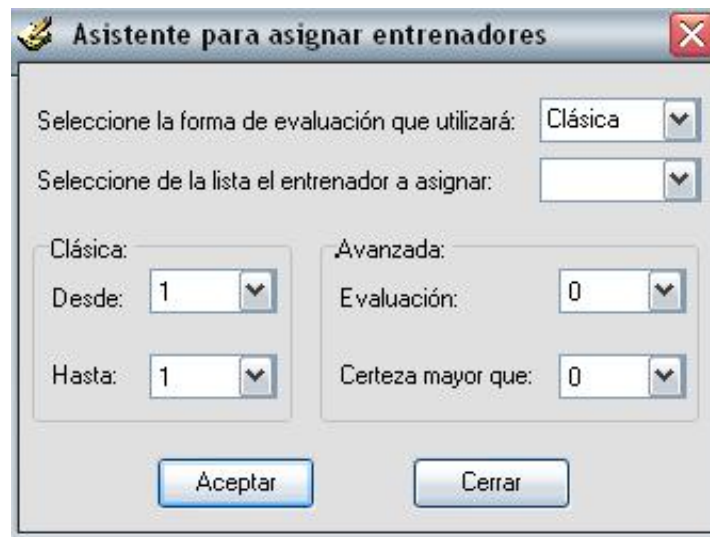


Figura 3.16 Ventana del asistente para asignar entrenadores

En el menú Preguntas (véase figura 3.17) se pueden confeccionar nuevas preguntas o modificar las ya preparadas, los tipos de preguntas disponibles son: Relacionar columnas, Verdadero o Falso y Marcar la correcta. Se debe comenzar a editar las preguntas conociendo que la configuración del tópico es adecuada.

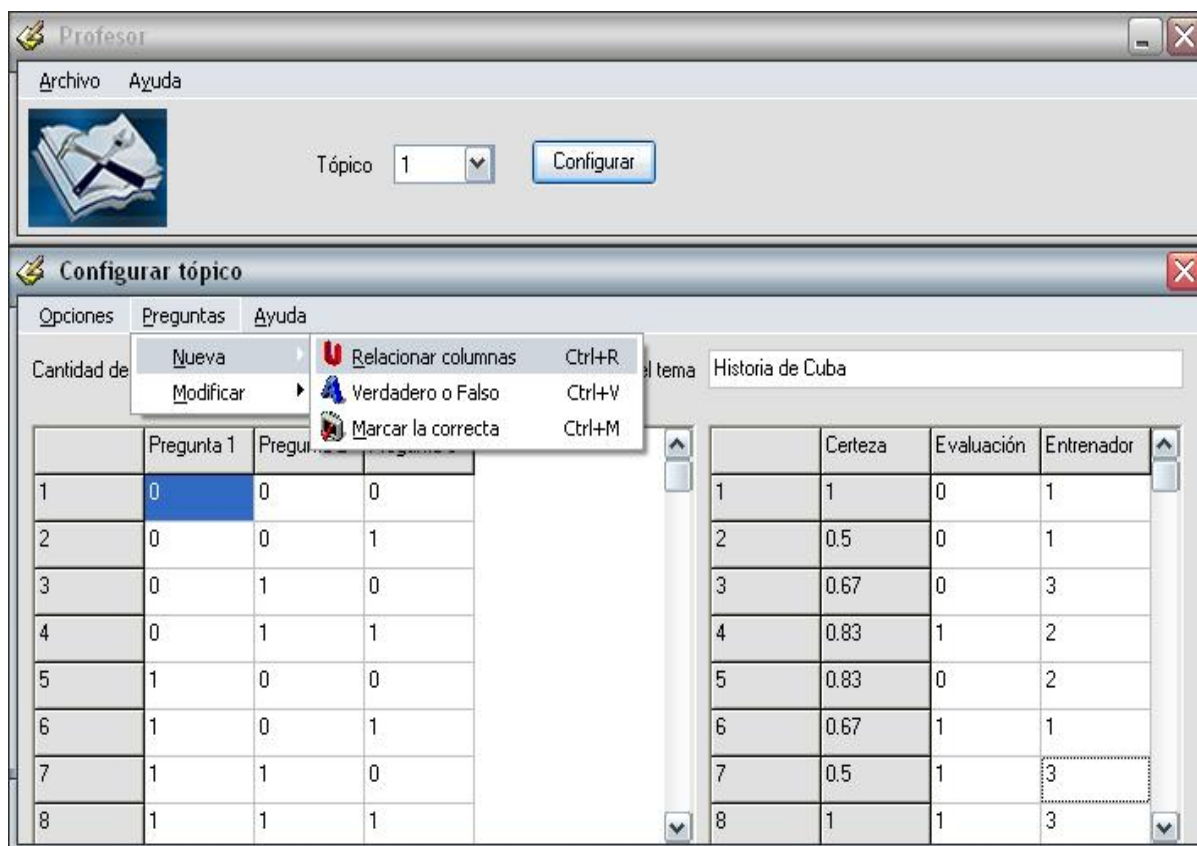


Figura 3.17 Menú Preguntas de la ventana para configurar un tópico

El trabajo con el editor de preguntas es sencillo, de tal manera que no hay que ser especialista en computación para construir o modificar una pregunta.

En el tipo de pregunta Relacionar columnas se debe especificar el título, la cantidad de páginas (véase figura 3.18), seguidamente la cantidad de elementos de la columna A (véase figura 3.19), y el texto correspondiente a los elementos de esta columna (véase figura 3.20), de manera similar con la columna B (véase figuras 3.21 y 3.22), y al terminar, la solución correcta debe ser especificada por el profesor (véase figura 3.23).

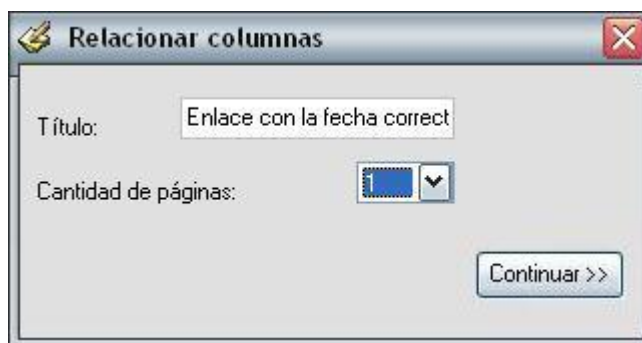


Figura 3.18 Ventana para definir título y cantidad de páginas

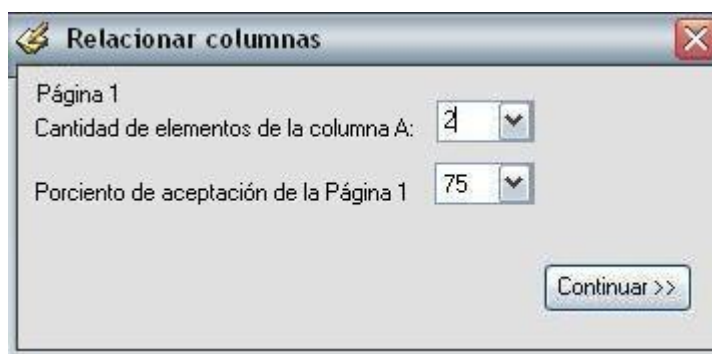


Figura 3.19 Ventana para definir la cantidad de elementos de la columna A

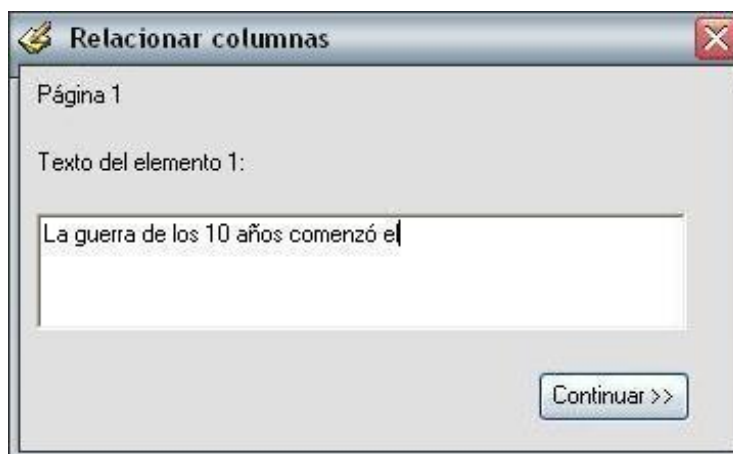


Figura 3.20 Ventana para insertar el texto del primer elemento

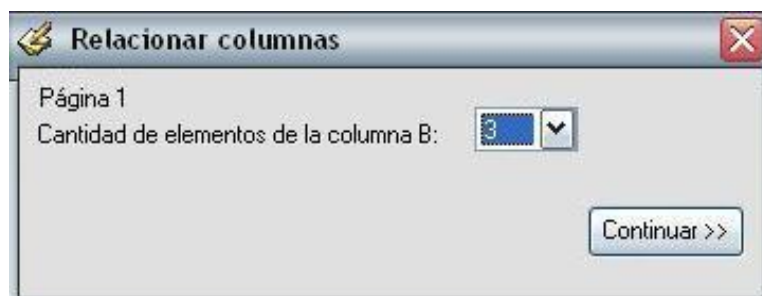


Figura 3.21 Ventana para definir la cantidad de elementos de la columna B

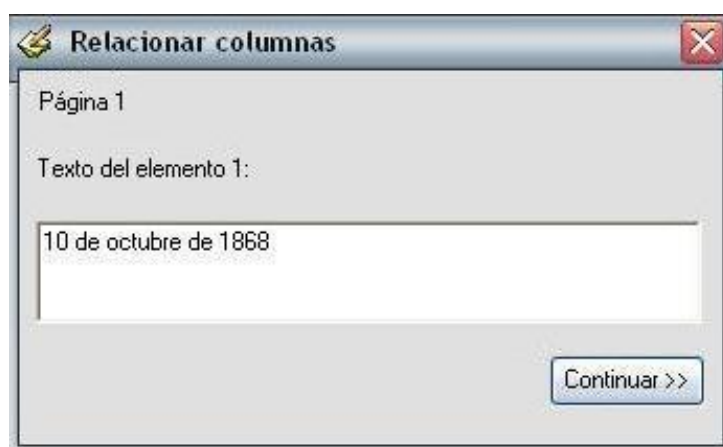


Figura3.22 Ventana para insertar el texto del segundo elemento

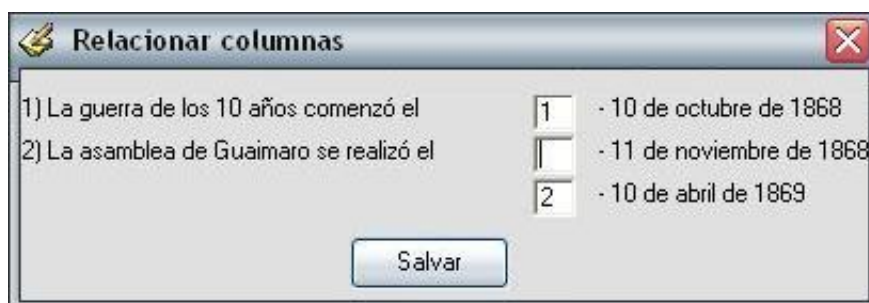
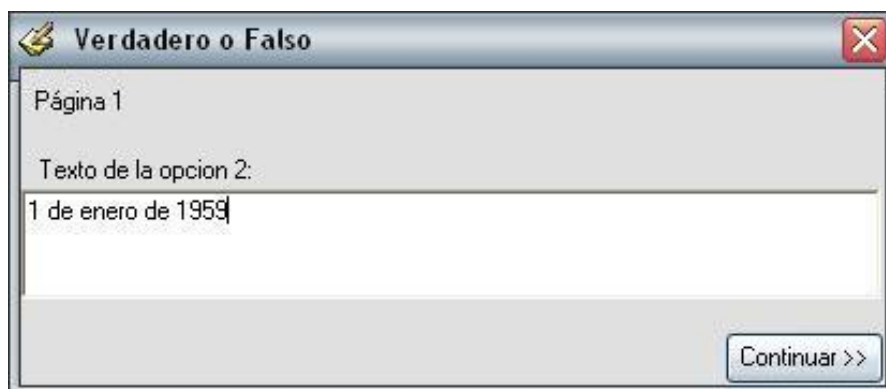


Figura 3.23 Ventana para dar solución a la pregunta

En el tipo de pregunta Verdadero o Falso se debe especificar el título, la cantidad de páginas (véase figura 3.24), seguidamente el texto del enunciado de la pregunta y la cantidad de posibilidades (véase figura 3.25). Posteriormente se introduce el texto de las opciones (véase figura 3.26) y se evalúan de verdadero o falso, según sea el caso (véase figura 3.27).


Figura 3.24 Ventana para definir título y cantidad de páginas

Figura 3.25 Ventana para definir el enunciado y la cantidad de posibilidades



The screenshot shows a window titled "Verdadero o Falso" with a pencil icon and a close button. Inside, it says "Página 1" and "Texto de la opción 2:". Below this is a text input field containing "1 de enero de 1959". At the bottom right is a button labeled "Continuar >>".

Figura 3.26 Ventana para insertar el texto de una opción



The screenshot shows the same window with the question "La Revolución Cubana triunfó el" displayed. Below the question are two options, each with a dropdown menu and a date: "Falso" with "- 1 de enero de 1960" and "Verdadero" with "- 1 de enero de 1959". At the bottom center is a button labeled "Salvar".

Figura 3.27 Ventana para dar solución a la pregunta

En el tipo de pregunta Marcar la correcta se debe especificar el título, la cantidad de páginas (véase figura 3.28), seguidamente el texto del enunciado de la pregunta y la cantidad de posibilidades (véase figura 3.29). Posteriormente se introduce el texto de las opciones (véase figura 3.30) y se selecciona la opción correcta según sea el caso (véase figura 3.31).

Marcar la correcta

Título: Seleccione la respuesta corr

Cantidad de páginas: 1

Porcentaje de aceptación: 60

Continuar >>

Figura 3.28 Ventana para definir título y cantidad de páginas

Marcar la correcta

Página 1

Texto del enunciado: Quién fue el protagonista de la Protesta de Baraguá

Cantidad de posibilidades: 3

Continuar >>

Figura 3.29 Ventana para definir el enunciado y la cantidad de posibilidades

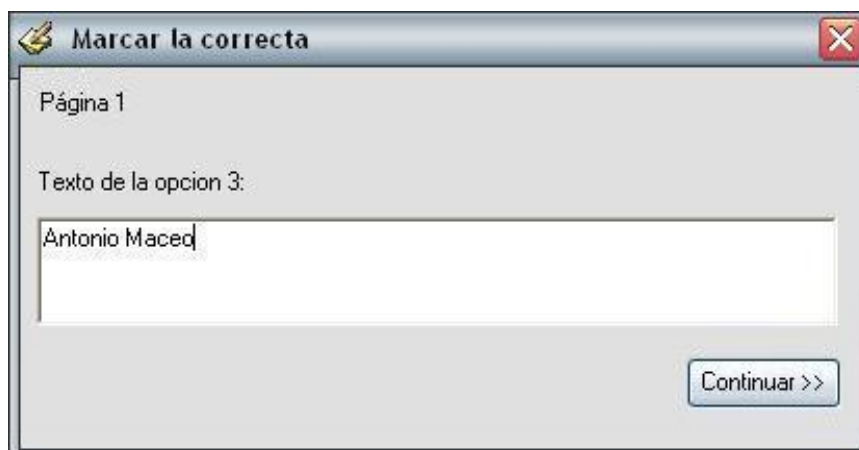


Figura 3.30 Ventana para insertar el texto de una opción



Figura 3.31 Ventana para dar solución a la pregunta

De esta forma la herramienta almacena las soluciones correctas que posteriormente serán comparadas con las soluciones ofrecidas por los estudiantes.

3.3 Interfaz para usuario Estudiante

En la ventana Estudiante (véase figura 3.32) aparecen las opciones que puede ejecutar el usuario estudiante.



Figura 3.32 Ventana del Estudiante

Este usuario tiene la posibilidad de seleccionar el sistema de enseñanza/aprendizaje con el que va a interactuar, seguidamente aparecen las preguntas que debe responder pertenecientes al tema y posteriormente pasa a estudiar con su SEI adaptado para él, donde podrá acceder a cada elemento del mapa, que conforma su software educativo (véase figura 3.33).

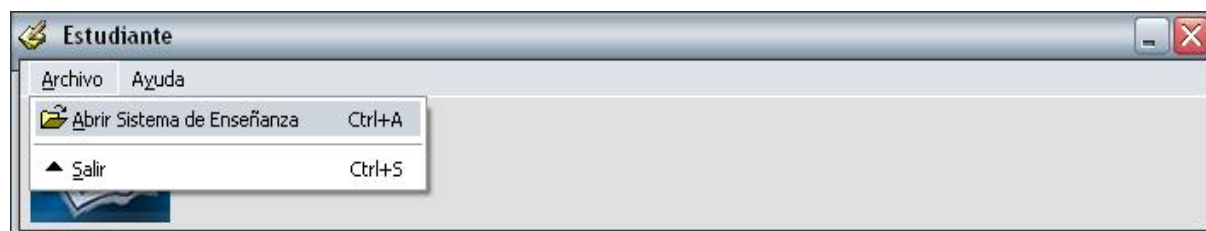


Figura 3.33 Menú Archivo de la ventana del Estudiante

En esta ventana se observa el sistema de enseñanza/aprendizaje estructurado en forma de Mapa Conceptual Interactivo, donde los conceptos del mapa son entrenadores del software educativo generado para el alumno según sus características.

Sus nodos son sensibles, al dar clic sobre ellos se accederá al entrenador designado en ese tópico, es por eso que se denominan interactivos ya que no solo tienen la función de mostrar un concepto o información, si no que se usan como soporte para el documento, entrenador o software con el cual el estudiante va a interactuar (véase figura 3.34).

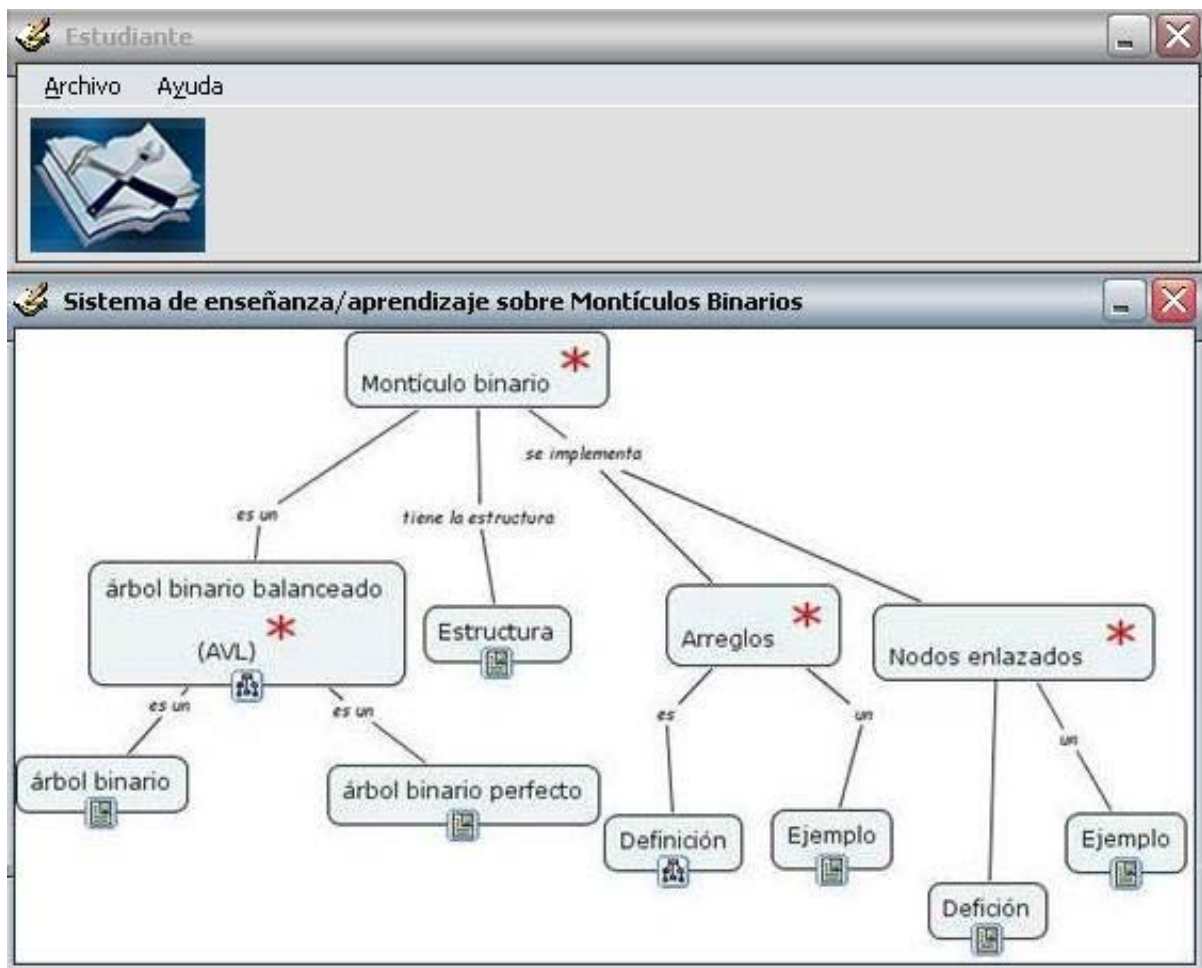


Figura 3.34 Ejemplo de un MC para aprender Montículos Binarios

CONCLUSIONES

Se logra la implementación de HESEI tomando como marco teórico el razonamiento basado en casos, que permitió aprovechar un grupo de características que son útiles en la elaboración de sistemas de enseñanza/aprendizaje inteligentes tales como:

1. La adquisición del conocimiento, que al razonar desde episodios específicos, evitó el problema de descomponer el conocimiento del dominio y generalizarlo en reglas.
2. La flexibilidad en la representación del conocimiento, ya que esta no se encuentra sólo reflejado en la representación de los casos, sino también en la forma de su organización, la estrategia de recuperación y la estrategia de adaptación.
3. El mantenimiento del conocimiento: el profesor puede agregar nuevos casos sin la intervención de un especialista en computación.
4. El reuso de las soluciones previas al resolver un problema, aumenta la eficiencia del sistema de enseñanza/aprendizaje elaborado.

Las soluciones derivadas están fundamentadas en casos reales lo que le permite al sistema justificar las decisiones al profesor.

Los sistemas de enseñanza/aprendizaje inteligentes que se elaboran con HESEI emplean un Modelo del Estudiante, el cual incluye los casos que contienen los rasgos afectivos y cognitivos del alumno.

La interfaz visual que utiliza HESEI permite elaborar de manera sencilla un sistema de enseñanza/aprendizaje inteligente con estructura de Mapa Conceptual para una temática determinada.

RECOMENDACIONES

- Incorporar nuevos tipos de preguntas para mejorar la interacción del estudiante con los software educativos elaborados con HESEI.
- Valorar el uso de representaciones del conocimiento combinadas con los Mapas Conceptuales, para lograr no sólo la organización del conocimiento si no también la realización de inferencias a partir de él, como pudieran ser las Redes Bayesianas y la Programación Lógica.

REFERENCIAS BIBLIOGRÁFICAS

AAMODT, A. & PLAZA, E. (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*.

ÁGUILA, L. & SHULCLOPER, J. L. (1984) Algoritmo CC para la elaboración de la información k-valente en problemas de Reconocimiento de Patrones. *Revista Ciencias Matemáticas*. Cuba.

ALEMÁN, A. (1998) La enseñanza de la Matemática asistida por computadora.

ALESSI, S. M. & TROLLIP, S. R. (1985) *Computer-Based Instruction: Methods and Development*, Englewood Cliffs, New Jersey, Prentice Hall.

AUSUBEL, D. P. (1989) *Psicología Educativa: un punto de vista cognoscitivo*, Trillas, México.

AYALA, G. (2002) Software Educativo.

AYAQUICA, I. O. (1997) Un nuevo algoritmo de escala exterior para el cálculo de testores típicos. *Memorias del II Taller Iberoamericano de Reconocimiento de Patrones*. La Habana, Cuba.

BELLO, R. E. (1996a) Cap 3 El Razonamiento Basado en Casos en la Inteligencia Artificial. *Sistemas Basados en Casos*. Santa Clara, Cuba, Edit. UCLV.

BELLO, R. E. (1996b) Cap 4 Arquitectura funcional de los Sistemas con Razonamiento Basado en Casos. *Sistemas Basados en Casos*. Santa Clara, Cuba, Edit. UCLV.

BELLO, R. E. (1996c) Cap 5 Organización de la Base de Casos. *Sistemas Basados en Casos*. Santa Clara, Cuba.

BELLO, R. E., ET AL. (2002) *Aplicaciones de la Inteligencia Artificial*, Jalisco, México, Universidad de Guadalajara.

BOOCH, G. (1991) *Object Oriented design: with applications*, U.S.A., The Benjamin Cummings Publishing's Company Inc.

BOOCH, G., RUMBAUGH, J. & JACOBSON, I. (1999) *The Unified Modeling Language User Guide*, Addison-Wesley Longman.

BRAVO, A. (1983) Algoritmo CT para el cálculo de los testores típicos de una matriz k-valente. *Revista Ciencias Matemáticas*. Cuba.

CABRERA, A. (1995) Informática Educativa: La revolución construccionista. . *Informática y Automática*.

GAINES, B. R. & SHAW, M. L. G. (1995) Collaboration Through Concept Maps. Universidad de Calgary, Alberta, Canadá.

GALVIS, A. (1994) *Ingeniería de Software Educativo*, Santafe de Bogotá, Colombia.

GARCÍA, Z. Z. (1993) Investigación y elaboración de Sistemas de Enseñanza Inteligentes. *Ciencias de la Computación*. Santa Clara, Cuba, UCLV.

GARCÍA, Z. Z., MAZAIRA, J. L. & FERNÁNDEZ, A. (2000) Hipermedia para la enseñanza de las estructuras básicas de control de la programación estructurada. *V Congreso Iberoamericano de Informática Educativa*. Chile.

GUTIÉRREZ, I. (2003) *Modelo para la Toma de Decisiones usando Razonamiento Basado en Casos en condiciones de Incertidumbre*, Villa Clara, Cuba, Departamento de Ciencia de la Computación. Universidad Central de Las Villas.

KENDALL, K. E. & KENDALL, J. E. (1997) *Análisis y diseño de sistemas*, Prentice Hall.

KOLODNER, J. (1992) An Introduction to Case Based Reasoning. *Artificial Intelligence Review*. 6 ed.

LABORDA, J. (1986) *Informática Educativa y futuro*.

LIEBOWITZ, J. (1993) *Roll your own hybrids*.

MORELL, C., ET AL. (2002) Elaboración de tecnologías de maquinado con el auxilio de computadora. *Centro Azúcar*.

NOVAK, J. D. & GOWIN, D. B. (1988) *Aprendiendo a Aprender*, Barcelona, Ediciones Martínez Roca, S.A.

PINHEIRO, L. E. (2000) *LDAP-Linux-Como*.

PONS, A. & SANTIESTEBAN, Y. (2003) LEX: Un nuevo algoritmo para el cálculo de los testores típicos. *Revista Ciencias Matemáticas*. Cuba.

RODRÍGUEZ, M. (1999) El concepto de tipo y la teoría de programación actual. *Revista GIGA*. Cuba.

ROSEMBERG, R. (1990) A critical analysis of research on Intelligent Tutoring Systems. *Expert Systems and Intelligent CAI*. Englewood Cliffs, New Jersey, Educational Technology Publications.

RUMBAUGH, J., BOOCH, G. & JACOBSON, I. (1997a) UML Notation.

RUMBAUGH, J., BOOCH, G. & JACOBSON, I. (1997b) UML Semantics.

SEVILLANO, M. L. (1998) *Nuevas Tecnologías, Medios de Comunicación y Educación. Formación inicial y permanente del profesorado*, España, Editorial CCS.

SHULCLOPER, J. R., ALBA CABRERA, E. & LAZO CORTES, M. (1995) *Introducción a la teoría de Testores Típicos*, México.

SHULCLOPER, J. R., ET AL. (1982) Algoritmos BT y TB para el cálculo de todos los tests típicos. *Revista Ciencias Matemáticas*. Cuba.

WENGER, E., ET AL. (1987) *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*, California, Morgan Kaufmann Publishers, Inc.

BIBLIOGRAFÍA

BERLANGA, A. J. & GARCÍA, F. J. (2004) Sistemas hipermedia adaptativo en el ámbito de la educación. España, Universidad de Salamanca.

CAÑAS, A. J. & CARVALHO, M. Concept Maps and AI: an Unlikely Marriage? Manaus, Brasil, Institute for Human & Machine Cognition.

JOSEMARÍA, C. (2001) Diseño de multimedias educativas. Argentina.

LAUREANO, A. L. & DE ARRIAGA, F. (2001) Técnicas de Diseño en Sistemas de Enseñanza Inteligentes. Revista UNAM. Ciudad Universitaria, México D.F.

MARTÍNEZ, Y. (2005) Introducción de los Test Adaptativos Informatizados en las Universidades Cubanas Universidad de Ciego de Ávila, Cuba.

ONTORIA, A. (2003) Mapas Conceptuales, México, D.F., Alfaomega Grupo Editor, S.A. de C.V.

PADRÓN, L. J. (2005) Las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC) en la formación del hombre nuevo. Universidad Médica de Villa Clara, Cuba.

PEARL, P. (1993) Issues in Case-Base Design System. Analysis and Manufacturing (AI-EDAM) Magazine. England.

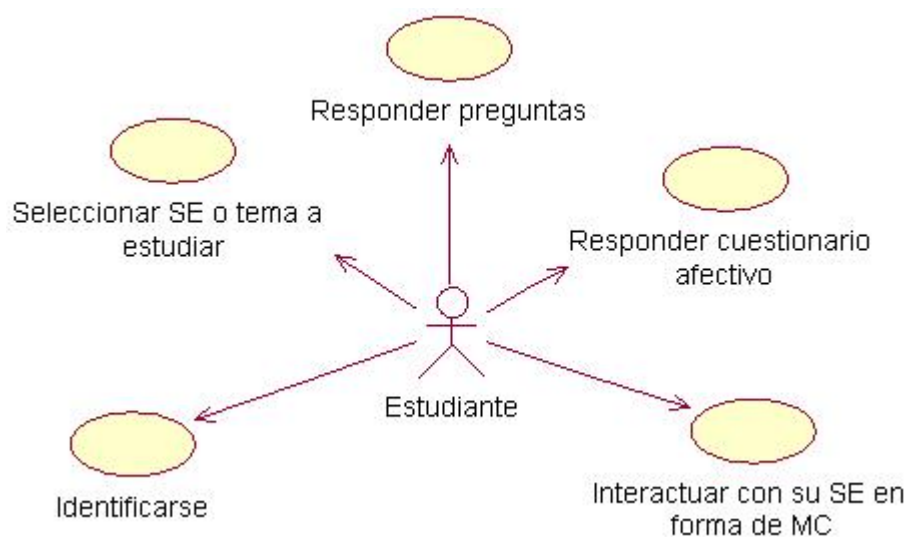
PEÑA, C. I., MARZO, J. L., LLUIZ, J. & FABREGAT, R. (2002) Un sistema de tutoría inteligente adaptativo considerando estilos de aprendizaje. España, Universidad de Girona.

VILLARREAL, G. (2004) Agentes Inteligentes en Educación. Chile, Universidad de Santiago de Chile.

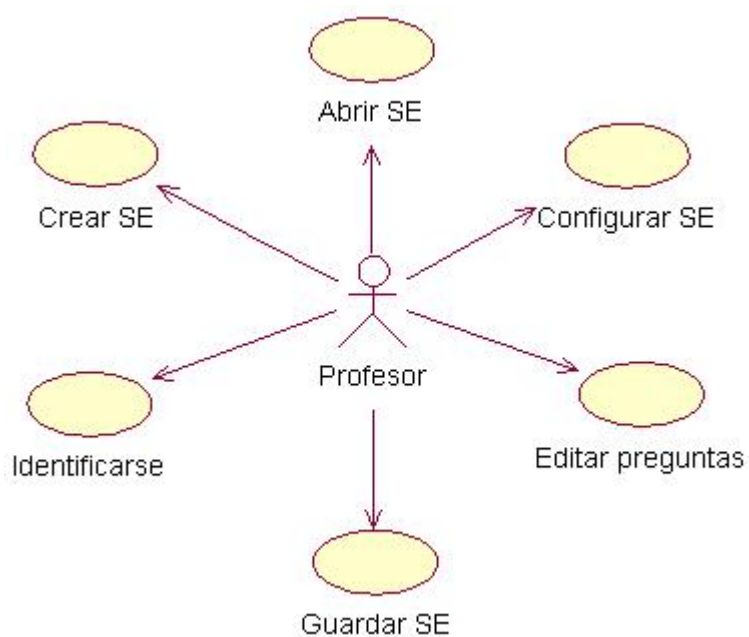
ANEXOS

Anexo 1. Casos de uso del Sistema

Actor Estudiante

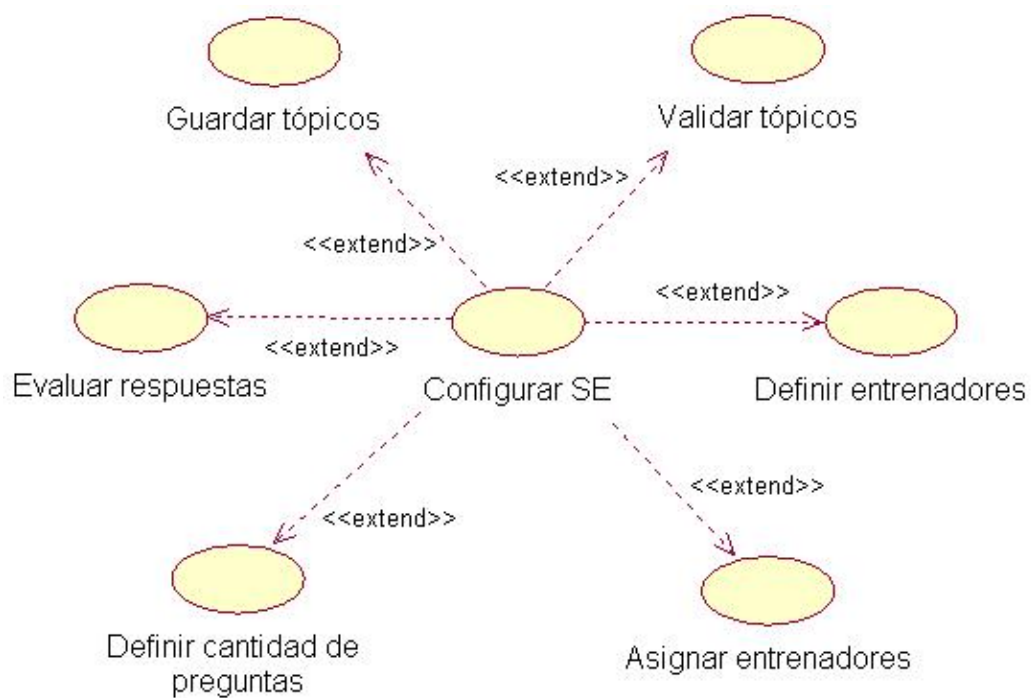


Actor profesor

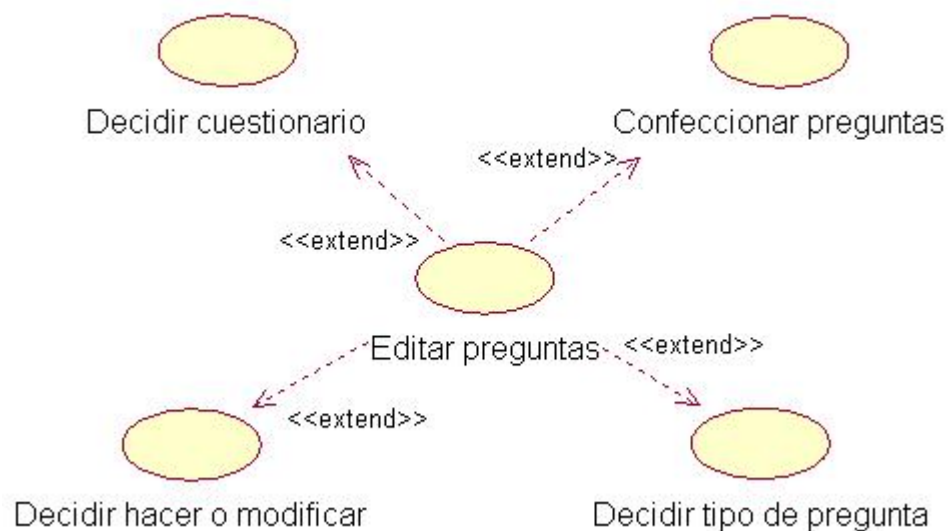


Anexo 2. Expansión de Casos de uso

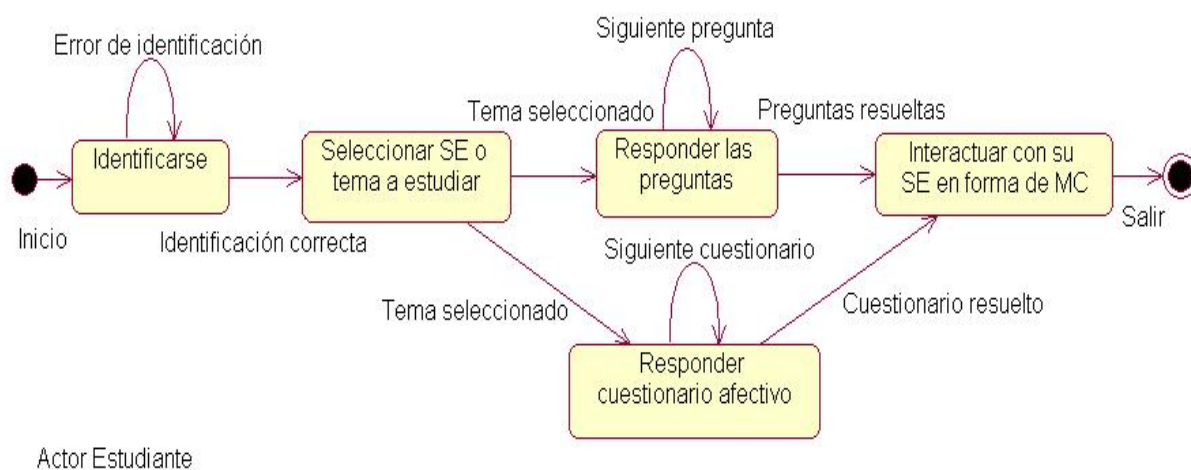
Expansión del caso Configurar Sistema de Enseñanza



Expansión del caso Editar Preguntas



Anexo 3. Diagrama de Transición de Estado



Anexo 4. Diagrama de Actividades