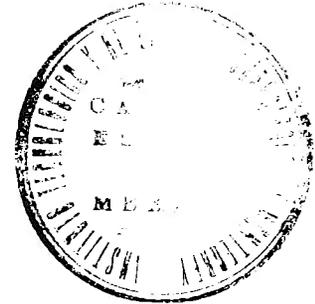


INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY  
CAMPUS ESTADO DE MÉXICO



BIBLIOTECA



**EVALUACIONES AUTOMÁTICAS Y RETROALIMENTACIÓN  
BASADAS EN UN SISTEMA DE REGLAS.  
"EVALUATEC"**

TESIS QUE PARA OPTAR EL GRADO DE  
MAESTRO EN CIENCIAS COMPUTACIONALES  
PRESENTA

**CARLOS ALBERTO DORANTES DOSAMANTES**

Asesor : Dra. CLAUDIA GUTIÉRREZ MAZZOTTI  
Comité de tesis : M. en C. ALEJANDRO MADARIAGA NAVARRETE  
Dr. CARLOS RODRÍGUEZ LUCATERO

Jurado : Dr. CARLOS RODRÍGUEZ LUCATERO  
Dr. RICARDO DUHNE AGUAYO  
Dra. CLAUDIA GUTIÉRREZ MAZZOTTI  
M. en C. ALEJANDRO MADARIAGA NAVARRETE

Presidente  
Secretario  
Vocal  
Vocal

Querétaro, Qro., México, diciembre de 1996.

# CONTENIDO

<b>TABLA DE FIGURAS .....</b>	<b>6</b>
<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>7</b>
1.1. ANTECEDENTES .....	9
1.2. MOTIVACIÓN DEL PROYECTO .....	11
1.2.1. <i>Situación actual de los sistemas computacionales aplicados a la evaluación en el I.T.E.S.M. Campus Querétaro.</i> .....	11
1.2.2. <i>Sistemas actuales de evaluación por computadora.</i> .....	11
1.2.3. <i>Conclusión de la situación planteada.</i> .....	15
1.3. CONTRIBUCIONES.....	16
<b>CAPÍTULO 2. TIPOS DE EVALUACIONES POR COMPUTADORA.....</b>	<b>17</b>
<b>CAPÍTULO 3. DISEÑO DE UN SISTEMA INTEGRAL DE EVALUACIÓN POR COMPUTADORA.....</b>	<b>19</b>
3.1. DIAGRAMA DE ENTIDAD-RELACIÓN.....	20
3.2. MÓDULOS DEL SISTEMA .....	21
3.2.1. <i>Diagrama general del Sistema Integral de Evaluación.</i> .....	22
3.2.2. <i>Módulo I. Interface con tablas de operación del sistema de Servicios Escolares.</i> .....	22
3.2.3. <i>Módulo II. Registro de temas y subtemas (programas analíticos)</i> .....	24
3.2.4. <i>Módulo III. Diseño de reactivos (preguntas y respuestas)</i> .....	24
3.2.5. <i>Módulo IV. Diseño de evaluaciones.</i> .....	25
3.2.6. <i>Módulo V. Intérprete de evaluaciones.</i> .....	27
3.2.7. <i>Módulo VI. Estadísticas y consultas</i> .....	28
<b>CAPÍTULO 4. SISTEMAS EXPERTOS Y REPRESENTACIÓN DEL CONOCIMIENTO BASADO EN REGLAS .....</b>	<b>29</b>
4.1. DEFINICIÓN DE SISTEMA EXPERTO .....	29
4.2. COMPONENTES DE UN SISTEMA EXPERTO.....	31
4.3. REPRESENTACIÓN DEL CONOCIMIENTO .....	32
4.4. TIPOS DE RAZONAMIENTO EN UN SISTEMA BASADO EN REGLAS.....	33
<b>CAPÍTULO 5. INTÉRPRETE DE EVALUACIONES Y RETROALIMENTACIÓN BASADA EN UN SISTEMA DE REGLAS. ....</b>	<b>35</b>
5.1. TIPOS DE REACTIVOS SOPORTADOS EN UNA EVALUACIÓN. ....	37
5.2. INFORMACIÓN PRESENTADA EN CADA REACTIVO.....	38
5.3. TIPOS DE EVALUACIONES SOPORTADOS.....	39
5.3.1. <i>Otras características del intérprete de Evaluaciones.</i> .....	40
5.4. EL SISTEMA BASADO EN REGLAS.....	40
5.4.1. <i>Definición del lenguaje de las reglas.</i> .....	40
5.4.2. <i>Compilador del lenguaje: Técnica de programación para reconocer las reglas.</i> .....	42
5.5. ALGORITMO DE RAZONAMIENTO .....	44
5.5.1. <i>Elección del tipo de razonamiento.</i> .....	45
5.5.2. <i>Descripción del algoritmo de razonamiento hacia adelante utilizado.</i> .....	45
5.5.3. <i>Metodología de factores de certeza para cálculo de valores de verdad.</i> .....	47
5.6. OTRAS UTILIDADES DEL SISTEMA BASADO EN REGLAS. ....	48
5.7. ALGORITMO EVALUADOR. ....	49
5.8. PROGRAMAS IMPLANTADOS DEL INTÉRPRETE (EVALUATEC). ....	51
<b>CAPÍTULO 6. RESULTADOS Y CONCLUSIONES. ....</b>	<b>57</b>
6.1. TESIS DE ESTE TRABAJO.....	57
6.2. APLICACIONES DEL SOFTWARE EVALUATEC.....	58
6.3. PRUEBAS REALIZADAS DEL SOFTWARE. ....	58
6.4. MEJORAS POSIBLES IDENTIFICADAS.....	60

6.5. CONCLUSIÓN FINAL .....	61
<b>REFERENCIAS:</b> .....	<b>63</b>
<b>BIBLIOGRAFÍA:</b> .....	<b>65</b>
<b>CONSULTAS EN INTERNET (WEB Y GOPHER).....</b>	<b>66</b>
<b>ANEXO A. DISEÑO DE LA BASE DE DATOS .....</b>	<b>68</b>
A1. ARCHIVO DE ALUMNOS.....	68
A2. ARCHIVO DE CARRERAS.....	69
A3. ARCHIVO DE GRUPOS. ....	69
A4. ARCHIVO DE MATERIAS.....	71
A5. ARCHIVO DE ALUMNO-GRUPO .....	72
A6. ARCHIVO DE AUTORES.....	74
A7. ARCHIVO DE MATERIA-TEMAS .....	74
A8. ARCHIVO DE REACTIVOS. ....	74
A9. ARCHIVO DE EVALUACIONES. ....	75
A10. ARCHIVO DE REACTIVOS-EVALUACIÓN.....	76
A11. ARCHIVO DE REGLAS .....	76
A12. ARCHIVO DE BITÁCORA .....	76
A13. ARCHIVO DE DETALLE BITÁCORA.....	77
<b>ANEXO B. REGLAS PARA RECOMENDACIONES PARA LA PRUEBA DE APTITUD ACADÉMICA.....</b>	<b>78</b>
<b>ANEXO C. CÓDIGO FUENTE DEL SOFTWARE.....</b>	<b>82</b>
C1. PREGLAS: PROGRAMA PARA CREAR LAS REGLAS.....	82
C1.1. <i>Freglas.frm</i> .....	82
C1.2. <i>Fprereg.frm</i> .....	85
C1.3. <i>Ffin2.frm</i> .....	85
C1.4. <i>Archreg.bas</i> .....	85
C1.5. <i>Compile.bas</i> .....	91
C2. SISEVAL: PROGRAMA EVALUADOR.....	106
C2.1. <i>Fprin.frm</i> .....	106
C2.2. <i>Fpregun.frm</i> .....	109
C2.3. <i>Ffin2.frm</i> .....	110
C2.4. <i>Archivos.bas</i> .....	110
C2.5. <i>Mrazona.bas</i> .....	118

## TABLA DE FIGURAS

FIG. 3.1. DIAGRAMA DE ENTIDAD RELACIÓN .....	20
FIG. 3.2. DIAGRAMA GENERAL DEL SISTEMA INTEGRAL DE EVALUACIÓN.....	22
FIG. 3.3. INTERFACE CON TABLAS DE OPERACIÓN DEL SISTEMA DE SERVICIOS ESCOLARES ....	23
FIG. 3.4. TEMAS Y SUBTEMAS (PROGRAMAS ANALÍTICOS).....	24
FIG. 3.5. DISEÑO DE REACTIVOS.....	25
FIG. 3.6. DISEÑO DE EVALUACIONES .....	26
FIG. 3.7. INTÉRPRETE DE EVALUACIONES .....	27
FIG. 3.8. ESTADÍSTICAS Y CONSULTAS .....	28
FIG. 4.1. COMPONENTES DE UN SISTEMA EXPERTO .....	32
FIG. 5.1. INTÉRPRETE DE REGLAS .....	36

## **CAPÍTULO 1. INTRODUCCIÓN**

Día con día la educación se ha vuelto más dependiente de la tecnología computacional. A partir de los años 70's, en algunas escuelas preparatorias, la computación pasó a ser una materia, y actualmente no sólo es una materia, sino una herramienta necesaria en cualquier institución educativa.

El uso de las computadoras en las escuelas ha ido evolucionando. Inicialmente el objetivo era que los alumnos aprendieran "computación", es decir prender y apagar una computadora, funciones básicas del sistema operativo, historia de las computadoras y en algunos casos, programar en algún lenguaje como BASIC. Actualmente la computación deja de ser un objeto de una clase especial para convertirse en una herramienta que apoya a cualquier materia y apoya también el proceso enseñanza-aprendizaje. Cualquier estudiante de secundaria o preparatoria actualmente debe conocer las bases y las aplicaciones principales de la computación, ya que es una herramienta muy útil para desarrollar tareas, trabajos, proyectos, consultar información, aprender a través de tutoriales, comunicación a través de redes, etc.

Un aspecto importante en el proceso de enseñanza-aprendizaje es la evaluación. El saber cómo los alumnos están desenvolviéndose en la adquisición de los conocimientos es relevante para poder medir el proceso de enseñanza-aprendizaje. Las evaluaciones han existido desde que se iniciaron las escuelas. Pero la aplicación de la tecnología computacional en la generación, aplicación y calificación de evaluaciones es realmente un aspecto innovador e importante para facilitar esta tarea.

Los ejemplos más importantes en la aplicación de la tecnología en esta área son los exámenes aplicados por computadora y los exámenes calificados por lectora óptica. Actualmente existen exámenes o evaluaciones por computadora, sobre todo de exámenes definidos para propósitos específicos como el SAT (Scholastic Aptitud Test), GRE (General Record Examination), entre otros. Este tipo de exámenes es llamado pruebas basadas en computadora (Computer Based Test). Sin embargo existen pocas herramientas para crear, diseñar y aplicar exámenes por computadora. Debido a esto surge la idea de crear un *software* con estas capacidades, y con la característica de poder dar recomendaciones específicas a cada alumno, dependiendo de su desempeño.

Los capítulos de este trabajo se presentan de la siguiente manera:

- 1) Introducción, antecedentes y visión general de la tecnología computacional que actualmente se aplica a la Educación y a la parte de evaluación o examinación.
- 2) Tipos de evaluaciones por computadora
- 3) Propuesta de un diseño de un Sistema Integral de Evaluación por Computadora.
- 4) Breve descripción de los sistemas expertos y de la representación del conocimiento basado en reglas.
- 5) Descripción del sistema computacional implantado llamado "Intérprete de Evaluaciones Automáticas y retroalimentación basada en un sistema de reglas" (EVALUATEC).
- 6) Resultados y conclusiones.

El primer capítulo es sólo un análisis muy general de la tecnología computacional aplicada a la Educación y las diferentes formas de evaluaciones por computadora. En el capítulo dos se describe brevemente los tipo de evaluaciones por computadora que existen. En el capítulo tres se propone un diseño de un sistema computacional de evaluación adaptado a la operación de una Universidad, basándose en la operación del Instituto Tecnológico y de Estudios Superiores de Monterrey (I.T.E.S.M.) Campus Querétaro. En el capítulo cuatro se da un panorama general de lo que es la

expertos, para explicar por qué se utilizó un sistema basado en reglas en el presente trabajo. En el capítulo cinco se detalla la explicación del desarrollo del intérprete de evaluaciones y retroalimentación basado en un sistema de reglas implantado en este trabajo. Y por último se explican resultados y conclusiones.

## 1.1. ANTECEDENTES.

En una Institución de Educación Superior siempre es de vital importancia estar a la vanguardia en lo que se refiere a Educación. Los avances tecnológicos en computación han sido aplicados de manera útil a la Educación. Como ejemplos podemos citar los avances en telecomunicaciones: internet, la red más grande del mundo; programas tutoriales como apoyos didácticos en todos los niveles; programas educativos vía satélite, consulta bibliográfica a través de información digital, etc.

**Basándose en la tecnología computacional y buscando una forma de aportar avances importantes a la Educación, surge la idea de desarrollar una herramienta práctica para realizar evaluaciones por computadora que sea capaz de retroalimentar automáticamente al examinado con recomendaciones particulares al avance de su aprendizaje.**

Una de las intenciones de este proyecto es el de contribuir a los esfuerzos que se han hecho en la aplicación de tecnología computacional al proceso de enseñanza-aprendizaje en el I.T.E.S.M. Campus Querétaro. De estos esfuerzos podemos mencionar:

i) "Proyecto CT": Programa institucional para utilizar el *software* CMU (Carnegie Mellon University). Tutor basado en un lenguaje especializado llamado CT para desarrollar

tutoriales en diversas áreas (1990). Este proyecto no tuvo éxito debido a la complejidad del lenguaje de programación y a los altos requerimientos de hardware.

ii) "Proyecto GT" [1]: *Software* de desarrollo de tutoriales en varias áreas. El 100% de la herramienta fue desarrollada por estudiantes la carrera de Ingeniería en Sistemas Computacionales (1991). Posteriormente apoyado por el Instituto de Efectividad del Grupo empresarial XABRE (IDEX). El desarrollo de este trabajo se detuvo cuando el IDEX se desintegró en 1992.

iii) "Proyecto SIMAT" [2]: Programación lógica para la construcción de tutores en Matemáticas. Primer lugar en la Categoría de Ciencias Computacionales en el Primer Concurso Nacional de Proyectos de Ingeniería y Ciencias (CONAPIC '93) Cetys Tijuana, B.C.

iv) "Proyecto HAEF" [3]: Herramienta de apoyo a la Enseñanza de la Física. Trabajo presentado en el Segundo Concurso Nacional de Proyectos de Ingeniería y Ciencias (CONAPIC '94) Cetys Tijuana, B.C.

v) "Proyecto ADSE" [4]: Herramienta para la Construcción de Sistemas Expertos. Desarrollo interno del departamento de Electrónica y Sistemas Computacionales del I.T.E.S.M. Campus Querétaro desde 1994. Este *Software* es utilizado para apoyar cursos de nivel licenciatura y desarrollar proyectos con la industria.

vi) "Plataforma Computacional para la aplicación de Tecnología Educativa": Herramienta para desarrollar material educativo computacional de apoyo a cursos, aplicando tecnología educativa. Proyecto en desarrollo, iniciado en Agosto de 1995 por el Departamento de Electrónica y Sistemas del I.T.E.S.M. Campus Querétaro.

## **1.2. MOTIVACIÓN DEL PROYECTO.**

### **1.2.1. Situación actual de los sistemas computacionales aplicados a la evaluación en el I.T.E.S.M. Campus Querétaro.**

Actualmente en el I.T.E.S.M. Campus Querétaro no existe ninguna herramienta computacional para crear, desarrollar y aplicar evaluaciones automáticas. Dentro de los proyectos importantes del Campus Querétaro está la elaboración de la carpeta del curso, en la cual están participando profesores de todo el Sistema I.T.E.S.M., y el objetivo es lograr tener clases de la mejor calidad. En este proyecto se tiene contemplado utilizar herramientas computacionales que apoyen el proceso de enseñanza-aprendizaje. Aún no se ha implantado oficialmente alguna herramienta para esto. Una de las áreas importantes para apoyar el proceso enseñanza-aprendizaje es la evaluación.

Actualmente el profesor puede saber la evaluación de sus alumnos aplicando un examen tradicional, en donde él lo califica, y posteriormente se retroalimenta con el alumno. De acuerdo a consultas académicas sería útil evaluar por computadora a un grupo de alumnos de un curso específico para identificar fallas, y retroalimentar al examinado con recomendaciones generadas automáticamente.

### **1.2.2. Sistemas actuales de evaluación por computadora.**

En el mercado existe actualmente mucho material educativo, pero la mayoría está enfocado a tutoriales desarrollados sobre temas específicos y pocos enfocados a evaluación. Algunos tutoriales poseen algún sistema de evaluación muy particular, el cual se basa en preguntas definidas de opción múltiple o de respuestas simples. En los últimos años ha tenido auge el desarrollo de tutoriales computacionales enfocados a la

capacitación de personal y al autoaprendizaje en áreas específicas. En la parte de generación de evaluaciones existen pocos sistemas computacionales.

Como ejemplos de sistemas o *software* relacionado con generación de evaluaciones por computadora podemos mencionar:

i) Sistema de Autoevaluación para el Aprendizaje (SAEA). Sistema creado por el I.T.E.S.M. para aplicar exámenes en el área de matemáticas y física. Este sistema genera exámenes escritos a partir de una base de reactivos de opción múltiple con cierta metodología. El examen no es aplicado por computadora. La calificación se realiza a través de lectora óptica y sólo registra la calificación obtenida, ya que el objetivo es dar la calificación oficial al alumno y unificar el nivel de conocimientos en todos los grupos del área de ciencias básicas.

ii) MICROCAT Testing System [5]: *Software* desarrollado desde 1984 por la empresa americana "Assessment Systems Corporation" ubicada en St. Paul, Main, U.S.A. Es un sistema para crear, administrar, calificar y analizar estadísticamente evaluaciones por computadora. Las evaluaciones pueden ser administradas en forma escrita o por medio de la computadora. Se basa en técnicas estadísticas para evaluar el desempeño y confiabilidad de los reactivos. Es un sistema completo, ya que soporta evaluaciones de varios tipos. Desarrollado para sistema operativo DOS. En el próximo capítulo se explican los tipos de evaluación que existen, y este paquete soporta todos estos tipos de evaluación.

iii) Learningspace de Lotus Notes: Herramienta desarrollada por "Lotus Development Corporation, an IBM subsidiary" desde 1995. Es un sistema muy completo, ya que no sólo es para crear, diseñar y aplicar evaluaciones, sino para diseñar todo un plan de un curso: tareas, proyectos, evaluaciones, exámenes rápidos, calendarización de actividades, grupos de discusión electrónicamente, trabajo en el salón de clases, etc. Dentro de la parte de generación de evaluaciones, está diseñado como una herramienta de seguimiento para el profesor. Desarrollado para Windows y ambientes Unix. Esta herramienta es sólo una parte de el sistema Lotus Notes.

iv) Test Designer Supreme: Desarrollado por la empresa americana "Super School *Software*" ubicada en Long Beach, California. Sistema diseñado como herramienta de

interesante sobre la generación de tecnologías para apoyar el proceso enseñanza-aprendizaje en una institución educativa.

v) pcPET (The Educator's Toolkit): *Software* diseñado para educadores profesionales. Incluye las siguientes funciones: planeador de clases, calendarización de evaluaciones y tareas, diseñador de evaluaciones, diseñador de cursos, planeador de calendario, seguimiento de desempeño de alumnos, reporteador, sistema de seguridad. Es un sistema desarrollado para sistema operativo DOS.

vi) The Incredible Quiz Machine: *Software* para crear, generar y aplicar exámenes por computadora. Es un sistema sencillo de generación de evaluaciones convencionales, en el cual los reactivos son de opción múltiple, falso/verdadero y respuestas cortas. Los alumnos pueden usar este programa como tarea, revisión o para estudio.

También existen exámenes por computadora para evaluar habilidades y conocimientos en áreas específicas. En 1947, se fundó una institución llamada "Educational Testing Service" (ETS) sin fines de lucro con el fin de realizar investigación en educación y en evaluación. Uno de los objetivos es el de crear, diseñar, administrar y calificar exámenes - tanto en forma tradicional, como por computadora - para clientes en educación, gobierno y empresas. ETS fue fundada por "The American Council on Education", "The Carnegie Foundation for the Advancement of Teaching" y por "the College Entrance Examination Board". El propósito de diseñar, desarrollar y aplicar evaluaciones es proveer una herramienta útil para permitir a individuos e instituciones demostrar logros, descubrir posibilidades y tomar decisiones. Esta institución, con sede en Princeton, New Jersey, anualmente administra más de 9 millones de exámenes en los Estados Unidos y en otros 180 países. Un ejemplo de los exámenes que diseña esta institución es el TOEFL, GRE, GMAT, SAT, entre otros.

Muchas de las evaluaciones desarrolladas por ETS utilizan tecnologías basadas en computadora para generar evaluaciones adaptativas, para retroalimentar inmediatamente con resultados, evaluar preguntas que no son fácil de evaluar por evaluaciones en papel, desarrollar habilidades, proveer ayuda interactiva como tutoriales y material de referencia, etc. Para soportar las evaluaciones basadas en computadora, ETS ha desarrollado un sistema propio y a parte ha establecido un

convenio con "Sylvan Learning Systems, Inc." para proveer una red nacional de centros de evaluaciones basadas en computadora. En enero de 1995 se liberaron evaluaciones internacionalmente en 21 países y para 1997 la red se expandirá para 170 países.

Actualmente las investigaciones de ETS se enfocan a mejorar evaluaciones, aprendizaje y habilidades a través de nuevas tecnologías y nuevas formas de evaluar.

Las nuevas tecnologías que se están estudiando buscan aplicarse para:

- proveer evaluaciones individualizadas
- permitir incluir dibujos o gráficos en las respuestas
- evaluar más ampliamente el trabajo de un estudiante
- proveer evaluaciones con multimedia
- métodos para evaluar más eficientemente respuestas del tipo ensayos o redacciones.

Entre los programas de ETS relacionados con evaluaciones basadas en el desempeño del examinado se encuentra:

- College Board's Advanced Placement Program (AP): evaluaciones basadas en el desempeño del estudiante (desde 1955)
- AEQUITAS (tm) : es un *software* para evaluar habilidades de oficina en procesos de capacitación y contratación para suavizar la transición de la escuela al trabajo.
- Algebridge (tm): sistema de tutorío y evaluación para ayudar a los estudiantes en la transición de la aritmética al álgebra.
- National Assessment of Educational Progress (NAEP): incluye preguntas con respuestas abiertas como ensayos y tareas que requieren herramientas como reglas, calculadoras, o material para experimentos científicos (ETS administra este proyecto bajo la dirección del departamento de Educación de los Estados Unidos de América).
- Pacesetter (tm): programa que coordina programas de evaluaciones, curriculum, y desarrollo de profesores para preparatorias y es patrocinado en conjunto por el ETS y por el College Board.

- PACKETS (r): tutorial y evaluaciones en matemáticas que se enfoca a la solución de problemas y pensamiento crítico.
- The Praxis Series: Evaluaciones profesionales para maestros principiantes. Es un nuevo programa que evalúa el desarrollo de profesores que consiste en tres programas. Estos miden habilidades académicas en lectura redacción y matemáticas, también miden conocimientos en el área y por último miden el desempeño en el salón de clases.

### **1.2.3. Conclusión de la situación planteada.**

Dentro de los sistemas de evaluación por computadora, se pueden diferenciar dos tipos:

1) Herramientas para generar evaluaciones para cualquier área. Estas herramientas son útiles para diseñar y aplicar evaluaciones de cualquier propósito. Se lleva un control de los resultados, y se da retroalimentación al examinado con la calificación instantánea, y al profesor o examinador para saber qué nivel tienen los evaluados.

2) Herramientas para evaluar conocimientos o habilidades en áreas específicas. Estas herramientas han sido muy útiles para aplicar evaluaciones estándar que son usadas por un gran número de instituciones, tanto educativas como de gobierno y empresariales. En éste tipo de evaluaciones el Educational Testing Service (ETS) es líder mundial.

Existen pocos sistemas de evaluación que se enfoquen a la parte de recomendaciones al examinado. Algunos sistemas tienen retroalimentación al examinado porque presentan gráficas estadísticas del desempeño en las evaluaciones contestadas. La retroalimentación directa, tomando como base el desempeño al contestar una evaluación, es un punto interesante y al parecer pocos sistemas lo realizan. Por esto es importante desarrollar un sistema capaz de generar evaluaciones por computadora adaptado a la operación de una Universidad y que tenga la capacidad

realizan. Por esto es importante desarrollar un sistema capaz de generar evaluaciones por computadora adaptado a la operación una Universidad y que tenga la capacidad para poder dar recomendaciones acertadas al examinado, como parte de su aprendizaje.

### **1.3. CONTRIBUCIONES.**

Las contribuciones del presente trabajo son:

- 1) Aportar el diseño e implantación de un sistema computacional para la aplicación de evaluaciones y retroalimentación automática con un enfoque genérico.
- 2) Aplicar técnicas de Inteligencia Artificial en la generación de recomendaciones en evaluaciones automáticas como apoyo al proceso de enseñanza-aprendizaje.
- 3) Crear la plataforma para implantar a corto plazo evaluaciones automáticas en el I.T.E.S.M. Campus Querétaro iniciando con una prueba piloto para el curso de preparación para el examen de admisión.
- 4) Contribuir de forma directa con los esfuerzos que apoyan la automatización del proceso enseñanza-aprendizaje en el I.T.E.S.M. Campus Querétaro.

## **CAPÍTULO 2. TIPOS DE EVALUACIONES POR COMPUTADORA**

Una evaluación es definida por el número y el contenido de sus reactivos, la manera en que los reactivos son seleccionados, y el método usado para registrar por computadora la calificación dependiendo de las respuestas del examinado. La forma en que se escogen los reactivos para una evaluación es llamada estrategia de selección de reactivos, y la forma en que se califican las respuestas es llamado método de calificación. Respecto a la estrategia de selección de reactivos las evaluaciones se clasifican en diferentes tipos.

En las evaluaciones administradas por computadora hay varios tipos de estrategias de selección de reactivos. Los más comunes son: convencional, adaptativos, de dominio referenciado personalizado, y simulación lógicamente armada [5].

a) Evaluación convencional: una evaluación convencional es construida por la selección de un conjunto fijo de reactivos para administrarse a un grupo de individuos. La calificación se obtiene contando el número de reactivos contestados correctamente. Tales evaluaciones son sencillas de diseñar y de calificar, pero tal vez no son muy eficientes si se aplica a un grupo de individuos que tienen diferentes niveles de conocimiento.

b) Evaluación adaptativa: las evaluaciones adaptadas son más eficientes para un grupo con diferentes niveles de habilidad. Estas evaluaciones están basadas en un concepto

simple: Más información puede ser obtenida de un reactivo si el reactivo concuerda con el nivel de habilidad del examinado. Para esto, la evaluación debe ir presentando los reactivos dependiendo si es correcta o no la respuesta. Este tipo de evaluaciones tiene complicaciones, ya que no se sabe el nivel de habilidad del examinado, y la calificación no se obtiene de sólo contar las respuestas correctas, ya que al aplicarlo en un grupo, los reactivos contestados por cada examinado son diferentes.

c) Evaluación de dominio referenciado personalizado: en este tipo de evaluación los reactivos son organizados en dominios, y los reactivos se administran en forma aleatoria a cada individuo. Se supone que cada reactivo es representativo del dominio, y que se obtendría el mismo resultado si se aplicaran todos los reactivos de ese dominio.

d) Evaluación de simulación lógicamente armada: los reactivos administrados en la evaluación dependen de una o más respuestas del examinado. Este tipo de evaluación se diferencia del adaptativo porque en el adaptativo los reactivos pueden ser contestados correcta o incorrectamente. En este tipo de evaluación, los reactivos no necesariamente son contestados correcta o incorrectamente, es como una elección, y dependiendo de la elección, se presentarán los siguientes reactivos. Otra diferencia es que en este tipo de evaluación no se busca que el reactivo tenga el mismo nivel de habilidad que el examinado. Un ejemplo de este tipo de evaluación sería un examen psicológico para identificar habilidades.

El presente trabajo soporta sólo el tipo de evaluación convencional, pero está diseñado para poder incorporar en un futuro el tipo de evaluación adaptativo y el de dominio referenciado personalizado.

### **CAPÍTULO 3. DISEÑO DE UN SISTEMA INTEGRAL DE EVALUACIÓN POR COMPUTADORA.**

El sistema propuesto está adaptado a una Universidad, basándose en la operación del I.T.E.S.M. Campus Querétaro, y está diseñado para generar, administrar, calificar y registrar evaluaciones por computadora que sean capaces, a su vez, de retroalimentar al examinado con recomendaciones acertadas.

El diseño de este proyecto se realizó después de analizar los sistemas de evaluación mencionados anteriormente, de consultar a algunos expertos en el área de educación del Campus Querétaro, y de analizar la operación del Campus en cuanto a los alumnos inscritos en grupos, materias impartidas por profesor, etc. Cabe mencionar que no se analizó el área de tecnología educativa. Para adaptar este sistema la parte de tecnología educativa, requiere de un esfuerzo importante que se escapa de las intenciones de este trabajo.

A continuación se mostrará un diagrama propuesto de las entidades relacionadas en la generación de evaluaciones por computadora. Este diagrama es la base conceptual del diseño de este proyecto. Posteriormente se explicará el diseño modular del sistema integral de evaluación.

### 3.1. DIAGRAMA DE ENTIDAD-RELACIÓN.

El presente diagrama explica la funcionalidad que tienen las entidades relacionadas con el sistema integral de evaluación.

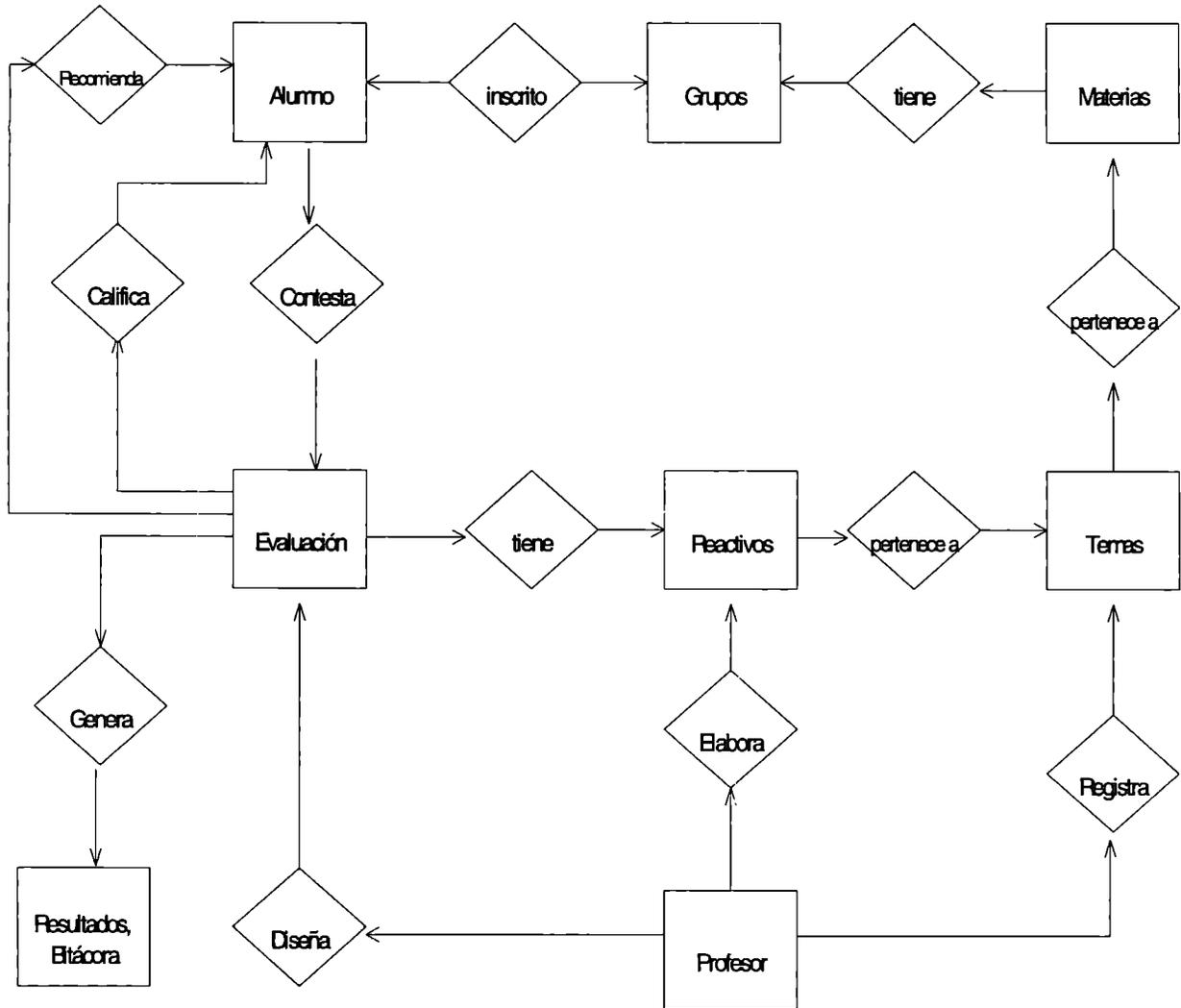


Fig. 3.1. Diagrama de Entidad Relación

Las entidades están representadas en rectángulo, y las relaciones en rombo.

Una interpretación del diagrama anterior sería por medio de las siguientes afirmaciones o declaraciones:

**BIBLIOTECA**



78992

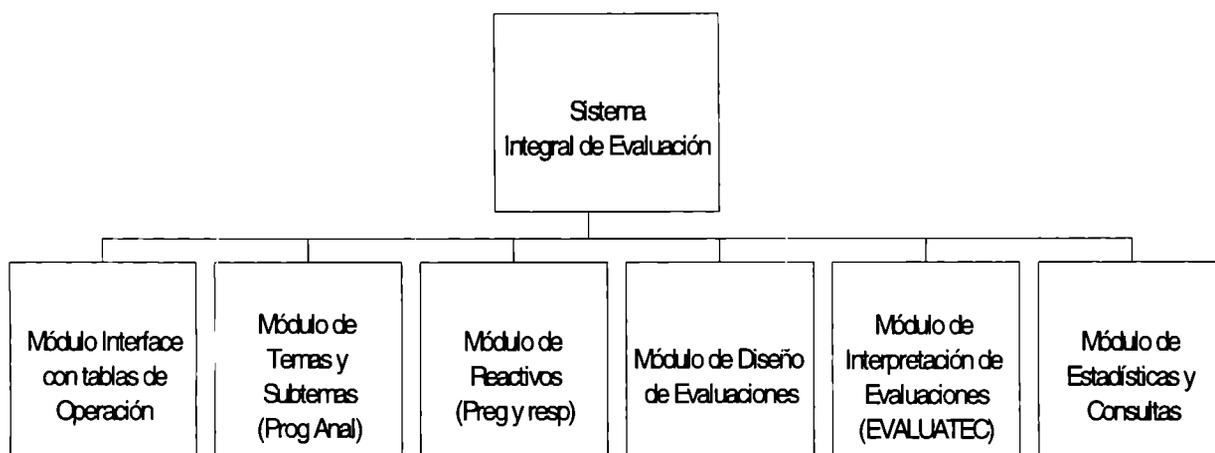
Un alumno está inscrito en grupos. Los grupos pertenecen a alguna materia. Dentro de cada materia se tiene definido un plan analítico o un conjunto de temas. Los reactivos son clasificados por temas o pertenecen a cierto tema. Una evaluación es un conjunto de reactivos. Un profesor elabora reactivos, registra los temas y diseña una evaluación. El alumno contesta la evaluación. La evaluación califica y da recomendaciones al alumno, y genera resultados en una bitácora.

A continuación se explicará el diseño de los módulos del sistema. El diseño de base de datos se encuentra en el Anexo A.

### **3.2. MÓDULOS DEL SISTEMA.**

Para poder implantar un sistema integral de evaluación que soporte las funciones expresadas gráficamente en el diagrama de entidad-relación, es necesario dividir el sistema en módulos. Todo el sistema funciona con una base de datos que fue diseñada para este trabajo (Anexo A). El diseño de la base de datos se realizó partiendo de un estudio minucioso del sistema computacional que actualmente opera en el departamento de Servicios Escolares del I.T.E.S.M. Campus Querétaro, del análisis de los diferentes sistemas de evaluación existentes y de la idea de involucrar un sistema basado en reglas para dar retroalimentación por medio de recomendaciones.

### **3.2.1. Diagrama general del Sistema Integral de Evaluación.**



**Fig. 3.2. Diagrama general del Sistema Integral de Evaluación**

Estos módulos son los que conforman todo el sistema integral de evaluación. Cabe mencionar que el módulo de interpretación de evaluaciones es el que se implantó en este proyecto.

### **3.2.2. Módulo I. Interface con tablas de operación del sistema de Servicios Escolares.**

#### **a) Descripción.**

Consta de programas de captura y de cargado de información de los archivos relacionados con el sistema actual de Servicios Escolares del I.T.E.S.M. Campus Querétaro. Para que el sistema integral de evaluación funcione de acuerdo a la operación del I.T.E.S.M., se debe de tener información de los grupos, alumnos, materias y profesores. Esta información es administrada por el departamento de Servicios Escolares.

b) Diagrama

Base de datos Sistema Escolar

Base de datos Sistema Evaluación

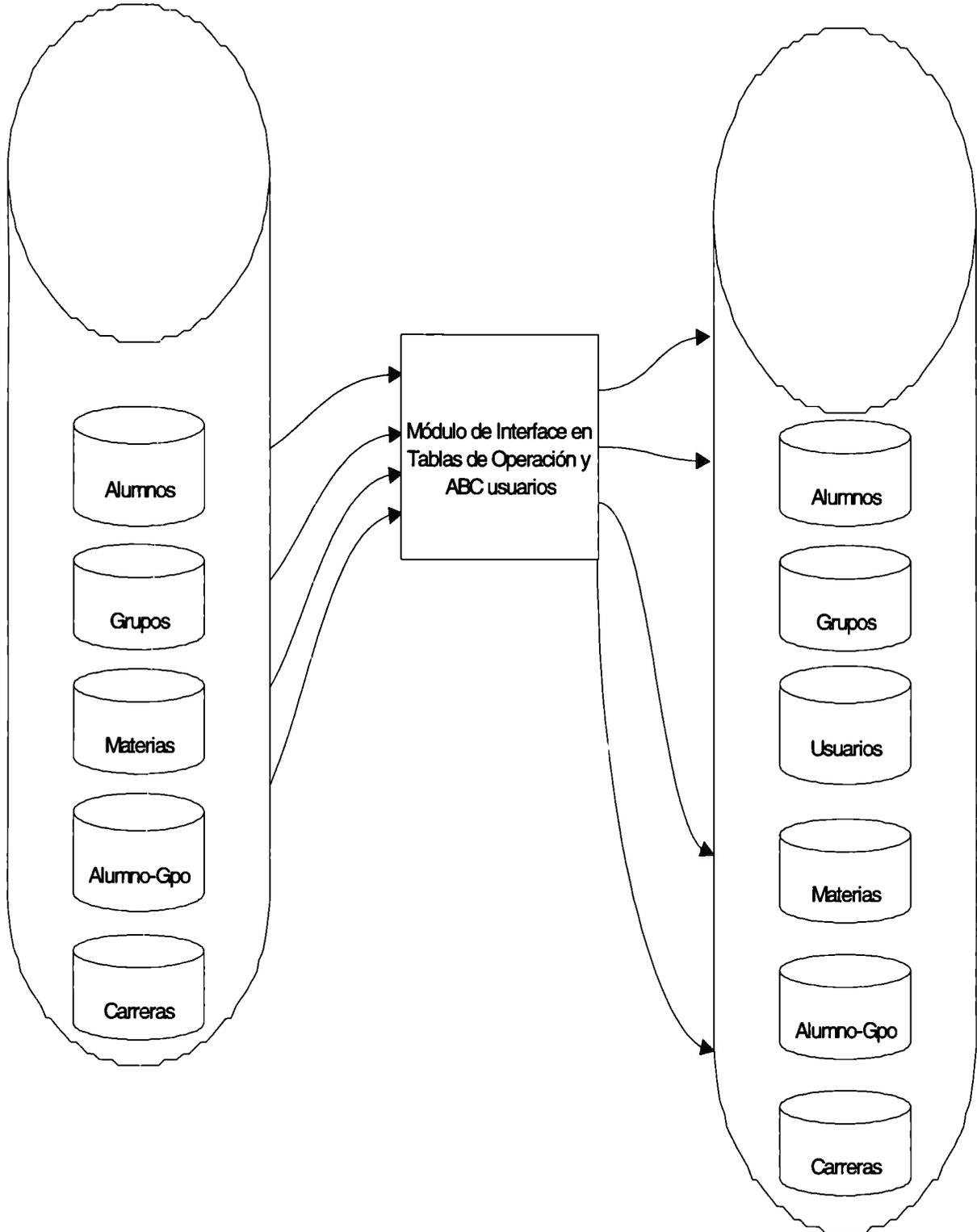


Fig. 3.3. Interface con tablas de operación del sistema de Servicios Escolares

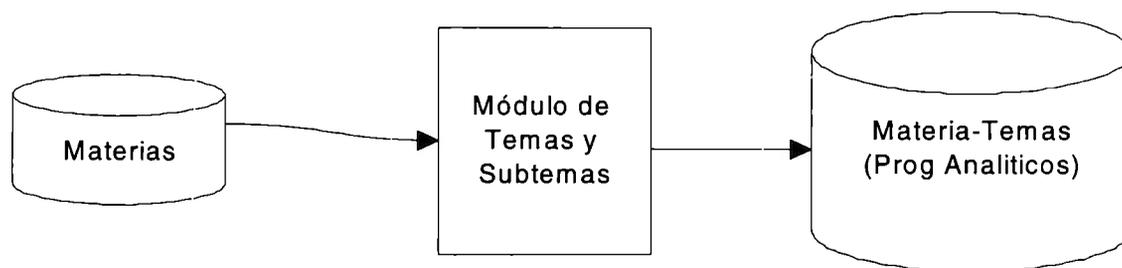
### **3.2.3. Módulo II. Registro de temas y subtemas (programas analíticos)**

#### a) Descripción:

Registro de temas y subtemas por materia (programas analíticos). En esta parte se podrá registrar todo un programa analítico: objetivos, requisitos, temas y subtemas. Las materias se codifican conforme a los planes de estudio del I.T.E.S.M., y los temas y subtemas se codifican numéricamente.

El objetivo de este módulo es tener registrados los programas analíticos de las materias, para que después exista una asociación con las preguntas y respuestas cuando se generen las evaluaciones. De esta forma el sistema podrá identificar a qué tema pertenece una pregunta, y podrá dar recomendaciones e identificar las áreas de fallas de los alumnos.

#### b) Diagrama:



**Fig. 3.4. Temas y subtemas (programas analíticos)**

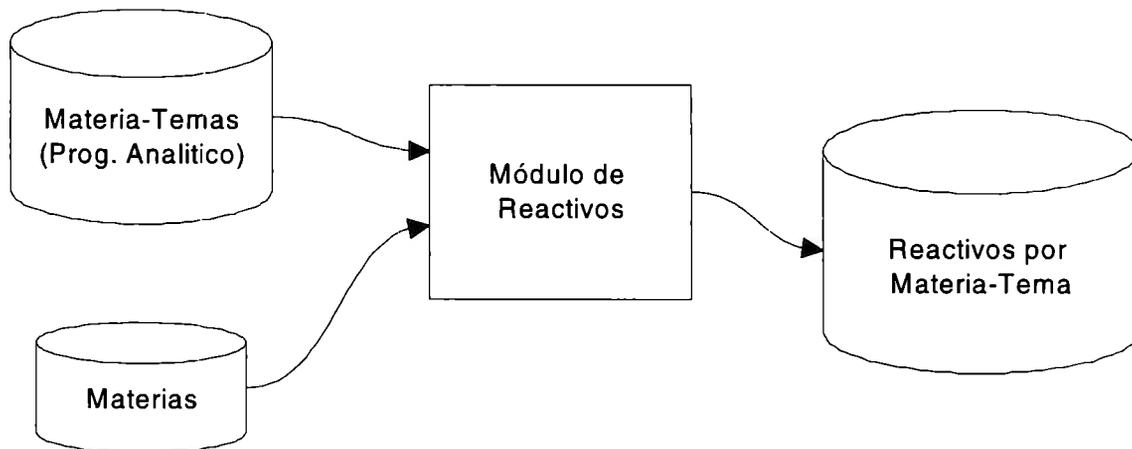
### **3.2.4. Módulo III. Diseño de reactivos (preguntas y respuestas)**

#### a) Descripción:

Este módulo es donde se van a registrar los reactivos (preguntas con respuesta), y se van a clasificar en temas o subtemas que existan dependiendo de los programas analíticos definidos en el módulo anterior. Cada reactivo tiene asociado un nivel de dificultad que puede ser actualizado de dos formas: a) automáticamente dependiendo de las respuestas que se vayan registrando; b) manualmente dependiendo del criterio

de la persona que crea el reactivo. Las preguntas se clasifican por temas y subtemas, y por materia. De tal manera que se tenga un banco de reactivos relacionados con materias y temas. Los tipos de reactivos soportados y sus características se explican en el capítulo cinco.

b) Diagrama:



**Fig. 3.5. Diseño de reactivos**

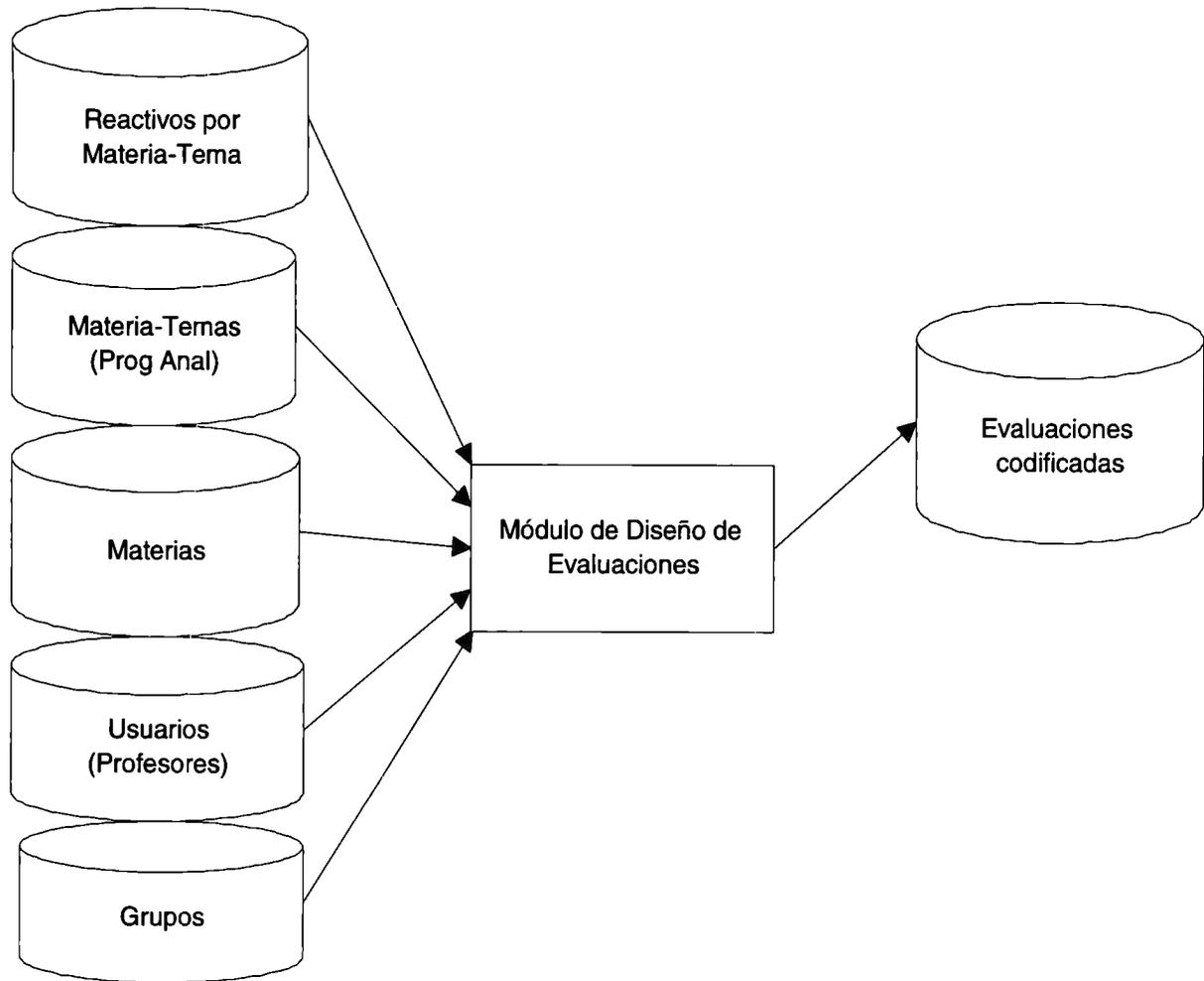
### **3.2.5. Módulo IV. Diseño de evaluaciones.**

a) Descripción:

Este módulo será capaz de diseñar una evaluación o examen utilizando el banco de reactivos y la base de datos de los temas y subtemas. Con este módulo se podrán diseñar diferentes evaluaciones, especificando: número de preguntas, recomendaciones por pregunta, tipo de evaluación, tiempo límite para contestar la evaluación, si es navegable (si se pueden ver todos los reactivos sin necesidad de responderlos), ponderación de cada reactivo, recomendaciones al final de la evaluación, etc. El sistema está diseñado para soportar tres de los cuatro tipos de evaluación explicados en el capítulo dos: a) convencional, b) adaptativo, c) de dominio referenciado personalizado. Las recomendaciones que se dan al final de la evaluación se editarán en forma de reglas de conocimiento, en donde se manejarán variables

como: respuestas correctas, respuestas erróneas, calificación, falla en temas (dependiendo de las respuestas erróneas) y recomendaciones. Estas reglas de conocimiento serán del tipo de un sistema experto basado en reglas, para que el sistema pueda dar recomendaciones en forma “Inteligente”.

b) Diagrama:



**Fig. 3.6. Diseño de evaluaciones**

### 3.2.6. Módulo V. Intérprete de evaluaciones.

#### a) Descripción:

Este módulo será el encargado de tomar el archivo de las evaluaciones codificadas y generará automáticamente la evaluación para que los alumnos de la materia-grupo especificada en la evaluación, la contesten. Cuando el alumno termina de contestar la evaluación, se da instantáneamente la calificación y se interpretan las reglas especificadas para dar las recomendaciones, utilizando un algoritmo de razonamiento que se explicará más adelante.

#### b) Diagrama

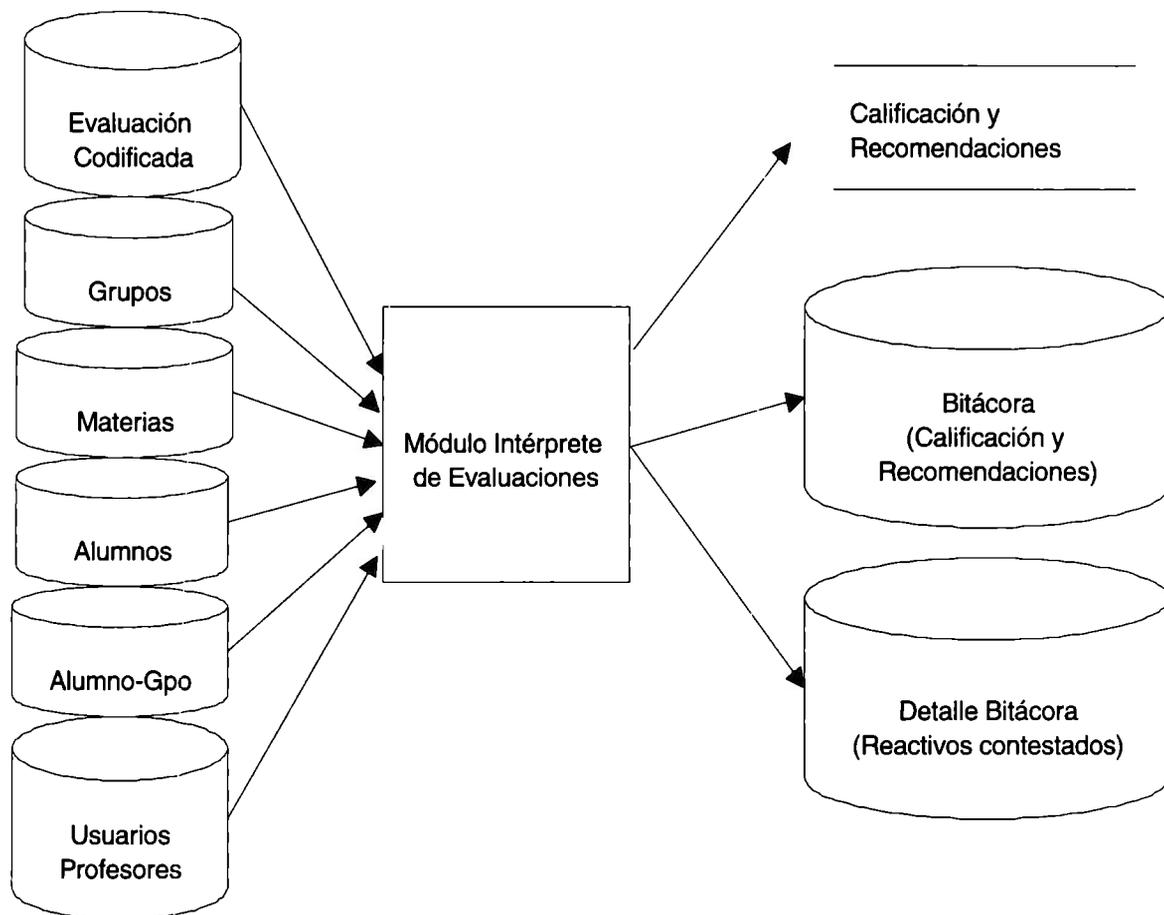


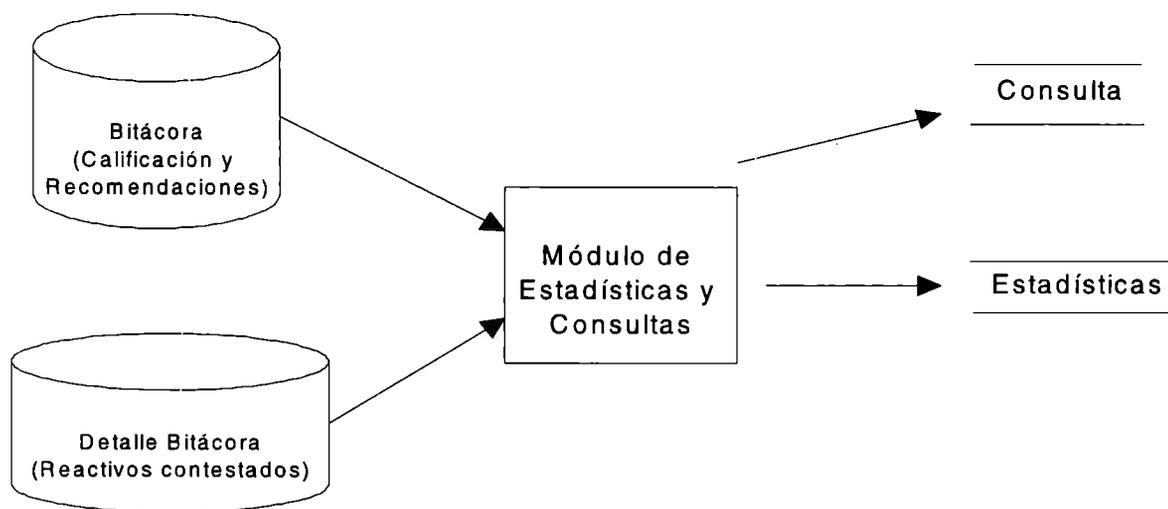
Fig. 3.7. Intérprete de evaluaciones

### 3.2.7. Módulo VI. Estadísticas y consultas

#### a) Descripción:

Este módulo se enfoca a llevar estadísticas de uso del sistema, registro de evaluaciones, reportes de desempeño por grupo, consultas de desempeño de alumnos, etc. Con este módulo un profesor puede tener retroalimentación con los resultados de los alumnos en las evaluaciones.

#### b) Diagrama:



**Fig. 3.8. Estadísticas y consultas**

## **CAPÍTULO 4. SISTEMAS EXPERTOS Y REPRESENTACIÓN DEL CONOCIMIENTO BASADO EN REGLAS**

El presente capítulo tiene por objetivo dar un panorama general de los sistemas expertos y de la representación del conocimiento basado en reglas. Es importante entrar en estos conceptos para entender la razón por la cual se utilizó un sistema basado en reglas en el presente trabajo.

### **4.1. DEFINICIÓN DE SISTEMA EXPERTO**

Un sistema experto es un programa por computadora en el que existe información organizada que representa un conocimiento en alguna área específica. Este conocimiento es utilizado para resolver algún problema o para realizar alguna tarea en forma similar a como lo haría un humano experto en el área.

El conocimiento se encuentra en una base de conocimientos. Para poder interpretar el conocimiento es necesario un razonamiento. En un sistema experto este razonamiento es hecho por un algoritmo llamado "motor de inferencias" que actúa sobre la base de conocimientos.

Los sistemas expertos son un tipo de Sistemas Basados en el Conocimiento, y éstos a su vez son una rama de la inteligencia artificial. Actualmente los sistemas

expertos han sido los más explotados de la inteligencia artificial como aplicaciones comerciales.

Para tener una idea más clara de lo que es un Sistema Experto, a continuación se transcriben algunas definiciones:

- 1) Sistema que usa el conocimiento humano capturado en una computadora para resolver problemas que ordinariamente requiere la experiencia de un experto humano [6].
- 2) Sistema que resuelve problemas que normalmente son solucionados por expertos humanos [7].
- 3) Aplicación informática que soluciona problemas complicados que de otra manera exigirían ampliamente la pericia humana. Para lograr esto, se simula el proceso de razonamiento humano mediante la aplicación específica de conocimientos y de inferencias [8].
- 4) Sistema de Inteligencia Artificial creado para resolver problemas en un dominio particular. Todo el conocimiento en un sistema experto es proveído por gente experta en el dominio [9].

La diferencia principal con un sistema convencional por computadora es que en el sistema convencional se manejan datos e instrucciones mezcladas en una secuencia de control definida para obtener resultados, y en un sistema experto se maneja información organizada lógicamente (base de conocimiento) y se utiliza la inferencia (motor de inferencia) sobre esta información para obtener resultados. Otra diferencia importante es que un sistema experto puede dar explicaciones de los resultados, y en un sistema convencional no.

La forma general como opera un sistema experto es:

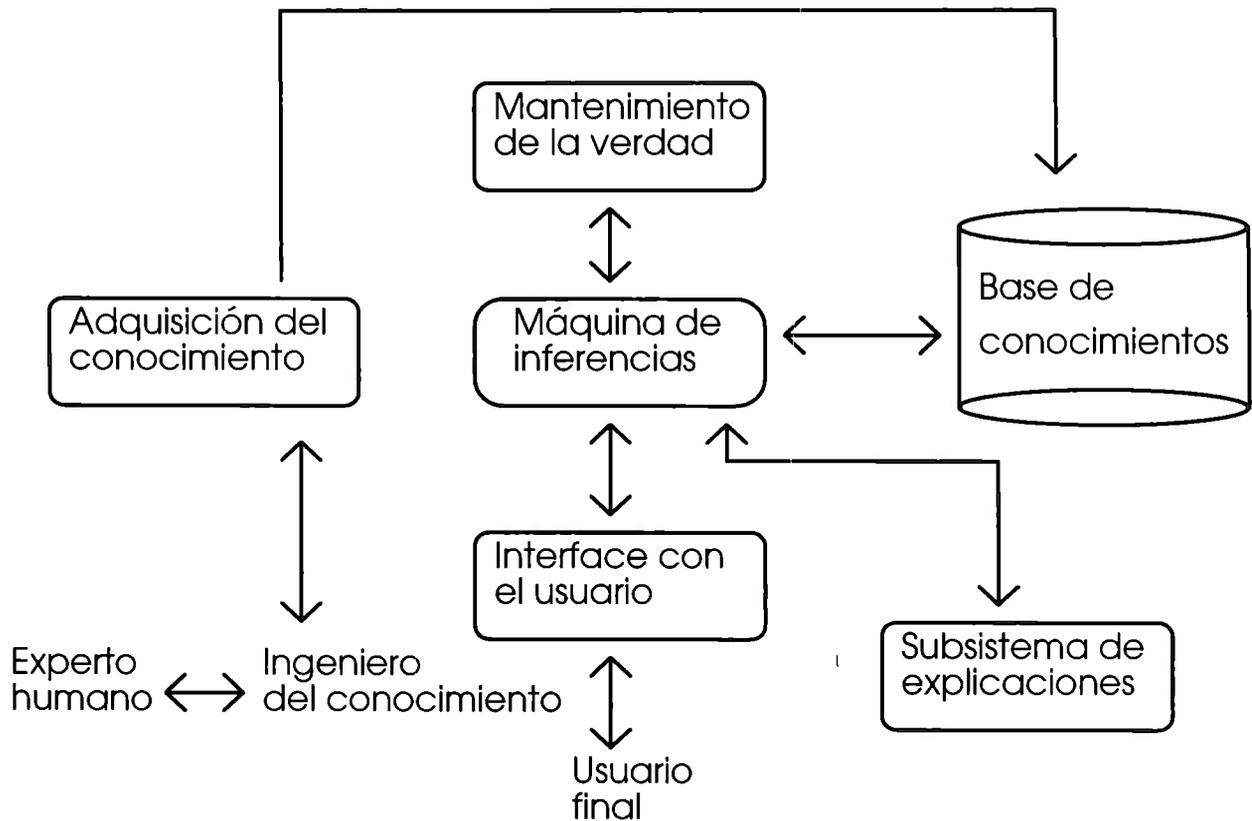
- a) Interacción con el usuario por medio de preguntas, o consultas del usuario al sistema.
- b) Interpretación de la base de conocimiento con un razonamiento. Para esto toma de entrada los datos proporcionados por el usuario.
- c) Despliegue de resultados, conclusiones o soluciones.
- d) Explicaciones de los resultados (opcional).

## 4.2. COMPONENTES DE UN SISTEMA EXPERTO.

Los componentes principales de un sistema experto y su descripción breve son:

- 1) Máquina de Inferencia: programa que realiza el razonamiento interpretando la base de conocimientos y teniendo como datos de entrada la información que le da el usuario.
- 2) Base de Conocimientos: información codificada que representa el conocimiento. Para esto existen varias formas de representar el conocimiento, que posteriormente se detallarán.
- 3) Interface de Usuario: es la forma en que el usuario final interactúa con el sistema experto.
- 4) Subsistema de Explicaciones: realiza explicaciones del por qué de los resultados obtenidos por el razonamiento.
- 5) Adquisición del conocimiento: proceso para adquirir y representar el conocimiento. Este proceso es realizado por: un ingeniero del conocimiento, el experto en el dominio e ingenieros en computación.
- 6) Mantenimiento de la verdad: subsistema que se encarga de verificar que no existan contradicciones o incongruencias en el conocimiento.

A continuación se detalla en forma gráfica la forma en que interactúan estos componentes:



**Fig. 4.1. Componentes de un sistema experto**

De estos componentes, nos enfocaremos sólo a la base de conocimientos y al motor de inferencias. A continuación se detallarán las formas de representar el conocimiento y posteriormente se explicará el funcionamiento de un motor de inferencias en un sistema basado en reglas.

### **4.3. REPRESENTACIÓN DEL CONOCIMIENTO.**

La base de conocimientos puede estar representada de varias maneras, dependiendo del tipo de conocimiento. El conocimiento puede componerse de hechos, conceptos, teorías, procedimientos y relaciones de representación del mundo real acerca de objetos, eventos, personas, etc. Las principales formas de representar el conocimiento son:

- 1) Redes semánticas: representación en gráficas dirigidas y etiquetadas y usualmente acíclicas. Cada vértice representa un concepto y los arcos representan las relaciones binarias entre conceptos.
- 2) Marcos o Frames: estructura de datos para representar un modelo mental de una situación típica o un objeto. El conocimiento sobre un objeto o evento está almacenado junto en memoria como una unidad (marco). Un marco es una colección de atributos, valores y relaciones con otros marcos, de tal forma que la información se encuentra jerarquizada.
- 3) Representación lógica: es la más antigua y se basa en declarar hechos o premisas y aplicar razonamiento lógico para obtener conclusiones o inferencias. Existen dos formas para computarizar la representación lógica: lógica proposicional (o cálculo proposicional) y lógica de predicados (o cálculo de predicados).
- 4) Reglas de producción: el conocimiento se representa a través de reglas de producción de la forma de condición-acción: "SI condición o premisa ENTONCES acción o conclusión".

De todas estas formas de representar el conocimiento, las reglas de producción han sido las más exitosas en los sistemas expertos. La mayoría de los sistemas expertos actuales están basado en un sistema de reglas.

#### **4.4. TIPOS DE RAZONAMIENTO EN UN SISTEMA BASADO EN REGLAS.**

El objetivo de un algoritmo de razonamiento en un sistema basado en reglas es obtener conclusiones adecuadas sobre las reglas definidas, para llegar a resolver un problema. Otra forma de verlo es encontrar una ruta por la cual se llega de un estado inicial, a un estado meta, y finalmente descubrir el estado meta. Existen dos formas de realizar razonamiento en un sistema basado en reglas:

- a) Razonamiento hacia atrás o dirigido por metas "Backward Chaining" (razonamiento inductivo)

b) Razonamiento hacia adelante o dirigido por datos "Forward Chaining" (razonamiento deductivo)

El razonamiento hacia atrás es útil para solución de problemas dirigidos por metas. En otras palabras es útil cuando está definida o se puede proponer una meta. En este razonamiento se empieza de una expectativa (meta) de qué es lo que pasa (hipótesis), y después se localizan las evidencias que soportan o contradicen la expectativa.

Por ejemplo, un sistema de consulta basado en reglas, podría usar razonamiento hacia atrás para poder responder preguntas del usuario, ya que empieza a hacer la búsqueda a partir de la pregunta que se hace.

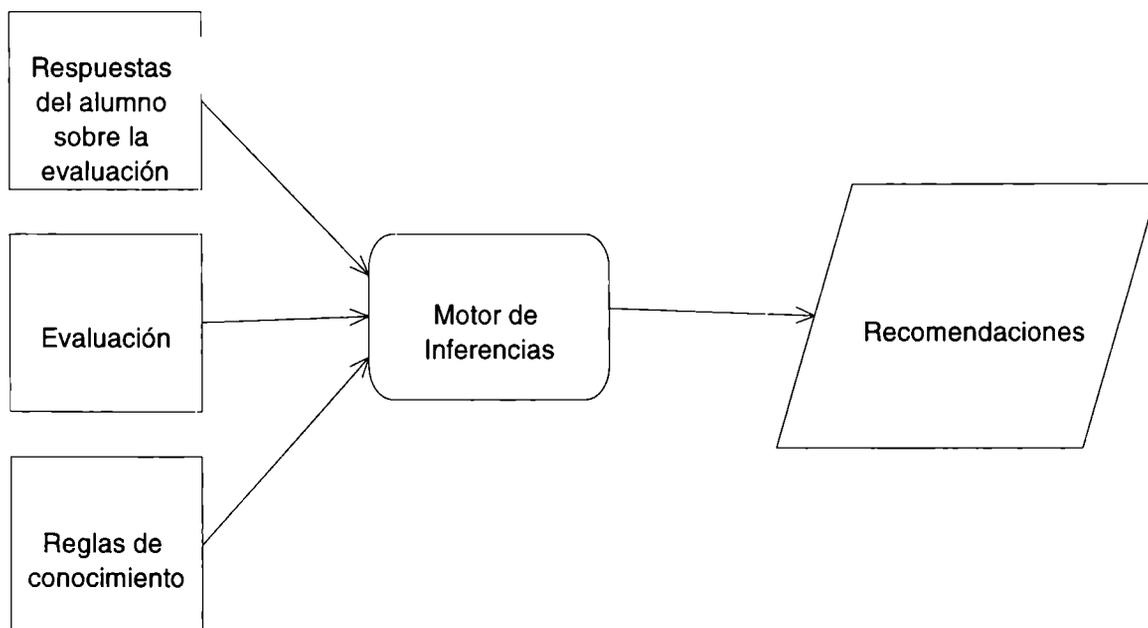
El razonamiento hacia adelante es útil para problemas en los cuales no se sabe exactamente a qué meta se quiere llegar, pero sí se sabe el estado inicial; sólo se realizan búsquedas sobre las evidencias y se obtienen conclusiones a partir de los datos iniciales. Se empieza de la información que se tiene disponible como entrada, y después se trata de llegar a conclusiones.

## **CAPÍTULO 5. INTÉRPRETE DE EVALUACIONES Y RETROALIMENTACIÓN BASADA EN UN SISTEMA DE REGLAS.**

Este intérprete de evaluaciones es la implantación del módulo V llamado Módulo de interpretación de evaluaciones, que se explicó en el capítulo tres.

La función principal es la de interpretar una evaluación codificada, que es creada en el módulo de diseño de evaluaciones, y ejecutarla para que un alumno la conteste. Es importante mencionar que este intérprete supone que el módulo de diseño de reactivos y el módulo de diseño de evaluaciones crean las evaluaciones en los archivos EVALUACION, EVAL-REAC y REACTIVO descritos en el Anexo A.

La característica más importante de este intérprete de evaluaciones es que retroalimenta al alumno con recomendaciones de acuerdo a su desempeño al contestarla. La función del intérprete se puede representar por la siguiente figura:



**Fig. 5.1. Intérprete de reglas**

El motor de inferencias es un algoritmo que realiza un razonamiento sobre las reglas definidas y tomando en cuenta las respuestas del examinado. Al final, las conclusiones son las recomendaciones.

En el programa evaluador implantado, se manejan dos tipos de recomendaciones. Uno es dar una recomendación por cada reactivo que no se conteste correctamente. En esta recomendación no se aplica ningún razonamiento, sólo se muestra una recomendación fija si se falla. El otro tipo de recomendación es la que se da al final de contestar toda la evaluación. En esta es donde se aplica el motor de inferencias. A continuación se detalla más sobre estos dos tipos de recomendaciones.

Las recomendaciones se pueden dar por reactivos y/o al final de la evaluación:

- a) Por reactivo: se da alguna recomendación específica si contestó bien o mal el reactivo. Al momento de diseñar la evaluación, el profesor indica qué recomendación se dará si falla, y cuál si acierta. Es posible que no se dé ninguna recomendación si el profesor así lo desea.
- b) Al final de la evaluación: las recomendaciones se dan una vez que el alumno terminó de contestar todos los reactivos. Es aquí donde se aplica el sistema basado en reglas,

para especificar en qué condiciones se darán las recomendaciones. Para lograr que se den las recomendaciones conforme al desempeño del alumno, se interpreta el conjunto de reglas de conocimiento, el cual es creado por el profesor en un programa diseñado para esto. Este conjunto de reglas de conocimiento tiene la estructura de una premisa lógica: Si <condición> entonces <conclusión/recomendación>. La estructura ó lenguaje de las reglas de conocimiento se explicará en el subcapítulo 5.4.

Adicionalmente, se le da la calificación final al alumno y se registra su resultado y sus respuestas en un una bitácora, la cual consta de dos archivos: uno es la bitácora del resultado final de la evaluación, y otra es el detalle de las respuestas de cada reactivo.

Es importante mencionar que esta herramienta de *software* no se podrá aplicar a cualquier área, debido a que existen habilidades y conocimiento que no se pueden medir con una evaluación computacional, sino con la interacción directa del profesor-alumno.

En los siguientes subcapítulos se explicará más a detalle el funcionamiento y características de este intérprete.

## **5.1. TIPOS DE REACTIVOS SOPORTADOS EN UNA EVALUACIÓN.**

El presente intérprete está diseñado para soportar los siguientes tipos de reactivos dentro de una evaluación:

1) De opciones múltiples generadas: Para este tipo de reactivo se generan 5 diferentes respuestas (sólo una correcta) tomando en cuenta un rango de valores que es definido por un parámetro dentro del reactivo. Normalmente este tipo es usado cuando las respuestas son solucionadas por alguna fórmula matemática, la cual es interpretada al momento de desplegar el reactivo.

2) De opciones múltiples: Se despliegan 5 opciones ya definidas en forma aleatoria, en las cuales sólo una es la correcta. Físicamente, en el archivo de reactivos, la opción

correcta siempre es la primera (opción A), pero al ser presentada se coloca en forma aleatoria. Esto permite evitar tener que especificar cuál es la correcta.

3) De valor específico: Espera que se lea un valor, ya sea numérico o alfanumérico. En este tipo, la respuesta debe de escribirse exactamente igual a la correcta.

4) Falso/Verdadero: Se despliegan 2 opciones: Falso y Verdadero.

En la implantación del intérprete, sólo son soportados los tipos 2, 3 y 4.

## 5.2. INFORMACIÓN PRESENTADA EN CADA REACTIVO.

Al presentar un reactivo en una evaluación aparece la siguiente información:

1) Encabezado: información general referente a la evaluación a la que pertenece el reactivo:

- a) Identificador de la evaluación
- b) Profesor del curso al que pertenece la evaluación
- c) Nombre del curso al que pertenece la evaluación
- d) Total de preguntas de la evaluación
- e) Tiempo límite en horas:minutos:segundos (sólo en caso de ser con tiempo limitado).
- f) Tiempo transcurrido en horas:minutos:segundos

2) Redacción de la pregunta

3) Gráfico o imagen: en caso de que el reactivo tenga asociado algún gráfico o imagen, este es presentado. Los gráficos soportados son estándar de Windows y son del tipo bitmat (.BMP), icon(.ICO) y metafile (.WMF).

4) Texto de la respuesta. Este es variable dependiendo del tipo de reactivo: si es de opción múltiple, aparecerán 5 opciones; en caso de ser falso/verdadero, aparecerán las dos opciones de falso y verdadero; en caso de ser de valor específico, sólo aparece un espacio para contestarlo.

5) Ayuda: botón de ayuda que funciona sólo si al crear el reactivo, se especificó algún tipo de ayuda. En caso de haber especificado la ayuda, se desplegará un texto con la ayuda al seleccionar el botón.

6) Tutorial: botón que funciona sólo si al crear el reactivo, se especificó algún programa ejecutable asociado al reactivo. En caso de haberlo especificado, se ejecutará al momento de seleccionar este botón.

7) Referencias: botón que funciona sólo si al crear el reactivo, se especificó alguna referencia. En caso haber especificado la referencia, se desplegará un texto al momento de seleccionar el botón.

### **5.3. TIPOS DE EVALUACIONES SOPORTADOS.**

El intérprete está diseñado para que se puedan tener los siguientes dos tipos de evaluaciones:

- a) Aleatorias
- b) Dirigidas

Las aleatorias son evaluaciones del tipo "Evaluación de dominio referenciado personalizado" explicado en el capítulo dos, en el cual se seleccionan al azar los reactivos al momento de contestar la evaluación. La selección puede hacerse sobre reactivos de algún tema en específico. Este tipo de evaluación no es soportado por este intérprete pero está diseñado para implantarse en un futuro.

Las dirigidas son evaluaciones del tipo "Evaluación convencional" explicado en el capítulo dos, en el cual los reactivos están definidos, por lo que cada vez que se ejecute, siempre se preguntarán los mismos reactivos.

El intérprete está diseñado para soportar los dos tipos de evaluación, pero sólo se implantó el tipo de evaluación dirigida.

### **5.3.1. Otras características del intérprete de Evaluaciones.**

Algunas otras características adicionales a las ya explicadas son:

1) Tiempo: se puede especificar si una evaluación tiene un tiempo limitado para contestarla, o si es de tiempo ilimitado

2) Navegación: se puede especificar que dentro de la evaluación se pueda navegar en todos los reactivos (hacia adelante o hacia atrás) sin necesidad de contestar cada uno, o de ir avanzando uno a uno conforme se contesta cada reactivo.

## **5.4. EL SISTEMA BASADO EN REGLAS.**

### **5.4.1. Definición del lenguaje de las reglas.**

Para poder dar recomendaciones automáticamente dependiendo del desempeño del alumno, es necesario definir un lenguaje en donde el profesor o el diseñador de la evaluación defina claramente en qué condiciones se darán recomendaciones.

El lenguaje de definición de las reglas de conocimiento tiene dos tipos de instrucciones. Uno es para especificar una condición y conclusión o recomendación; y el otro es para especificar preguntas que se harán al examinado al finalizar la evaluación y que se tomarán en cuenta para la recomendación final. En resumen el lenguaje se basa en las siguientes instrucciones:

- a) SI <Condición> / <Proposición> ENTONCES <Proposición > / RECOMENDACION 'Texto'
- b) Pregunta <Proposición> 'Texto de la pregunta'

La primera instrucción es la más utilizada en un sistema de reglas. La segunda instrucción se adicionó para poder hacer preguntas al examinado después de haber

contestado toda la evaluación. En estas preguntas el examinado solo puede contestar sí o no, y se presentan sólo si se requiere saber la pregunta para concluir algo.

La <Condición> puede ser:

Reactivo <IdReactivo> es correcto/incorrecto

Calificación es [>/</>=/<=/=] <NN.NN>

Aciertos (<Idreactivo> [[,<Idreactivo>][,<Idreactivo>][,... ] ) [>/</>=/<=/=] <NN>

En donde <IdReactivo> es el número o identificador del reactivo. Cada reactivo tiene un número o identificador único.

La <Proposición> es una palabra que se interpreta como una proposición que tiene un valor de verdad y es calculado al momento del razonamiento que hará el intérprete.

La condición Aciertos se utiliza para especificar la condición de cuántos aciertos tuvo de un conjunto de reactivos dados. Este tipo de condición normalmente no se encuentra en un sistema basado en reglas, pero se adicionó porque es muy práctico para este tipo de sistema.

Representando el lenguaje mediante una gramática libre de contexto y sin recursión a la izquierda, tenemos:

S --> si CONDICION entonces CONC S

S --> Pregunta PROPOSICION 'Texto alfanumérico' S

S --> Ø

CONDICION --> CONDICION Y

CONDICION --> CONDICION O

CONDICION Y --> COND y CONDICION Y

CONDICION Y --> COND

CONDICION O --> COND O CONDICION O

CONDICION O --> COND

COND --> Reactivo IDREACTIVO es CORRECCION

COND --> Calificación es OPERADOR NUMERO

COND --> Aciertos (IDREACTIVO [[,IDREACTIVO],[IDREACTIVO],... ])OPERADOR NN  
COND --> PROPOSICION  
CORRECCION--> incorrecto  
CORRECCION--> correcto  
IDREACTIVO--> "String alfanumérico de 10 caracteres como máximo"  
CONC--> PROPOSICION CERTEZA  
CONC--> Recomendación 'Texto alfanumérico'  
PROPOSICION--> NEGACION "Variable alfanumérica"  
NEGACION --> No  
NEGACION -->  $\emptyset$   
CERTEZA -> (NN %) /  $\emptyset$   
NN --> "Número entero"  
OPERADOR --> >/</>=/<=/</>  
NUMERO--> "Número real"

*Nota:  $\emptyset$  indica vacío.*

#### **5.4.2. Compilador del lenguaje: Técnica de programación para reconocer las reglas.**

Las reglas definidas a través del lenguaje debe ser analizado por un compilador, el cual utiliza una técnica de programación para reconocer las reglas. La técnica elegida es un algoritmo analizador de descenso recursivo ("Top Down") [10]. Algunos autores le llaman intérprete a este tipo de compiladores. Lo que hace este compilador es leer las reglas de la evaluación y generar una estructura en memoria, en la cual se separan las condiciones y conclusiones. Esta estructura en memoria será la entrada para el algoritmo de razonamiento.

La estructura en memoria que deja el compilador se compone de dos arreglos: un arreglo unidimensional para las conclusiones, y un arreglo bidimensional para las condiciones:

Conclusiones

Condiciones

Conc 1	Condición 11	Condición 12	Condición 1n
Conc 2	Condición 21	Condición 22	Condición 2n
Conc 3	Condición 31	Condición 32	Condición 3n
Conc 4	Condición 41	Condición 42	Condición 4n
...	...	...	...
Conc m	Condición m1	Condición m2	Condición mn

Esta estructura representa a un conjunto de m reglas con n condiciones cada regla.

Estos arreglos representan al siguiente conjunto de reglas:

Si condición 11 y/o Condición 12 y/o Condición 1n entonces Conc 1

Si condición 21 y/o Condición 22 y/o Condición 2n entonces Conc 2

Si condición 31 y/o Condición 32 y/o Condición 3n entonces Conc 3

Si condición 41 y/o Condición 42 y/o Condición 4n entonces Conc 4

Si condición m1 y/o Condición m2 y/o Condición mn entonces Conc m

De esta manera se pueden tener las reglas y condiciones que se deseen. Para este proyecto se definió tener un máximo de 500 reglas con un máximo de 30 condiciones en cada regla.

Cada elemento de los arreglos contiene información sobre la conclusión o condición.

Para las conclusiones se definieron los siguientes campos:

- Nombre\_prop: A50 : Nombre de la conclusión
- Valor\_verdad: N2 : Valor de verdad (-100 es falso y 100 verdadero)
- ul\_valor\_verdad: N2: Último valor de verdad (-100 falso y 100 verdadero).

Este se usa cuando hay más de una proposición que concluye lo mismo.

- Visitado: Booleano: Si la proposición fue visitada. Esta es una bandera que se utiliza al hacer el razonamiento.

- Concluido: Booleano: Si ya fue concluida la conclusión. Esta es una bandera usada al hacer el razonamiento.

- Conjunción: Booleano: Si las condiciones están anidadas con conjunciones o con disyunciones. No se pueden combinar conjunciones con disyunciones.
- TipoConc: N1: Tipo de conclusión: 0=Recomendación, 1=Proposición
- Valasig: N3: Valor de verdad asignado al definir la regla
- Negación: Booleano: Si se concluye la negación de la proposición.

Para las condiciones se definieron los siguientes campos:

- Nombre\_prop: A50 : Nombre de la condición
- Valor\_verdad: N2 : Valor de verdad (-100 es falso y 100 verdadero)
- ul\_valor\_verdad: N2: Último valor de verdad (-100 falso y 100 verdadero). Este se usa cuando hay más de una proposición que concluye lo mismo.
- Visitado: Booleano: Si la proposición fue visitada. Esta es una bandera que se utiliza al hacer el razonamiento.
- Concluido: Booleano: Si ya fue concluida la conclusión. Esta es una bandera usada al hacer el razonamiento.
- Negación: Booleano: Si se concluye la negación de la proposición.
- Conjunción: Booleano: Si las condiciones están anidadas con conjunciones o con disyunciones. No se pueden combinar conjunciones con disyunciones.
- Tipo: N1: Tipo de condición: 0=hecho (del tipo si reactivo n es correcto), 1=Proposición y 2 = Pregunta

*Notación de los campos:*

*An = Alfanumérico de n caracteres*

*Nn = Numérico de n dígitos*

*Booleano = de dos valores, falso o verdadero.*

## **5.5. ALGORITMO DE RAZONAMIENTO.**

El presente algoritmo se aplica al final de la evaluación. Toma de entrada la estructura en memoria que deja el compilador de las reglas de la evaluación y las

respuestas que el alumno dio a cada reactivo. La salida es el conjunto de recomendaciones al alumno como retroalimentación. A continuación se explicará la razón del algoritmo seleccionado.

### **5.5.1. Elección del tipo de razonamiento.**

Con respecto a la definición de las formas de razonamiento en un sistema basado en reglas, descritas en el capítulo cuatro, el razonamiento que se eligió en este trabajo fue el razonamiento hacia adelante o "Forward Chaining". La razón de esta elección es debido a que el objetivo es encontrar qué recomendaciones se van a dar, y no hay un objetivo en específico al cual el razonamiento tenga que llegar. No se tiene definida la meta donde llegar. El razonamiento hacia atrás o "Backward Chaining" no sería el más adecuado ya que se tendría que definir una meta que fuera llegar a una recomendación.

### **5.5.2. Descripción del algoritmo de razonamiento hacia adelante utilizado.**

A grandes rasgos, el algoritmo de razonamiento es el siguiente:

- a) Leer la primera regla (el primer registro de los arreglos de condiciones y conclusiones).
- c) Si todas las condiciones tienen valor de verdad o pueden ser calculados entonces:
  - Si la conclusión es una recomendación y si tiene un valor de verdad mayor a 20, desplegarla con el valor de verdad calculado por la metodología de factores de certeza.
  - Si la conclusión es una proposición, calcular el valor de verdad por la metodología de factores de certeza.
- d) Si se leyó la última regla y no se han concluido todas las recomendaciones ir a a)
- e) Si no es la última regla, leer la siguiente e ir a c)

Este ciclo termina cuando son concluidas todas las recomendaciones o cuando al realizar más de un ciclo, los valores de verdad de las conclusiones y de las recomendaciones no cambian.

La metodología de factores de certeza es utilizada para manejar incertidumbre. En este caso, se manejará incertidumbre para dar recomendaciones con alguna probabilidad de que la recomendación sea la más adecuada. Cada proposición o recomendación tiene asignado un valor de verdad, o factor de certeza. El rango de un factor de certeza va de -100 a 100. 100 representa completa certeza de que una proposición es verdadera o que la recomendación es la más adecuada. 20 representa un mínimo para tener evidencia que sea verdadero. 0 representa no evidencia ni a favor ni en contra. Por lo que las recomendaciones que se darán al final de una evaluación serán solo aquellas que alcanzaron un valor de certeza igual o mayor a 20. Los factores negativos representan evidencia de que la proposición es falsa o que la recomendación no es la adecuada. -100 representa completa certeza de que una proposición es falsa o que la recomendación no es nada adecuada. -20 representa un mínimo para tener evidencia que es falso.

### **5.5.3. Metodología de factores de certeza para cálculo de valores de verdad.**

Un factor de certeza es un mecanismo relativamente informal para cuantificar el grado al cual, fundamentado en la presencia de un conjunto dado de evidencias, creemos o no en una conclusión dada. Los factores de certeza se han aplicado ampliamente en dominios donde las evidencias se van recogiendo en forma incremental [8].

El concepto de factores de certeza se desarrolló para uno de los primeros sistemas expertos llamado MYCIN y se ha empleado exitosamente en muchos otros sistemas [8].

En este caso se utilizó la metodología de factores de certeza propuesta por el Ing. Francisco Cantú [11] con algunas adaptaciones que se explican a continuación:

Para calcular el valor de verdad de una proposición o una recomendación, se tiene que trabajar con los factores de certeza asignados a cada elemento de la premisa de una regla.

Para calcular el valor de verdad de un elemento de una premisa que sea una condición del tipo <si reactivo 00000001 es correcto>, el valor de verdad será 100 si se contestó bien el reactivo 00000001, y -100 si no se contestó bien. Para una condición del tipo <si acierto (000000001, 000000002, ...) = n>, el valor de verdad será 100 si el número de aciertos del conjunto de reactivos especificados cumple con la condición dada, y -100 en caso contrario.

Para calcular el valor de verdad de una proposición que está como conclusión, se debe hacer el cálculo del valor de verdad de la siguiente forma:

a) Para una conjunción de proposiciones, el valor de verdad es el mínimo de los factores de certeza de las premisas, siempre y cuando el mínimo sea mayor a cero. En caso de que el factor de certeza mínimo es menor que cero, se asignará cero de factor de certeza a la conclusión.

b) Para una disyunción de proposiciones, el valor de verdad es el máximo de los factores de certeza de las premisas, siempre y cuando el máximo sea mayor a cero. En caso de que el factor de certeza mínimo es menor que cero, se asignará cero de factor de certeza a la conclusión.

c) En caso de no haber conjunción o disyunción, el valor de verdad es igual al valor de verdad de la premisa siempre y cuando sea mayor a cero. En caso de ser menor o igual a cero, será cero.

d) En caso de haber más de una regla con la misma conclusión, se hace una combinación de factores de certeza de la siguiente forma:

En caso de que los factores de certeza sean positivos:

$$\text{Factor de certeza} = CF1 + CF2/100*(100-CF1)$$

En caso de que los factores de certeza sean negativos:

$$\text{Factor de certeza} = -(|CF1| + |CF2|/100*(100-|CF1|))$$

En caso de que un factor de certeza sea negativo y otro positivo:

$$\text{Factor de certeza} = (CF1 + CF2) * 100 / (100 - A),$$

$$A = \text{MIN} (|CF1|, |CF2|)$$

donde: CF1= Factor de certeza de la regla 1

CF2= Factor de certeza de la regla 2

## 5.6. OTRAS UTILIDADES DEL SISTEMA BASADO EN REGLAS.

El sistema basado en reglas diseñado en este proyecto para dar recomendaciones, tiene la potencialidad para aplicarse en problemas donde se involucra algún conocimiento. Esto es, puede adaptarse para ser una pequeña herramienta de sistemas expertos, ya que está basado en reglas y tiene un motor de inferencias para realizar razonamiento. Para adaptarlo, habría que rediseñar el lenguaje para que acepte hechos, en lugar de especificar si un reactivo es correcto o incorrecto. También habría que implantar el algoritmo de razonamiento hacia atrás, ya que algunos problemas lo requieren, y también habría que implantar un módulo de explicaciones del razonamiento.

## 5.7. ALGORITMO EVALUADOR.

Antes de explicar el algoritmo evaluador, es necesario tener presente la estructura de los archivos de evaluaciones y de reactivos que se encuentran en el Anexo A.

El archivo REACTIVOS tiene toda la información referente al reactivo: identificador, clave de materia al que pertenece, número de tema al que pertenece, autor del reactivo, tipo de reactivos, descripción, redacción de la pregunta, respuestas, etc. El archivo EVALUACION tiene información sobre la evaluación: periodo en que se aplica, clave de materia, número de grupo en el que se aplica la evaluación, identificador, número de preguntas, autor, descripción, tiempo límite para contestarla y si es navegable o no. El archivo EVAL-REAC es el detalle de los reactivos que se encuentran en la evaluación. Este contiene: identificador de la evaluación, tipo de reactivo, identificador del reactivo, clave de materia (se usa sólo en caso de ser evaluación tipo aleatoria), número de tema (se usa sólo en caso de ser evaluación tipo aleatoria), ponderación del reactivo en la evaluación y banderas para poner recomendación por reactivo en caso de fallo o acierto.

El algoritmo evaluador utilizado es el siguiente:

1. Pedir la matrícula del alumno
2. Pedir el periodo (semestre) actual y la clave de la materia-grupo
3. Desplegar las diferentes evaluaciones que existen para esa materia-grupo
4. Leer la evaluación deseada
5. Si la evaluación no existe en el archivo EVALUAC entonces
  - 6.1. Avisar que no existe esa evaluación, ir a 4
6. Si existe la evaluación en EVALUAC entonces
  - 6.1. nreactivo = 1
  - 6.2. Ciclo
    - 6.2.1. Leer el reactivo nreactivo del archivo REAC-EVAL de la evaluación seleccionada.

- 6.1.2. Preguntar Reactivo.
- 6.1.3. Guardar en memoria el reactivo de la evaluación
- 6.1.4. Si contestó bien el reactivo
  - 6.1.4.1. Si se debe dar recomendación de acierto, desplegarla
  - 6.1.4.2. Actualizar en memoria el acierto en ese reactivo con su ponderación respectiva.
  - 6.1.4.3.  $nreactivo = nreactivo + 1$
- 6.1.5. Si no contestó bien el reactivo
  - 6.1.5.1. Si se debe dar recomendación de falla, desplegarla
  - 6.1.5.2. Actualizar en memoria la falla en ese reactivo.
  - 6.1.5.3.  $nreactivo = nreactivo + 1$
- 6.1.6. Si la evaluación es navegable y no se contestó el reactivo
  - 6.1.6.1. Actualizar en Memoria que no se contestó el reactivo
- 6.1.7. Si la evaluación es navegable y se eligió ver el siguiente reactivo
  - 6.1.7.1. Si  $nreactivo < totalreactivos$ 
    - 6.1.7.1.1.  $nreactivo = nreactivo + 1$
    - 6.1.7.1.2. Ir a 6.2.
  - 6.1.7.2. Si  $nreactivo = totalreactivos$ 
    - 6.1.7.2.1. Salir del ciclo
- 6.1.8. Si la evaluación es navegable y se eligió ver el reactivo anterior y  $nreactivo > 1$ 
  - 6.1.8.1.  $nreactivo = nreactivo - 1$
  - 6.1.8.2. Ir a 6.2

Fin de Ciclo (Hasta que seleccione terminar de contestar evaluación o conteste la última pregunta)

6.2. Inicializar la calificación, número de aciertos y número de fallos a cero

6.3. Ciclo de  $i=1$  a Totpreg realizar

- 6.3.1. Leer de Memoria el registro generado en 6.1.3, en donde está registrado si falló, acertó o no contestó el reactivo y de la evaluación.
- 6.3.2. Si es correcta la respuesta del reactivo  $i$ 
  - 6.3.2.1. Aumentar en 1 la variable de total de aciertos

6.3.2.2. Aumentar el % de ponderación del reactivo  $i$ , en la calificación final

6.3.3. Si no es correcta la respuesta del reactivo  $i$

6.3.3.1. Aumentar en 1 la variable de total de fallos

6.3.4. Registrar en el archivo de Detalle de Bitácora (DETBIT) lo que contestó en esta pregunta.

Fin de Ciclo

6.4. Mostrar la calificación obtenida.

6.5. Registrar en Bitácora (BITACORA) la calificación final, duración de la evaluación, matrícula del alumno y fecha.

6.7. Interpretar las reglas de conocimiento que se encuentran en el archivo REGLAS, aplicando el compilador y el algoritmo de razonamiento explicados en el subcapítulo 5.4 y 5.5.

6.8. Mostrar las recomendaciones en orden de importancia (o de certeza).

7. Ir a 1) hasta que se seleccione Salir.

## **5.8. PROGRAMAS IMPLANTADOS DEL INTÉRPRETE (EVALUATEC).**

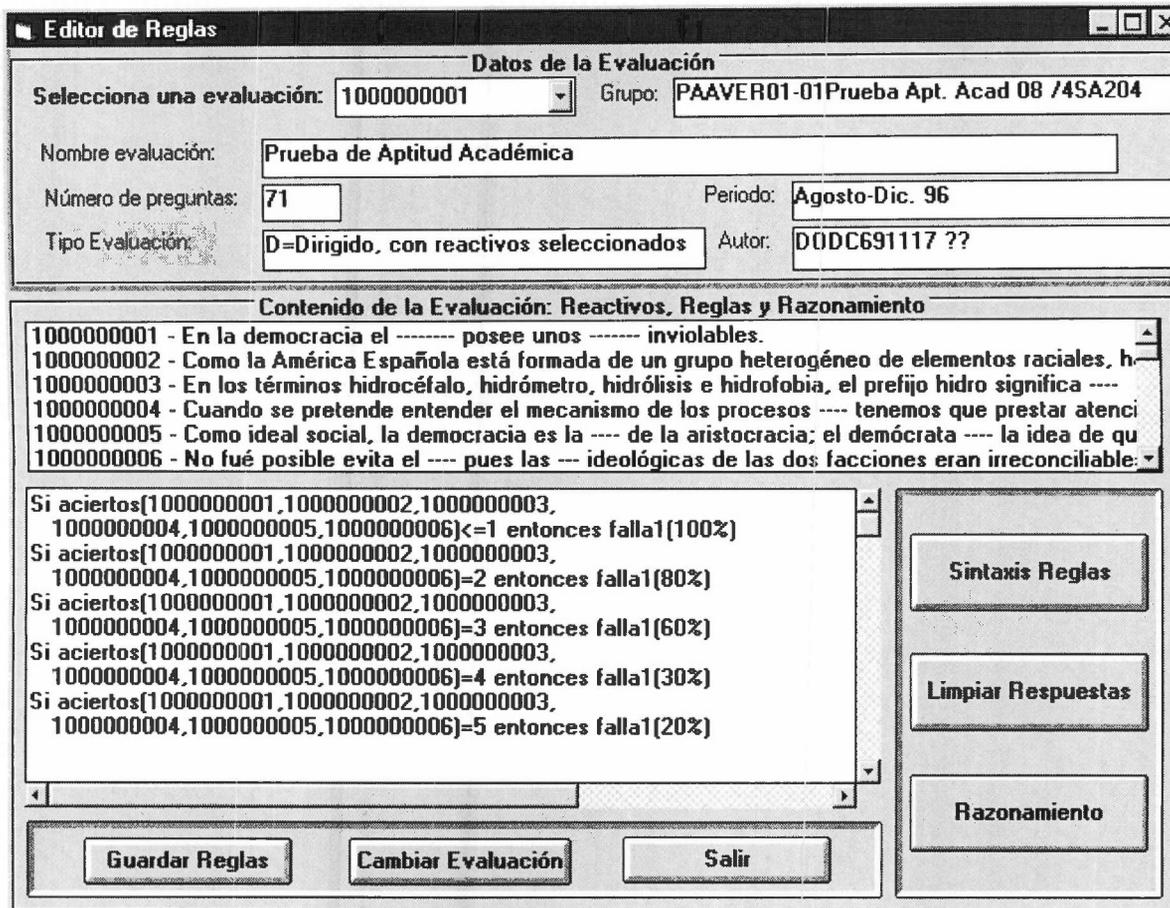
El intérprete se desarrolló en el lenguaje Visual Basic 3.0 para Windows. El *software* desarrollado se le nombró EVALUATEC. Se implantaron dos programas:

- 1) Programa para crear las reglas por evaluación
- 2) Programa evaluador

El código de cada programa se encuentra en el Anexo C.

El primer programa es un editor para escribir las reglas de las evaluaciones. En este programa se muestra toda la información de la evaluación: sus reactivos, materia y grupo al que pertenece, autor, reglas, etc. También se puede hacer una simulación para hacer una corrida para verificar qué recomendaciones se dan.

La pantalla principal de este programa es la siguiente:



En esta pantalla, al elegir la evaluación 1000000001 enseguida muestra toda la información referente. Después permite editar las reglas que se desean aplicar en la evaluación. Tiene la posibilidad de compilar el conjunto de reglas, y de realizar el razonamiento para probar dichas reglas. También es posible ver el contenido de cada reactivo de la evaluación al seleccionarlo con doble "click" con el ratón. Esto es de utilidad, ya que al estar editando las reglas, se debe de saber qué reactivos están involucrados. Al seleccionar un reactivo aparece la siguiente pantalla:

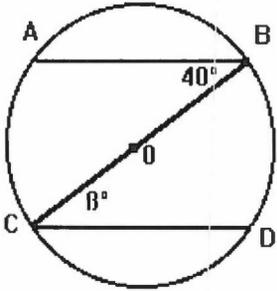
Contenido del Reactivo

Idreactivo: 1000000042    Materia: PAAVER01 Prueba Apt. Acad

Descripción Reactivo:    Tema: Tema no definido

Tipo Reactivo: 2=Opciones múltiples    Autor: Autor no definido

En el círculo cuyo centro es  $O$ , las cuerdas  $AB$  y  $CD$  son paralelas. Si el ángulo  $ABC = 40^\circ$  entonces  $B = ?$



A) 40  
 B) 10  
 C) 20  
 D) 60  
 E) 180

Respuesta: A)

Ayuda    Tutorial    Referencias    Recomendaciones    Dificultad    Salir

En esta pantalla se presenta toda la información del reactivo. En este ejemplo se mostró un reactivo de opción múltiple. Los botones de la parte inferior muestran la siguiente información al seleccionarlos:

- Ayuda: si se definió una ayuda, esta será mostrada como un texto.
- Tutorial: si se definió algún tutorial, éste se ejecutará.
- Referencias: si se definieron referencias, se mostrarán como un texto.
- Recomendaciones: si se definieron recomendación en caso de fallo y en caso de acierto, éstas se desplegarán como un texto.
- Dificultad: Parámetro que se va calculando con la utilización del reactivo. En este proyecto no se tiene contemplado actualizar este parámetro. Sólo fue considerado para incluir en un futuro el tipo de evaluación adaptativa.

El segundo programa es el evaluador, es el que ejecuta la evaluación para que sea contestada por el alumno y al final da las recomendaciones.

La pantalla de inicio es la siguiente:

Pantalla Principal EVALUATEC

### Sistema de Evaluación Automática (EVALUATEC)

Matrícula:	390726
Nombre:	DORANTES D. ALBERTO
Semestre:	9635 Semestre Ago-Dic 96
Cve Materia-Gpo:	PAAVER0101
IdEvaluación:	1000000001

Ir a Evaluación

Salir

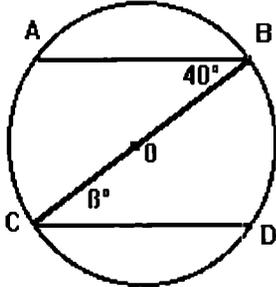
En esta pantalla el usuario dio de entrada su matrícula, el semestre actual, la clave de la materia-grupo y la clave o identificador de la evaluación a contestar. Posteriormente debe de seleccionar el botón "Ir a Evaluación". Al seleccionarlo le aparecerá el primer reactivo de la evaluación seleccionada para ser contestado. En el caso de este ejemplo, aparecería en pantalla lo siguiente:

**Sistema de Evaluación Automática**

Evaluación:  Datos Evaluación Profesor:

Materia:  Total Preguntas:  Límite Tiempo:

42.- En el círculo cuyo centro es O, las cuerdas AB y CD son paralelas. Si el ángulo ABC = 40° entonces B = ?



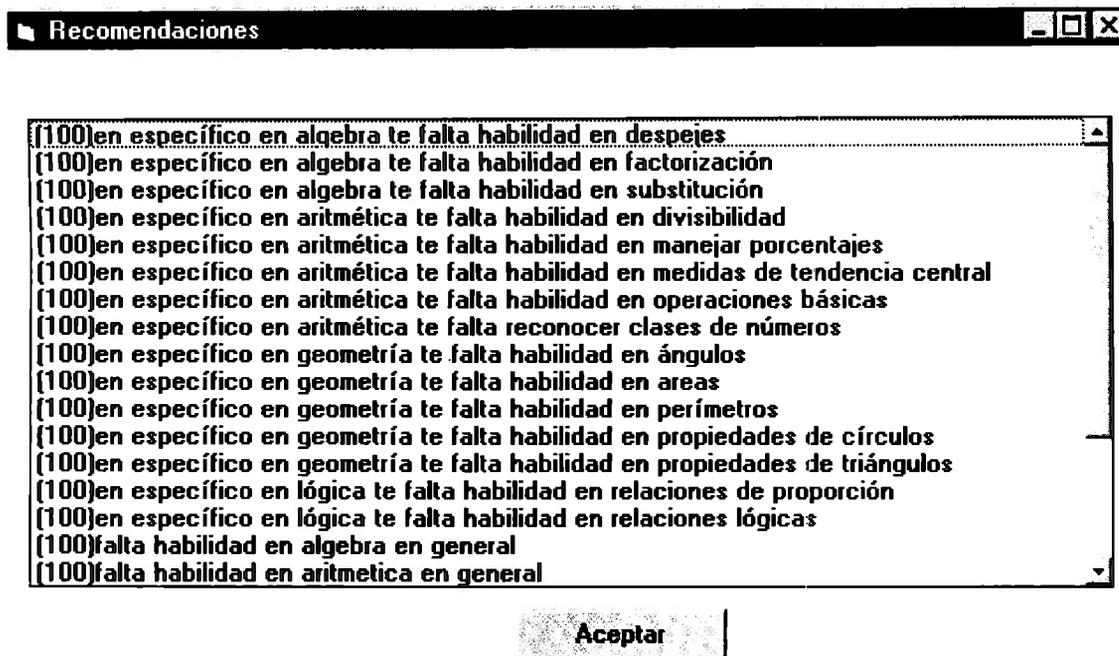
A) 10  
B) 20  
C) 60  
D) 180  
E) 40

Elige Opción:

Este tipo de pantalla es en donde el examinado va a contestar los reactivos. En este caso se presentó un reactivo de opción múltiple. A parte de contestar el reactivo, existen botones para acceder información de ayuda, tutorial y referencias los cuales son definidos por el profesor al momento de diseñar el reactivo. El tiempo se va marcando en la parte superior derecha, y en caso de tener tiempo límite también se especifica. También aparecen dos botones con flechas para navegar en la evaluación, sin necesidad de ir contestando todos los reactivos. Esto es porque en esta evaluación se puede navegar en los reactivos. Y por último un botón de terminar para finalizar la evaluación.

Al terminar la evaluación, aparece la calificación obtenida y las recomendaciones finales en orden de importancia. Internamente registra en dos archivos de bitácora (ver Anexo A) las respuestas que se dieron en cada reactivo, la calificación final, la matrícula del alumno y el tiempo que tomó para la evaluación. Estos archivos de

bitácora que se generan, son los que se utilizan para el módulo de estadísticas y consultas explicado en el capítulo 3. La forma en que aparecen las recomendaciones en pantalla es:



Al principio de cada recomendación aparece la probabilidad en porcentaje de que la recomendación sea la más relevante. Estos valores de probabilidad van de 20 a 100%.

## **CAPÍTULO 6. RESULTADOS Y CONCLUSIONES.**

La idea principal del presente trabajo surgió por el interés de aplicar técnicas de inteligencia artificial al proceso de enseñanza-aprendizaje. El resultado de este trabajo fue un *software* llamado EVALUATEC para aplicar evaluaciones por computadora con la capacidad de dar retroalimentación al examinado por medio de recomendaciones de acuerdo a su desempeño.

### **6.1. TESIS DE ESTE TRABAJO.**

La tesis de este trabajo se resume a dos afirmaciones:

- a) Las evaluaciones de un proceso de enseñanza-aprendizaje se pueden llevar a cabo por computadora en forma automática, aplicando herramientas tecnológicas descritas en este trabajo.
- b) En estas evaluaciones se puede aplicar un sistema basado en reglas para generar retroalimentación automática.

## **6.2. APLICACIONES DEL SOFTWARE EVALUATEC.**

El presente *software* fue diseñado para aplicarse en los cursos impartidos en el I.T.E.S.M. Campus Querétaro. El objetivo a futuro es que profesores de algunos cursos empiecen a utilizar este *software* para aplicar exámenes rápidos o ejercicios vistos en clases por medio de la red del Campus. Con esto los alumnos tendrían acceso a las evaluaciones desde la sala de alumnos del Campus.

Al final del desarrollo del *software*, se identificaron otras aplicaciones útiles. Una de ellas es la de aplicar por computadora un examen similar al de admisión al I.T.E.S.M. con el fin de dar retroalimentación al aspirante. Otra de las aplicaciones útiles puede ser para aplicar encuestas por computadora, aunque en este caso no se utilizaría la parte de reglas de conocimientos para dar recomendaciones.

## **6.3. PRUEBAS REALIZADAS DEL SOFTWARE.**

Actualmente se está desarrollando un proyecto en el I.T.E.S.M. Campus Querétaro para lograr disminuir el porcentaje de rechazados a carreras profesionales y a preparatoria. El sistema I.T.E.S.M. aplica una Prueba de Aptitud Académica diseñada por el College Board para decidir la admisión de los aspirantes. Desde hace un año se empezó a ofrecer un curso opcional para los aspirantes a presentar dicha prueba. Realmente este curso no ha dado resultados significativos, ya que el porcentaje de rechazo sigue siendo igual que antes. Dentro del proyecto para disminuir el porcentaje de rechazados, se tiene planeado hacer un análisis del contenido de este examen. Hasta la fecha sólo se han identificado áreas de conocimiento y habilidades que abarca la prueba.

Aprovechando esta situación, se decidió probar el *software* con una evaluación similar a la Prueba de Aptitud Académica. La evaluación que se probó fue la que viene al final del folleto de Orientación para tomar la Prueba de Aptitud académica que se

entrega a los aspirantes a presentar el examen de admisión. Esta prueba consta de cuatro partes y un total de 96 preguntas. De estas preguntas se registraron solo 71, ya que no se registraron las preguntas relacionadas con comprensión de lecturas. Cada parte es de 25 minutos. Debido a que el *software* no tiene la opción de dividir una evaluación por secciones, se definió una sola sección de 90 minutos, ya que no se incluyeron las preguntas de comprensión de lecturas. Los reactivos fueron dados de alta tal cual como están en la prueba, con todo y gráficos asociados. Los reactivos y la evaluación se registraron directamente en la base de datos, por medio de un programa llamado "Data Manager" que contiene el paquete computacional "Visual Basic 3.0". El registro de los reactivos debe hacerse en el módulo de diseño de reactivos, y el diseño de la evaluación debe hacerse en el módulo de diseño de evaluaciones, explicados en el capítulo tres.

Después de registrar los reactivos de la prueba, se procedió a generar las reglas de conocimiento para dar las recomendaciones. Para saber qué recomendaciones se darán es necesario saber qué evalúa la prueba y a qué se refiere cada reactivo. Para esto se consultó un documento del proyecto mencionado anteriormente [12] para disminuir el porcentaje de rechazados, y también se estudió cada reactivo para identificar a qué área pertenece cada uno. Las reglas que se definieron para esta prueba se encuentran en el anexo B.

Al momento de hacer estas recomendaciones, se identificaron algunas deficiencias en el lenguaje definido originalmente, y se hicieron algunas adecuaciones. Una de éstas fue la de aumentar la condición para identificar cuántos aciertos tuvo el examinado de un conjunto seleccionado de reactivos.

Es importante resaltar que para definir las reglas más adecuadamente, es necesario que un experto en esta prueba las diseñe. Gran parte del éxito de este *software* depende de la definición correcta de las reglas.

Realmente no se han hecho pruebas de campo para demostrar la eficiencia del *software* EVALUATEC. Pero lo que se hizo fueron pruebas con pequeñas evaluaciones, y con la evaluación de una Prueba de Aptitud Académica para verificar su correcto funcionamiento. Se tiene pensado utilizar este *software* como apoyo al

curso que actualmente se está dando para preparar a los aspirantes para la Prueba de Aptitud Académica.

Otra aplicación importante de este *software* es la parte del razonamiento de las reglas. Se puede hacer una adaptación al *software* para generar una herramienta sencilla de sistemas expertos y así aplicarse para resolver problemas donde se involucra algún tipo de conocimiento. Para generar esta herramienta habría que implantar el razonamiento hacia atrás ("Backward Chaining"), el submódulo de explicaciones descrito en el capítulo cuatro, y adaptar el lenguaje para que sea más general.

#### **6.4. MEJORAS POSIBLES IDENTIFICADAS.**

Durante el desarrollo del proyecto se identificaron mejoras, tanto en el diseño propuesto de un sistema integral de evaluación, como en la implantación del *software*. Algunas de estas mejoras fueron incorporadas, pero otras no. Las mejoras que no se incorporaron son:

En el diseño del sistema:

- a) Aumentar Planes de Estudio para poder diseñar evaluaciones integradas por carreras.
- b) Diseñar un sistema más general en el que se incluya un sistema de tutoriales asociados por temas y por materia, y que se encuentre ligado a las recomendaciones que se generan a los alumnos.
- c) Integrar una metodología de tecnología educativa a todo el proyecto. Esto se haría con expertos en el área de educación.

En la implantación:

- a) Diseñar una interface con un editor más apropiado, para la redacción de reactivos con cualquier tipo de gráficos, signos y fórmulas.

- b) Modificar el lenguaje de la definición de las reglas para incluir información académica como: promedios de alumnos, resultados de examen de admisión, calificaciones parciales en las materias que cursan los alumnos, asistencias a clases, etc. En el sistema propuesto se contempla toda esta información, pero no está incorporada en el lenguaje de definición de las reglas. Esto haría que las recomendaciones fueran más certeras.
- c) Diseñar una forma de generar evaluaciones automáticamente por temas, en el cual se le de como entrada el o los temas, el número de preguntas y un conjunto de reglas de conocimiento que involucre los temas. Para esto habría que implantar el algoritmo de razonamiento hacia atrás (Backward Chaining), para ir haciendo las preguntas dependiendo de los temas a evaluar. Esta forma de evaluaciones sería una especie de evaluación adaptada al tema seleccionado.
- d) Poder dividir una evaluación en varias partes y cada parte con límite de tiempo y reglas por separado.
- e) Soportar evaluaciones del tipo adaptativas, de dominio referenciado personalizado y de simulación lógicamente armada (tipos de evaluación descritos en el capítulo 2) .
- f) implantación del módulo de estadísticas. Este módulo llevaría el control histórico de las evaluaciones contestadas por alumno y por grupo, para dar retroalimentación tanto al alumno como para el profesor.

## **6.5. CONCLUSIÓN FINAL.**

En resumen se puede concluir que el presente trabajo es un ejemplo más de la aplicabilidad de la inteligencia artificial en situaciones reales. El presente proyecto aporta tres aspectos valiosos:

- a) Propuesta de un sistema integral de evaluación por computadora.
- b) Propuesta de una aplicación de técnicas de Inteligencia Artificial en el proceso de enseñanza-aprendizaje: sistema basado en reglas para determinar recomendaciones en una evaluación.

c) Propuesta de un método de razonamiento hacia adelante en un sistema basado en reglas. Este método se puede adaptar como una herramienta práctica de sistemas expertos para resolver problemas en los que se involucre conocimiento.

## REFERENCIAS:

- [1] DORANTES, CARLOS A.; XOTLANIHUA, David; MADARIAGA, Alejandro. Manual de instrucciones de Sistema Generador de Tutoriales (GT). Departamento de Electrónica y Sistemas, I.T.E.S.M. Campus Querétaro, Querétaro, Qro., 1991.
- [2] TORRES, Mónica; VALENZUELA, Luis; MADARIAGA, Alejandro. Proyecto SIMAT. Memorias del primer Concurso Nacional de Proyectos de Ingeniería y Ciencias (CONAPIC '93), Tijuana, B.C., 1993.
- [3] SÁNCHEZ, Omar; MADARIAGA, Alejandro. Proyecto HAEF. Memorias del segundo Concurso Nacional de Proyectos de Ingeniería y Ciencias (CONAPIC '94), Tijuana, B.C., 1994.
- [4] MADARIAGA, Alejandro. Proyecto ADSE. Memorias del tercer Congreso Internacional de Investigación en Ciencias Computacionales, Tijuana, B.C., 1996.
- [5] "User's Manual for the MicroCAT Testing System". Third Edition. *Software* desarrollado por ASSESSMENT SYSTEM CORPORATION, St. Paul, MN, U.S.A., 1989.
- [6] TURBAN, Efraim. "Expert Systems and Applied Artificial Intelligence". Ed. Moura, New York, 1992.
- [7] RICH, Elaine, KNIGHT Kevin. "Artificial Intelligence". Second Edition. Ed. McGraw-Hill, 1991.
- [8] ROLSTON, David. "Principios de Inteligencia Artificial y Sistemas Expertos". Traducido por Alfonso Pérez Gama. Ed. McGraw-Hill, Naucalpan, Edo. de Méx., 1992.
- [9] LEVINE, Robert; DRANG, Diane E.; EDELSON Barry. "AI and Expert Systems". Second Edition. Ed. McGraw-Hill, 1990.
- [10] AHO, Alfred V.; RAVI, Sethi; ULLMAN, Jeffrey D. "Compilers: principles, techniques and tools". 1a, Edición. Ed. Addison-Wesley, 1986

[11] CANTÚ, Francisco J. “Evaluación y Selección de Herramientas de Desarrollo de Sistemas Expertos”. Programa de Seminarios en Sistemas Expertos, Herramientas de S.E., I.T.E.S.M. Campus Monterrey, Monterrey, N.L., 1989.

[12] DE LA TORRE, Roberto; LOPEZ, Marco V., BURGOS, Alvaro; NORIEGA, Julio; TORRES, Luis F.; DE LA TORRE, Irma. “Preparación para el examen de admisión de futuros aplicantes”. Modelo de crecimiento; área: Incremento en la eficiencia; subárea: admisión. I.T.E.S.M. Campus Querétaro, Querétaro, Qro., 1996.

## **BIBLIOGRAFÍA:**

AHO, Alfred V.; RAVI, Sethi; ULLMAN, Jeffrey D. "Compilers: principles, techniques and tools". 1a, Edición. Ed. Addison-Wesley, 1986

CANTÚ, Francisco J. "Evaluación y Selección de Herramientas de Desarrollo de Sistemas Expertos". Programa de Seminarios en Sistemas Expertos, Herramientas de S.E., I.T.E.S.M. Monterrey, N.L. 1989.

DE LA TORRE, Roberto. "The development and evaluation of a system for computerized adaptive testing". Thesis to certify Ph.D., University of Iowa, 1991.

KELLER, Robert. "Expert System Technology Development & Application". Ed. Prentice Hall, New Jersey, 1987.

LEVINE, Robert; DRANG, Diane E.; EDELSON Barry. "AI and Expert Systems". Second Edition. Ed. McGraw-Hill, 1990.

NELSON, Ross. "Guía Completa de Visual Basic para Windows". Segunda Edición, McGraw-Hill, Naucalpan, Edo de México, 1994.

RICH, Elaine, KNIGHT Kevin. "Artificial Intelligence". Second Edition. Ed. McGraw-Hill, 1991.

ROLSTON, David. "Principios de Inteligencia Artificial y Sistemas Expertos". Traducido por Alfonso Pérez Gama. Ed. McGraw-Hill, Naucalpan, Edo. de Méx., 1992.

THE COLLEGE BOARD. "Orientación para tomar la Prueba de Aptitud Académica". Hato Rey, Puerto Rico, 1983..

TURBAN, Efraim. "Expert Systems and Applied Artificial Intelligence". Ed. Moura, New York, 1992.

WEISKAMP, Keith; HENGL, Terry. "Artificial Intelligence Programming with Turbo Prolog". Ed. John Willey & Sons, Inc., New York, 1988.

## **CONSULTAS EN INTERNET (WEB Y GOPHER)**

"This is Educational Testing Service (ETS)"  
gopher://etsis4.ets.org:70/00/about.ets/tie.asc, Octubre 1994.

"Innovation and Technologies of ETS"  
gopher://etsis4.ets.org:70/00/about.ets/iat.asc, Octubre 1994.

"Computer Based Testing of ETS"  
gopher://etsis4.ets.org:70/00/cbt/cbt.asc, Octubre 1994.

"Computerized Adaptive Testing (CAT)"  
gopher://etsis4.ets.org:70/00/cbt/wicat.asc, Octubre 1994.

"Benefits of Computer Based Testing (CBT)"  
gopher://etsis4.ets.org:70/00/cbt/bocbt.asc

"Computerized Mastery Testing (CMT)"  
gopher://etsis4.ets.org:70/00/cbt/wicmt.asc, Octubre 1994.

"Our Commitment (ETS)"  
gopher://etsis4.ets.org:70/00/cbt/ectcbt.asc, Octubre 1994.

"CBT Test Center Network"  
gopher://etsis4.ets.org:70/00/cbt/ctcn.asc, Octubre 1994.

"What's new on the ETS Gopher"  
gopher://etsis4.ets.org:70/00/news.asc.

Integrated Computer Products, Your source for Vocational, Academic & Home Software."Teachers Tools".  
<http://www.a-netservice.com/azbus/icp/teacher.html>, 18 de Febrero de 1996.

The Praxis Series: Professional Assessments for Beginning Teachers. ETS\* Net.  
"Computer-Based Testing (CBT) for Praxis I: Academic Skills Assessments".

<http://www.ets.org/praxis/prxcbt.html>

“What is ETS”

<http://www.ets.org/visitors.html>

“Test Designed Supreme”

<http://www.consultantalliance.com/Superschoolsoftware/test.html>

“The J.B. Barber Technology in Education Grant”. Authentic assessment for a better evaluation of student performance.

<http://www.consultantalliance.com/Superschoolsoftware/grant.html>

“La tecnología en la enseñanza” (Revista)

<http://www.mpsnet.com.mx/quipus/REVIS.HTML>

“La tecnología en la enseñanza” (Editorial)

<http://www.mpsnet.com.mx/quipus/r5edit.html>

“Diferentes usos de las computadoras en las Escuelas”

<http://www.mpsnet.com.mx/quipus/difeuso.html>

“Tilt Group E-Evaluation”

<http://www.elec.gla.ac.uk/TILT/E-Eval.html>, 20 de marzo de 1996

“Computer assisted learning (CAL) within the Faculty of Medicine”

<http://www.med.ubc.ca/calubc.html>

“Evaluation of Learning Technology in Higher Education (elthe)”

<http://annick.stir.ac.uk/elthe/reflib/>

“Welcomo to TopGrade Software”

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

“Northwest Council for Computer Education”

<http://www.jazzie.com/ncce/w96.html>

## ANEXO A. DISEÑO DE LA BASE DE DATOS

Se utilizó la siguiente notación para describir a cada campo de cada archivo:

An = Alfanumérico de n caracteres

Nn = Numérico de n caracteres

A& = Alfanumérico de tamaño variable (sin límite de caracteres)

DDMMAAAA = Formato de fecha: día (2 dígitos), mes (2 dígitos), año (4 dígitos)

Real = Número real

Boolean = de solo dos valores: falso o verdadero.

La estructura de cada archivo y la descripción de cada campo de la base de datos es la siguiente:

### A1. Archivo de Alumnos

Alumnos:		Nombre del archivo: SESALUM
Periodo	: A3	{ Semestre o Periodo en el que está inscrito. Notación: AAS, donde: AA: Año S: Semestre: Para Prog. Semestral: 1=Ene-May, 2=Verano, 3=Ago-Dic Para Prog. Trimestral: 1= Ene-Abr, 2=Abr-Jun, 3=Jun-Ago, 4= Sep-Dic }
TipPer	: A1	{ Tipo de Periodo: S= Semestral, T=Trimestral }
Matricula	: A8	{ número de matrícula del alumno }
Carrera	: A3	{ siglas de la carrera inscrita. Debe existir en el archivo de carreras }
Plan	: A2	{ año del plan de estudios de la carrera inscrita }
Nombre	: A30	{ nombre abreviado del alumno: Ap. Paterno, Inicial Materno, Nombre(s) }
Sexo	: A1	{ sexo del alumno. M=Masculino, F=Femenino }
Cvinal	: A2	{ Clave de ingreso del alumno en ese periodo o semestre. Valores válidos: PI = Primer Ingreso al Sistema RN= Reingreso RC=Reingreso con Cambio de Ciclo (de Prepa del Campus a Profesional del Campus) TN=Transferencia de otro campus de Prepa a prepa, o de Prof. a prof. TC= Transferencia de Prepa de otro Campus a Profesional del Campus. }
Semestre	: N1	{ Semestre en que se encuentra el alumno }
PromSec	: N3	{ Promedio que obtuvo en Secundaria. El formato es de 00 a 100 }
PromPre	: N3	{ Promedio que obtuvo en Preparatoria, si está en Profesional }
PromPro	: N3	{ Promedio que obtuvo en Profesional, si está en Maestría }
PromGral	: N3	{ Promedio general que lleva sin contar el semestre actual }
PromUlse	: N3	{ Promedio que obtuvo en el semestre anterior }
TotRepro	: N2	{ Número de materias reprobadas sin contar el semestre actual }
PAAV	: N3	{ Resultado en la parte verbal de la Prueba de Aptitud Académica. El rango es de 0 a 800 }
PAAM	: N3	{ Resultado en la parte matemática de la Prueba de Aptitud Académica. El rango es de 0 a 800 }
PAG	: N3	{ Resultado de la Prueba de Aptitud de Graduados (sólo para alumnos de Maestría ) }
Llaves archivo SESALUM:		
Llave0	: Periodo, TipPer, Matricula	(única, primaria)
Llave1	: Matricula	(No única)

## A2. Archivo de Carreras.

Carreras: Nombre del archivo: SESCARRE  
Clave : A3 { Clave de la carrera  
FORMATO: XXX, alfabético }  
Nivel : A1; { Nivel al que pertenece la carrera:  
FORMATO Y VALORES VALIDOS:  
'0': Preparatoria  
'1': Profesional  
'2': Maestría }  
Nombre : A60 { Nombre completo de la carrera }  
TipoPer : A1  
{ Tipo de Periodos:  
FORMATO Y VALORES VALIDOS:  
'S'= Semestral  
'T'= Trimestral }  
NumPer : N2 { Número de Periodos que se ofrecen en el  
Campus.  
FORMATO Y VALORES VALIDOS:  
Numérico, del 0 al 15 }  
Completa : A1 { Si está completa en el Campus  
FORMATO Y VALORES VALIDOS:  
'S' = Completa  
'N' = No Completa, a terminar en otro Campus }  
Status : A1 { Status de la Carrera  
FORMATO Y VALORES VALIDOS:  
'A' = Carrera abierta  
'C' = Cerrada para los Nuevos Ingresos,  
solo abierta para los de Reingreso  
'K' = Carrera cerrada completamente }  
TipoImp : A1 { Tipo de Impartición:  
FORMATO Y VALORES VALIDOS:  
'P' = Presencial  
'S' = Satelital }  
Director : A40 { Nombre del Director de la carrera }

Llaves archivo SESCARRE:

Llave0 : Clave { única, primaria }

## A3. Archivo de Grupos.

Grupos: Nombre del archivo: SESGRUPO  
Periodo : A3 { Semestre en que se está dando o se dio el grupo  
FORMATO Y VALORES VALIDOS:  
AAP, donde:  
AA = Año (del 70 al 99)  
P = Periodo Semestral o Trimestral  
Si TipPer='S':  
1 = Sem. Enero-May  
2 = Verano Semestral  
3 = Sem. Agosto-Dic  
Si TipPer='T':

1 = Trim. Ene-Abr.  
 2 = Trim. Abr-Jun.  
 3 = Verano Trim.  
 4 = Trim. Sep-Dic.

**TipPer** : A1 { Tipo de Periodo como cursó la materia  
 FORMATO Y VALORES VALIDOS:  
 S = Periodo Semestral  
 T = Periodo Trimestral }

**CveMat** : A8 { Clave de la materia, incluye el plan  
 FORMATO: DD PPM MM, donde:  
 DD = Clave del Depto de la materia  
 PP = Plan de la materia  
 MMM= Número de la materia  
 VALORES VALIDOS:  
 DD debe ser alfabetico  
 PP debe ser numérico  
 MMM debe ser numérico  
 VALIDACION EN B.D.:  
 CveMat DEBE existir en SESMATER.TST }

**Grupo** : A2 { Número de Grupo  
 FORMATO Y VALORES VALIDOS:  
 99 = Numérico, del 01 al 99 }

**RFCProf** : A10 { RFC de profesor que imparte  
 FORMATO Y VALORES VALIDOS:  
 Formato de un RFC oficial }

**Imparte** : A3 { Depto. que imparte el grupo  
 FORMATO: XXX }

**NomProf** : A30 { Nombre del profesor }

**Alumnos** : N3 { Número de alumnos inscritos en el grupo  
 FORMATOS VALIDOS:  
 999 Numérico, de 0 a 999 }

**TipoProf** : A1 { Tipo de contrato del profesor:  
 FORMATO: Numérico  
 VALORES VALIDOS:  
 '1'= Planta  
 '2'= Auxiliar-Tiempo Parcial  
 '3'= Auxiliar-Tiempo Completo  
 '4'= Auxiliar-Planta }

**Graacadem** : Char; { Grado académico máximo del profesor al dar el grupo  
 FORMATO Y VALORES VALIDOS:  
 'S'= Sin Licenciatura  
 'L'= Con Licenciatura  
 'M'= Con Maestría  
 'D'= Con Doctorado }

**Hora1** : A5 { Hora de inicio de la clase con duración en medias horas:  
 FORMATO:  
 HH+/D, donde:  
 HH: Hora de inicio de la clase  
 +: Si hay un +, quiere decir 30 minutos más, sino, debe haber espacio  
 /: Línea de división obligatoria  
 D: Número de medias horas que dura la clase }

**Dia1** : A5 { Día(s) en que se imparte la clase  
 FORMATO con valores válidos:  
 L = Lunes, Miércoles y Viernes  
 M = Martes y Jueves  
 LU = Lunes  
 MA = Martes

MI = Miércoles  
 JU = Jueves  
 VI = Viernes  
 Combinaciones de días, ejemplo: LU MA, MA VI }  
 Aula1 : A3 { Aula donde se imparte la clase  
 FORMATO: NNN }  
 Hora2 : A5 { Segundo horario de la clase  
 Formato Igual que Hora1 }  
 Dia2 : A5 { Segundo día del segundo horario  
 Formato igual que Dia1 }  
 Aula2 : A3 { Salón del segundo horario  
 Formato igual que Aula1 }

Llaves archivo SESGRUPO.

Llave0 : Periodo, TipPer, Cvemat, Grupo ( única, primaria)  
 Llave1 : Periodo, TipPer, RFCprof, Cvemat, Grupo ( única)

#### A4. Archivo de Materias

Materias: Nombre del archivo: SESMATER  
 Cvemat : A8 { depto de la clave de la materia  
 FORMATO: DD PP, donde:  
 DD = Clave del Depto de la materia  
 PP = Plan de la materia  
 VALORES VALIDOS:  
 DD debe ser alfabetico  
 PP debe ser num,rico }  
 RmNombre : A20 { Nombre de la materia que cursó  
 FORMATO: Cualquier string }  
 ObjetivoGen: A255 { Objetivo General de la materia  
 FORMATO: Cualquier string }  
 ObjetivoEsp: A255 { Objetivo específico de la materia  
 FORMATO: Cualquier string }  
 RmHClase : A2 { Horas clase a la semana  
 FORMATO : 99, de 2 caracteres  
 VALORES VALIDOS:  
 DEL 00 AL 40 }  
 RmHLab : A2 { Horas laboratorio a la semana  
 FORMATO : 99, de 2 caracteres  
 VALORES VALIDOS:  
 DEL 00 AL 40 }  
 RmUnidades: A2 { Unidades de la materia  
 FORMATO : 99, de 2 caracteres  
 VALORES VALIDOS:  
 DEL 00 AL 24 }  
 Nivel : A1 { Nivel en donde se imparte la materia  
 FORMATO: 9  
 VALORES VALIDOS:  
 0 = Si la materia es de Preparatoria  
 1 = Si la materia es de Profesional  
 2 = Si la materia es de Maestría }  
 Req1 : A9 { Requisito 1  
 FORMATOS Y VALORES VALIDOS:  
 1) DD PPM&, en donde:  
 DD PPM es la clave de una materia  
 & puede ser:

- +: Indica que el requisito es haber aprobado esta materia y a parte la siguiente.
- /: Indica que el req. es haber aprobado esta materia o la siguiente
- \$: Indica que el requisito es estar cursando o haber aprobado esta materia
- ' ': Indica que solo tiene el requisito es haber aprobado esa materia
- C : Indica que el req. es haber cursado, sin necesidad de aprobada.
- 2) NT : Indica que no tiene requisitos
- 3) ' ': Indica que no tiene requisitos
- 4) SE 00N\$: Indica que debe estar o haber aprobado todo el semestre N

VALIDACION EN B.D.

DD PPM MM Debe existir en MATERIAS.TST  
}

Req2 : A9 { Requisito 2  
Mismo formato y valores válidos que Req1 }

Req3 : A9 { Requisito 2  
Mismo formato y valores válidos que Req1 }

Req4 : A9 { Requisito 2  
Mismo formato y valores válidos que Req1 }

Req5 : A9 { Requisito 2  
Mismo formato y valores válidos que Req1 }

AreaAsig : A2 { Clave de área de Conocimiento donde se asigna la materia  
FORMATO: 99  
VALORES VALIDOS:  
Del 00 al 99

NOTA: No se usa por ahora este campo. Cuando se use, debe crearse una tabla de áreas de conocimiento que contenga la descripción de las claves.

Este campo y los dos siguientes fueron diseñado para que se puedan clasificar las materias por áreas de conocimiento. Ver posibles mejoras.)

AreaAlt1 : A2 { Area Alternativa 1 (Roster)  
FORMATO: 99  
VALORES VALIDOS:  
Del 00 al 99  
No se usa por ahora este campo }

AreaAlt2 : A2 { Area Alternativa 2 (Roster)  
FORMATO: 99  
VALORES VALIDOS:  
Del 00 al 99  
No se usa por ahora este campo }

Carga : Real { Carga académica para el profesor }

Imparte : A3 { Departamento académico que imparte la materia }

ObjetivoGen: A& { Objetivo General del curso }

ObjetivoEsp: A& { Objetivo específico del curso }

Llaves archivo SESMATER:

Llave0 : Cvemat (única)

## A5. Archivo de Alumno-Grupo

Alumno-Grupo: Nombre del archivo: SESALGPO

**Periodo** : A3 { Semestre en que curso la materia  
**FORMATO Y VALORES VALIDOS:**  
 AAP, donde:  
 AA = Año (94,95,etc)  
 P = Periodo Semestral o Trimestral  
 1 = Sem. Enero-May o Trim. Ene-Abr.  
 2 = Verano Semestral o Trim. Abr-Jun.  
 3 = Sem. Agosto-Dic o Verano Trim.  
 4 = Trim. Sep-Dic. }

**TipPer** : A1 { Tipo de Periodo que cursa o cursó la materia  
**FORMATO Y VALORES VALIDOS:**  
 S = Periodo Semestral  
 T = Periodo Trimestral }

**MatriAIG** : A6 { Matrícula del alumno  
**FORMATO NUMERICO:** 999999 de longitud=6  
**VALIDACION EN B.D.:**  
 La matricula debe existir el SESALUM.TST }

**CveMat** : A8 { Clave Depto de la materia cursando, incluye el plan  
**FORMATO:** DD PPM MM, donde:  
 DD = Clave del Depto de la materia  
 PP = Plan de la materia  
 MMM = Clave de la materia  
**VALORES VALIDOS:**  
 DD debe ser alfabetico  
 PP debe ser numérico  
 MMM debe ser numérico }

**GruAIG** : A2 { Numero de Grupo inscrito  
**FORMATO Y VALORES VALIDOS:**  
 99 = Numérico, del 01 al 99  
**VALIDACION EN B.D.:**  
 Periodo+TipPer+Depalg+Matalg+GruAlg DEBE existir  
 en SESGRUPO.TST}

**C1** : A2; { Calificación primer parcial  
**FORMATO Y VALORES VALIDOS:**  
 '99': del '01' al '10' Calificación num,rica  
 'NP': No presentó el examen  
 'SD': Sin derecho por faltas  
 'SC': Sin calificación  
 'A' : Acreditada por Examen  
 'CP': Calificación Pendiente }

**C2** : A2 { Calificación segundo parcial  
 Igual que C1 }

**C3** : A2 { Calificación tercer parcial  
 Igual que C1 }

**C4** : A2 { Calificación final  
 Igual que C1 }

**F1** : A2 { Faltas primer parcial  
**FORMATO Y VALORES VALIDOS:**  
 99, Num,rico de 00 a 99 }

**F2** : A2 { Faltas segundo parcial  
 Igual que F1 }

**F3** : A2 { Faltas tercer parcial  
 Igual que F1 }

**F4** : A2 { Faltas totales  
 Igual que F1 }

Llaves archivo SESALGPO:

Llave0 : Periodo, TipPer, Matrialg, Cvemat (única, primaria)  
Llave1 : Periodo, TipPer, Cvemat, Gualg, Matrialg (única)

## A6. Archivo de Autores

Usuarios o Profesores: Nombre del archivo: AUTOR  
RFCautor : A10 { RFC del Usuario o profesor }  
Nombre : A30 { Nombre abreviado del usuario }  
Password : A8 { Password del usuario o profesor, que usará para entrar al diseño de reactivos }

Llaves archivo AUTOR:  
Llave0 : RFCautor (única, Primaria)

## A7. Archivo de Materia-Temas

Materia-Tema: Nombre del archivo: PANALITICO  
CveMat : A8 { Clave de la materia al que pertenece el tema o subtema. Debe existir en el archivo MATERIAS }  
NTema : A20 { Número de tema o subtema. Cada dos caracteres será el número de tema o subtema. Por lo tanto se pueden tener hasta una profundidad de 10 niveles de subtemas. Ejemplo: Para el subtema 1.1.2.1: 01010201, para el tema 1.: 01  
NomTema : A50 { Nombre o título del Tema o Subtema. }  
DescTema : A255 { Descripción del Tema o Subtema }

Llaves archivo PANALITICO:  
Llave0 : Cvemat, Ntema (única, primaria)

## A8. Archivo de Reactivos.

Reactivos: Nombre del archivo: REACTIVOS  
CveMat : A8 { Clave de la materia a la que pertenece el reactivo }  
NTema : A20 { Número de tema o subtema. Mismo formato que en TEMAS }  
IdReactivo : A10 { Identificador del reactivo. Número consecutivo del reactivo. Identificador único del reactivo }  
RFCautor : A8 { Clave del autor o profesor que diseña el reactivo }  
Descripcion: A50 { Descripción del reactivo }  
Redacción : A& { Redacción del reactivo o pregunta }  
Tipo\_respuesta: N1 { Tipo de respuesta:  
Valores válidos: 1 = opciones múltiples generadas para un ejercicio dado  
2 = opciones múltiples  
3 = Valor específico  
4 = Falso/Verdadero }  
Solucion : A& { Forma de solucionar el reactivo, en caso que sea un ejercicio de tipo\_respuesta=1. En caso de ser de tipo\_respuesta=3, en este campo se registra el valor de la respuesta }  
Ayuda : A255 { Descripción de cómo solucionar el reactivo o alguna otra ayuda para solucionarlo }  
Tutor : A50 { Nombre de un programa ejecutable tutorial para poder accederlo al estudiar el reactivo (no al contestarlo) }

Valores : A255 { Sólo se usa en caso de ser de tipo\_respuesta=1. Variables sustituibles con un rango de valores, para que al generar una evaluación cada examen sea diferente, con diferentes parámetros. }

Grafico : A50 { Nombre de una imagen o dibujo que se desplegará al mostrar el reactivo }

Disciminacion: A10 { Variable utilizada para clasificar al reactivo conforme a su comportamiento. Esta variable se adicionó para posteriormente generar evaluaciones adaptativas. Para este proyecto no se usará. }

Dificultad : A10 { Variable utilizada junto con Discriminación para clasificar por nivel de dificultad al reactivo conforme a las respuestas que se le han dado. Variable adicionada para posteriormente generar evaluaciones del tipo adaptativo.No se utilizará por el momento en este proyecto. }

verdadero\_falso: Boolean { Sólo se usa en caso de que tipo\_respuesta=4. Si es verdadero, quiere decir que la respuesta del reactivo es verdadero, y si es falso, la respuesta es falsa. }

Opcion\_A : A255 { En caso de que tipo\_respuesta=2 es una opción como respuesta, y es la respuesta correcta. En caso de que tipo\_respuesta=4, es la opción "Verdadero" }

Opcion\_B : A255 { En caso de que tipo\_respuesta=2 es una opción (no correcta) como respuesta. En caso de que tipo\_respuesta=4, es la opción "Falso" }

Opcion\_C : A255 { En caso de que tipo\_respuesta=2 es una opción (no correcta) como respuesta }

Opcion\_D : A255 { En caso de que tipo\_respuesta=2 es una opción (no correcta) como respuesta }

Opcion\_E : A255 { En caso de que tipo\_respuesta=2 es una opción (no correcta) como respuesta }

RecomA : A150 { Recomendación en caso de acierto }

RecomF : A150 { Recomendación en caso de falla }

Fecha : DDMMAAAA { Fecha en que se creó el reactivo }

#### Llaves archivo REACTIVOS:

Llave0 : Idreactivo (única, Primaria)

Llave1 : Cvemat, Ntema, Idreactivo (única)

## A9. Archivo de Evaluaciones.

Evaluaciones: Nombre del archivo: EVALUACION

Periodo : A3 { Periodo (Semestre o Trimestre) en el que se aplicará la evaluación

FORMATO Y VALORES VALIDOS:  
 AAP, donde:  
 AA = Año (94,95,etc)  
 P = Periodo Semestral o Trimestral  
 1 = Sem. Enero-May o Trim. Ene-Abr.  
 2 = Verano Semestral o Trim. Abr-Jun.  
 3 = Sem. Agosto-Dic o Verano Trim.  
 4 = Trim. Sep-Dic. }

TipPer : A1 { Tipo de Periodo

FORMATO Y VALORES VALIDOS:  
 S = Periodo Semestral  
 T = Periodo Trimestral }

CveMat : A8 { Clave de la materia a la que pertenece la evaluación }

Grupo : A2 { Número de grupo de la materia, al que se le aplicará la evaluación }

IdEval : A10 { Número consecutivo de la evaluación. Por cada materia se numerarán las evaluaciones }

NumPreg : N3 { Número de preguntas que tiene la evaluación }

TipoEval : A1 { Tipo de evaluación:  
 Valores válidos: A = Aleatorio D= Dirigido, con reactivos asignados  
 En este proyecto se implantará sólo el tipo de evaluación Dirigido }

RFCautor : A10 { RFC del profesor o autor que diseñó la evaluación }

Descripcion: A120 { Descripción de la evaluación }

TiempoMax : N3 { Número de segundos máximo que se tiene para contestar la evaluación. Para este proyecto no se usará, las evaluaciones se pueden contestar sin límite de tiempo }

Navegable : A1 { Si se puede navegar por toda la evaluación. Para este proyecto no se usará, las preguntas tienen que ser contestadas una por una en el orden en que aparecen, y no se puede regresar.  
Valores válidos: S = Sí es navegable N=No es navegable }

Llaves archivo EVALUACION:

Llave0 : Periodo, TipPer, Cvemat, Grupo, Idevaluacion (única)  
Llave1 : Idevaluacion (única, primaria)

## A10. Archivo de Reactivos-Evaluación.

Reactivos-Evaluación: Nombre del archivo: EVAL-REAC

IdEval : A10 { Identificador de la evaluación }

TipoReac : A1 { Tipo de reactivo.  
Valores válidos: A = Aleatorio de algún tema y materia especificada  
D = Dirigido.  
En este proyecto se implantará sólo el Dirigido }

IdReactivo : A10 { Identificador del reactivo. Sólo se usa en caso de que sea dirigido. }

CveMat : A8 { Sólo se usa en caso de que el tipo de reactivo sea aleatorio. Es la clave de la materia de donde se seleccionará el reactivo }

NTema : A20 { Sólo se usa en caso de que el tipo de reactivo sea aleatorio.  
Es el número de tema de dicha materia de donde va a seleccionar el reactivo }

Pond : Real { Ponderación en porcentaje de este reactivo para la evaluación final }

PonrecA : Boolean { Bandera para poner la Recomendación del Reactivo en caso de acierto:  
Valores válidos: Verdadero = Sí poner la recomendación  
Falso = No poner la recomendación }

PonrecF : Boolean { Bandera para poner la Recomendación del Reactivo en caso de fallo:  
Valores válidos: Verdadero = Sí poner la recomendación especificada en el archivo de reactivos  
Falso = No poner la recomendación }

Llaves archivo EVAL-REAC:

Llave0 : Ideval (No única)

## A11. Archivo de Reglas

Evaluación-Reglas: Nombre del archivo: REGLAS

IdEval : A10 { Identificador de la evaluación }

Reglas : A& { Texto de las reglas de conocimiento a aplicar al final de la evaluación.  
El formato será el lenguaje para definir reglas, que se explicará más adelante }

Llaves archivo REGLAS:

Llave0 : Ideval (única, primaria)

## A12. Archivo de Bitácora

Bitácora: Nombre de la tabla: BITACORA  
Periodo : A3 { Semestre o Periodo en el que está inscrito.

Notación: AAS, donde:  
 AA: Año  
 S: Semestre:  
 Para Prog. Semestral: 1=Ene-May, 2=Verano, 3=Ago-Dic }

TipPer : A1 { Tipo de Periodo: S= Semestral T=Trimestral }  
 IdEval : A10 { Número consecutivo de la evaluación. Por cada materia se numerarán las evaluaciones }  
 Matricula : A6 { Matrícula del alumno que contestó el examen }  
 Numpreg : N2 { Número de preguntas de la evaluación }  
 Numa : N2 { Número de preguntas contestadas correctamente }  
 Numf : N2 { Número de preguntas mal contestadas }  
 Calif : A2 { Calificación en escala de 01 a 10 }  
 Tiempo : N4 { Tiempo en segundos que tardó en contestar la evaluación }  
 Recomen : A&{ Recomendaciones que se le dieron al final de la prueba }  
 Fecha : DDMMAAAA { Fecha en que contestó la evaluación }

Llaves archivo BITACORA:

Llave0 : Periodo, TipPer, Ideval, Matricula (No única)  
 Llave1 : Matricula, Periodo, TipPer, Ideval (No única)  
 Llave2 : Matricula

### A13. Archivo de Detalle Bitácora.

Detalle de bitácora: Nombre del archivo: DETBIT  
 Periodo : A3 { Semestre o Periodo en el que está inscrito.  
 Notación: AAS, donde:  
 AA: Año  
 S: Semestre:  
 Para Prog. Semestral: 1=Ene-May, 2=Verano, 3=Ago-Dic }

TipPer : A1 { Tipo de Periodo: S= Semestral T=Trimestral }  
 Matricula : A6 { Matrícula del alumno que contestó la pregunta }  
 IdEval : A10 { Identificador de la evaluación que contestó }  
 IdReactivo : A10 { Identificador del reactivo que contestó }  
 Acierto : A1 { Si acertó o no.  
 Valores válidos: S = Sí acertó N = No acertó }  
 Pond : Real { Porcentaje de ponderación que tuvo el reactivo en la evaluación }  
 Fecha : DDMMAAAA { Fecha en que se contestó el reactivo }

Llaves archivo DETBIT:

Llave0 : Periodo, TipPer, Ideval, Idreactivo, Matricula (No única)

## ANEXO B. REGLAS PARA RECOMENDACIONES PARA LA PRUEBA DE APTITUD ACADEMICA

Estas reglas fueron definidas para probar el *software* EVALUATEC. Estas reglas pertenecen a una evaluación similar a la Prueba de Aptitud Académica. Los reactivos fueron numerados empezando por el 1000000001.

Si aciertos(1000000001,1000000002,1000000003,  
1000000004,1000000005,1000000006)<=1 entonces falla1(100%)

Si aciertos(1000000001,1000000002,1000000003,  
1000000004,1000000005,1000000006)=2 entonces falla1(80%)

Si aciertos(1000000001,1000000002,1000000003,  
1000000004,1000000005,1000000006)=3 entonces falla1(60%)

Si aciertos(1000000001,1000000002,1000000003,  
1000000004,1000000005,1000000006)=4 entonces falla1(30%)

Si aciertos(1000000001,1000000002,1000000003,  
1000000004,1000000005,1000000006)=5 entonces falla1(20%)

Si aciertos(1000000007,1000000008,1000000009,  
1000000010,1000000011,1000000012)<=1 entonces falla2(100%)

Si aciertos(1000000007,1000000008,1000000009,  
1000000010,1000000011,1000000012)=2 entonces falla2(80%)

Si aciertos(1000000007,1000000008,1000000009,  
1000000010,1000000011,1000000012)=3 entonces falla2(60%)

Si aciertos(1000000007,1000000008,1000000009,  
1000000010,1000000011,1000000012)=4 entonces falla2(30%)

Si aciertos(1000000007,1000000008,1000000009,  
1000000010,1000000011,1000000012)=5 entonces falla2(20%)

Si aciertos(1000000013,1000000014,1000000015,  
1000000016,1000000017)<=1 entonces falla3(100%)

Si aciertos(1000000013,1000000014,1000000015,  
1000000016,1000000017)=2 entonces falla3(80%)

Si aciertos(1000000013,1000000014,1000000015,  
1000000016,1000000017)=3 entonces falla3(40%)

Si aciertos(1000000013,1000000014,1000000015,  
1000000016,1000000017)=4 entonces falla3(20%)

Si aciertos(1000000018,1000000019,1000000020,  
1000000021,1000000022,1000000023)<=1 entonces falla4(100%)

Si aciertos(1000000018,1000000019,1000000020,  
1000000021,1000000022,1000000023)=2 entonces falla4(80%)

Si aciertos(1000000018,1000000019,1000000020,  
1000000021,1000000022,1000000023)=3 entonces falla4(60%)

Si aciertos(1000000018,1000000019,1000000020,  
1000000021,1000000022,1000000023)=4 entonces falla4(30%)

Si aciertos(1000000018,1000000019,1000000020,  
1000000021,1000000022,1000000023)=5 entonces falla4(20%)

Si aciertos(1000000024,1000000025,1000000026,  
1000000027,1000000028,1000000029)<=1 entonces falla5(100%)

Si aciertos(1000000024,1000000025,1000000026,  
1000000027,1000000028,1000000029)=2 entonces falla5(80%)  
Si aciertos(1000000024,1000000025,1000000026,  
1000000027,1000000028,1000000029)=3 entonces falla5(60%)  
Si aciertos(1000000024,1000000025,1000000026,  
1000000027,1000000028,1000000029)=4 entonces falla5(30%)  
Si aciertos(1000000024,1000000025,1000000026,  
1000000027,1000000028,1000000029)=5 entonces falla5(20%)

Si aciertos(1000000030,1000000031,1000000032,  
1000000033,1000000034)<=1 entonces falla6(100%)  
Si aciertos(1000000030,1000000031,1000000032,  
1000000033,1000000034)=2 entonces falla6(80%)  
Si aciertos(1000000030,1000000031,1000000032,  
1000000033,1000000034)=3 entonces falla6(40%)  
Si aciertos(1000000030,1000000031,1000000032,  
1000000033,1000000034)=4 entonces falla6(20%)

Si aciertos(1000000035,1000000044,1000000046,  
1000000050,1000000055,1000000067)<=1 entonces fallam1  
Si aciertos(1000000035,1000000044,1000000046,  
1000000050,1000000055,1000000067)=3 entonces fallam1 (50%)  
Si aciertos(1000000035,1000000044,1000000046,  
1000000050,1000000055,1000000067)=5 entonces fallam1 (20%)

Si aciertos(1000000036,1000000039,1000000065,  
1000000069,1000000070,1000000073)<=1 entonces fallam2  
Si aciertos(1000000036,1000000039,1000000065,  
1000000069,1000000070,1000000073)=2 entonces fallam2(70%)  
Si aciertos(1000000036,1000000039,1000000065,  
1000000069,1000000070,1000000073)=3 entonces fallam2(50%)  
Si aciertos(1000000036,1000000039,1000000065,  
1000000069,1000000070,1000000073)>3 y  
aciertos(1000000036,1000000039,1000000065,  
1000000069,1000000070,1000000073)<6 entonces fallam2(20%)

Si aciertos(1000000036,1000000039,1000000053,  
1000000054,1000000058,1000000059,  
1000000062,1000000069,1000000075)<=2 entonces fallam3  
Si aciertos(1000000036,1000000039,1000000053,  
1000000054,1000000058,1000000059,  
1000000062,1000000069,1000000075)=3 entonces fallam3(70%)  
Si aciertos(1000000036,1000000039,1000000053,  
1000000054,1000000058,1000000059,  
1000000062,1000000069,1000000075)=4 entonces fallam3(55%)  
Si aciertos(1000000036,1000000039,1000000053,  
1000000054,1000000058,1000000059,  
1000000062,1000000069,1000000075)=5 entonces fallam3(45%)  
Si aciertos(1000000036,1000000039,1000000053,  
1000000054,1000000058,1000000059,  
1000000062,1000000069,1000000075)=6 entonces fallam3(30%)  
Si aciertos(1000000036,1000000039,1000000053,  
1000000054,1000000058,1000000059,  
1000000062,1000000069,1000000075)>6 y  
aciertos(1000000036,1000000039,1000000053,  
1000000054,1000000058,1000000059,  
1000000062,1000000069,1000000075)<9 entonces fallam3(20%)

Si aciertos(100000035,100000044,100000046)=0 entonces fallam4  
Si aciertos(100000035,100000044,100000046)=1 entonces fallam4(66%)  
Si aciertos(100000035,100000044,100000046)=2 entonces fallam4(20%)

Si aciertos(100000048,100000051)=0 entonces fallam5  
Si aciertos(100000048,100000051)=1 entonces fallam5(50%)

Si reactivo 100000062 es incorrecto entonces fallam6  
Si reactivo 100000066 es incorrecto entonces fallam7  
Si reactivo 100000068 es incorrecto entonces fallam8  
Si aciertos(100000040,100000054,100000060,  
100000072)<=1 entonces fallam9  
Si aciertos(100000040,100000054,100000060,  
100000072)=2 entonces fallam9(50%)  
Si aciertos(100000040,100000054,100000060,  
100000072)=3 entonces fallam9(20%)

Si aciertos(100000042,100000045,100000052,  
100000057,100000063,100000064,  
100000067,100000071)<=2 entonces fallam10  
Si aciertos(100000042,100000045,100000052,  
100000057,100000063,100000064,  
100000067,100000071)=3 entonces fallam10(65)  
Si aciertos(100000042,100000045,100000052,  
100000057,100000063,100000064,  
100000067,100000071)=4 entonces fallam10(50)  
Si aciertos(100000042,100000045,100000052,  
100000057,100000063,100000064,  
100000067,100000071)=5 entonces fallam10(30)  
Si aciertos(100000042,100000045,100000052,  
100000057,100000063,100000064,  
100000067,100000071)>5 y  
aciertos(100000042,100000045,100000052,  
100000057,100000063,100000064,  
100000067,100000071)<8 entonces fallam10(20)

Si aciertos(100000043,100000047,100000072)=0 entonces fallam11  
Si aciertos(100000043,100000047,100000072)=1 entonces fallam11(66%)  
Si aciertos(100000043,100000047,100000072)=2 entonces fallam11(20%)

Si aciertos(100000063,100000064)=0 entonces fallam12  
Si aciertos(100000063,100000064)=1 entonces fallam12(50%)

Si reactivo 100000076 es incorrecto entonces fallam13  
Si aciertos(100000038,100000041,100000049,  
100000061,100000065,100000074)<=1 entonces fallam14  
Si aciertos(100000038,100000041,100000049,  
100000061,100000065,100000074)=2 entonces fallam14(70%)  
Si aciertos(100000038,100000041,100000049,  
100000061,100000065,100000074)=3 entonces fallam14(50%)  
Si aciertos(100000038,100000041,100000049,  
100000061,100000065,100000074)>3 y  
aciertos(100000038,100000041,100000049,  
100000061,100000065,100000074)<6 entonces fallam14(20%)

Si aciertos(1000000059,1000000065,1000000067,  
1000000070,1000000071,1000000073)<=1 entonces fallam15  
Si aciertos(1000000059,1000000065,1000000067,  
1000000070,1000000071,1000000073)=2 entonces fallam15(70%)  
Si aciertos(1000000059,1000000065,1000000067,  
1000000070,1000000071,1000000073)=3 entonces fallam15(50%)  
Si aciertos(1000000059,1000000065,1000000067,  
1000000070,1000000071,1000000073)>3 y  
aciertos(1000000059,1000000065,1000000067,  
1000000070,1000000071,1000000073)<6 entonces fallam15(20%)

Si fallam1 y fallam4 y fallam5 y fallam6 y fallam7 entonces fallaaritmetica  
Si fallam2 y fallam3 y fallam8 entonces fallaalgebra

Si fallam9 y fallam10 y fallam11 y fallam12 y fallam13 entonces fallageometria  
Si fallam14 y fallam15 entonces fallalogica

Si fallaaritmetica entonces Recomendacion 'Falta habilidad en Aritmetica en general'  
Si fallaalgebra entonces Recomendación 'Falta habilidad en Algebra en general'  
Si fallalogica entonces Recomendación 'Falta habilidad en Lógica en general'

Si fallam1 entonces Recomendación 'En específico en Aritmética te falta habilidad en Operaciones Básicas'  
Si fallam4 entonces Recomendación 'En específico en Aritmética te falta habilidad en Divisibilidad'  
Si fallam5 entonces Recomendación 'En específico en Aritmética te falta habilidad en Medidas de tendencia central'  
Si fallam6 entonces Recomendación 'En específico en Aritmética te falta reconocer clases de números'

Si fallam7 entonces Recomendación 'En específico en Aritmética te falta habilidad en manejar porcentajes'  
Si fallam2 entonces Recomendación 'En específico en Algebra te falta habilidad en Substitución'  
Si fallam3 entonces Recomendación 'En específico en Algebra te falta habilidad en Despejes'  
Si fallam8 entonces Recomendación 'En específico en Algebra te falta habilidad en Factorización'

Si fallam9 entonces Recomendación 'En específico en Geometría te falta habilidad en Areas'  
Si fallam10 entonces Recomendación 'En específico en Geometría te falta habilidad en Propiedades de Triángulos'  
Si fallam11 entonces Recomendación 'En específico en Geometría te falta habilidad en Perímetros'  
Si fallam12 entonces Recomendación 'En específico en Geometría te falta habilidad en Ángulos'  
Si fallam13 entonces Recomendación 'En específico en Geometría te falta habilidad en Propiedades de Círculos'

Si fallam14 entonces Recomendación 'En específico en Lógica te falta habilidad en Relaciones de proporción'  
Si fallam15 entonces Recomendación 'En específico en Lógica te falta habilidad en Relaciones lógicas'

Si falla1 y falla2 y falla3 y falla4 y falla5 y falla6 y falla7  
entonces Recomendación ' Fallaste en casi toda el área Verbal, Repasa toda la parte verbal '  
Si falla1 entonces Recomendación ' Repasar Asociación lógica de ideas relacionadas'  
Si falla2 o falla5 entonces Recomendación ' Repasar Antónimos y vocabulario '  
Si falla3 entonces Recomendación ' Repasar Relaciones entre palabras '  
Si falla4 entonces Recomendación ' Repasar Completar oraciones lógicas'  
Si falla6 entonces Recomendación ' Repasar Analogías entre relaciones '

Si aciertos(1000000035,1000000036)=0 entonces falla7 (100%)  
Si aciertos(1000000035,1000000036)=0 entonces falla7 (50%)

Si calificación<7.3 entonces Recomendación 'No aprobaste el examen, fuiste rechazado'  
Si calificación>=7.3 entonces Recomendación 'Felicidades, aprobaste el examen'

## ANEXO C. CÓDIGO FUENTE DEL SOFTWARE.

Lenguaje: Visual Basic 3.0.

Nota: No se especifica el código que Visual Basic genera para cada objeto de cada forma (botones, capturas, etc.). Sólo se muestra el código que se desarrolló para este proyecto.

### C1. PREGLAS: Programa para crear las reglas.

Este programa consta de tres formas y 3 archivos de código:

- Freglas.frm
- Fprereg.frm
- Ffin2.frm
- archreg.bas
- compilar.bas

#### C1.1. Freglas.frm.

```
Sub Combo1_DropDown ()
    Evaluacion.Index = "Llave0"
    Evaluacion.MoveFirst
    Combo1.Clear
    While Not Evaluacion.EOF
        Combo1.AddItem Evaluacion(4)
        Evaluacion.MoveNext
    Wend
End Sub
```

```
Sub Combo1_KeyPress (keyAscii As Integer)
    If keyAscii = 13 Then
        DatosEvaluacion (Combo1.Text)
    End If
End Sub
```

```
Sub Combo1_LostFocus ()
    If Combo1.Text <> "" Then
        DatosEvaluacion (Combo1.Text)
    End If
End Sub
```

```
Sub Command1_Click ()
    Aplicareglas
    If Not e_error Then
        MsgBox (" Sintaxis OK")
    End If
End Sub
```

```
Sub Command2_Click ()
    RegistraReglas
    Text1.SetFocus
End Sub
```

```

Sub Command3_Click ()
Dim s As Integer
s = MsgBox("Guardas los cambios de las reglas?", 3)
If s = 6 Then
    RegistraReglas
End If
If s <> 2 Then
    CierraArchivos
End
End If
End Sub

```

```

Sub Command3D1_Click (Index As Integer)
Dim s As Integer
Dim i As Integer
If Index = 1 Then
s = MsgBox("Guardas los cambios de las reglas?", 3)
If s = 6 Then
    RegistraReglas
ElseIf s <> 2 Then
Text1.Text = ""
Combo1.Clear
'While Combo1.ListCount > 0
' Combo1.RemoveItem 0
' Wend
Combo1.Text = ""

For i = 0 To 5
    Label5(i).Caption = ""
Next i
List1.Clear
Combo1.SetFocus
End If
ElseIf Index = 0 Then
    RegistraReglas
    Text1.SetFocus
ElseIf Index = 2 Then
s = MsgBox("Guardas los cambios de las reglas?", 3)
If s = 6 Then
    RegistraReglas
End If
If s <> 2 Then
    CierraArchivos
End
End If
End If
End Sub

```

```

Sub Command3D2_Click (Index As Integer)
Select Case Index
Case 0
    Aplicareglas
    If Not e_error Then
        MsgBox (" Sintaxis OK")
    End If
Case 1
    Inicializa

```

```

Case 2
If Not compilaste Then
  MsgBox ("Primero tienes que checar sintaxis (compilar) ")
Else
  If Not (e_error) Then
    RazonaForward
    Darecomendaciones
  Else
    MsgBox ("Hubo error al compilar")
  End If
End If
End Select
End Sub

```

```

Sub Command4_Click ()
If Not compilaste Then
  MsgBox ("Primero tienes que checar sintaxis (compilar) ")
Else
  If Not (e_error) Then
    RazonaForward
    Darecomendaciones
  Else
    MsgBox ("Hubo error al compilar")
  End If
End If
End Sub

```

```

Sub Command5_Click ()
Dim i As Integer
Text1.Text = ""
Combo1.Clear
'While Combo1.ListCount > 0
' Combo1.RemoveItem 0
' Wend
Combo1.Text = ""

For i = 0 To 5
  Label5(i).Caption = ""
Next i
List1.Clear
Combo1.SetFocus
End Sub

```

```

Sub Command6_Click ()
  Inicializa
End Sub

```

```

Sub Form_Load ()
  Inicializa
  AbreArchivos
End Sub

```

```

Sub List1_Db1Click ()
Dim idreactivo As String * 10
idreactivo = Mid$(List1.List(List1.ListIndex), 1, 10)
DatosReactivo (idreactivo)
End Sub

```

## C1.2. Fprereg.frm

```
Sub Command1_Click (Index As Integer)
    InformacionBotones (Index)
End Sub
```

## C1.3. Ffin2.frm

No tiene código.

## C1.4. Archreg.bas

```
Option Explicit
Global BDSISEVAL As DATABASE
Global Reglas As table
Global evaluacion As table
Global Evareact As table
Global reactivos As table
Global referencias As table
Global Autor As table
Global Sesmater As table
Global sesgrupo As table
Global panalitico As table
Global Idevaluacion As String * 10
Global Esnuevo As Integer
Global totpreg As Integer
Global compilaste As Integer
Global Const maxreactivos = 500
Global Const maxpreg = 100
Type Tarreglo
    idreactivo As String * 10 ' identificador del reactivo
    acierto As Integer 'si acertó el reactivo: 1=correcto, 0=no contestó, -1=incorrecto
End Type
Type Tpreg
    nombre_prop As String
    preg As String
    valor_verdad As Integer
End Type
Global TablaP(1 To maxpreg) As Tpreg
Global Arreglo(1 To maxreactivos) As Tarreglo
Global indarreglo As Integer
Global indpregunta As Integer
Global calificacion As Single

Sub AbreArchivos ()
    Set BDSISEVAL = OpenDatabase("c:\vb\siseval\SISEVAL.MDB")
    Set Reglas = BDSISEVAL.OpenTable("REGLAS")
    Set evaluacion = BDSISEVAL.OpenTable("EVALUACION")
    Set Evareact = BDSISEVAL.OpenTable("EVAL-REAC")
    Set reactivos = BDSISEVAL.OpenTable("REACTIVOS")
    Set Autor = BDSISEVAL.OpenTable("AUTOR")
    Set Sesmater = BDSISEVAL.OpenTable("SESMATER")
```

```

Set sesgrupo = BDSISEVAL.OpenTable("SESGRUPO")
Set panalitico = BDSISEVAL.OpenTable("Panalitico")
End Sub

```

```

Sub CierraArchivos ()
Reglas.Close
evaluacion.Close
Evareact.Close
reactivos.Close
Autor.Close
Sesmater.Close
sesgrupo.Close
panalitico.Close
BDSISEVAL.Close
End Sub

```

```

Sub DatosEvaluacion (Ideval As String)
Dim i As Integer
Dim tipo_eval As String * 1
Dim st As String

```

```

    compilaste = False
    Idevaluacion = Ideval
    evaluacion.Index = "Llave1"
    evaluacion.Seek "=", Idevaluacion
    If evaluacion.NoMatch Then
        MsgBox ("No existe esa evaluacion")
        For i = 0 To 5
            Form1.Label5(i).Caption = ""
        Next i
    Else
        Form1.Label5(0).Caption = evaluacion(8)
        Form1.Label5(1).Caption = Format$(evaluacion(5))
        tipo_eval = evaluacion(6)
        Select Case tipo_eval
            Case "D"
                Form1.Label5(2).Caption = "D=Dirigido, con reactivos seleccionados"
            Case "A"
                Form1.Label5(2).Caption = "A=Aleatorio"
        End Select
        sesgrupo.Index = "Llave0"
        sesgrupo.Seek "=", evaluacion(0), evaluacion(1), evaluacion(2), evaluacion(3)
        If sesgrupo.NoMatch Then
            Form1.Label5(3).Caption = "No existe grupo asignado"
        Else
            st = evaluacion(2) + "-" + evaluacion(3)
            Sesmater.Index = "Llave0"
            Sesmater.Seek "=", evaluacion(2)
            If Sesmater.NoMatch Then
                st = st + " No existe materia "
            Else
                On Error Resume Next
                st = st + Sesmater(1) + " " + sesgrupo(10) + sesgrupo(11) + sesgrupo(12)
                If Error Then st = st
            End If
            Form1.Label5(3).Caption = st
        End If
    End Sub

```

```

End If
Form1.Label5(4).Caption = StPeriodo(evaluacion(0), evaluacion(1))
Autor.Index = "Llave0"
Autor.Seek "=", evaluacion(7)
If Autor.NoMatch Then
    Form1.Label5(5).Caption = evaluacion(7) + " ???"
Else Form1.Label5(5).Caption = evaluacion(7) + " " + Autor(1)
End If
End If
End If
DatosReacEval
ReglasEval
End Sub

```

```

Sub DatosReacEval ()
Dim sigue As Integer
Dim st As String
totpreg = 0
Evareact.Index = "Llave0"
Evareact.Seek "=", Idevaluacion
sigue = 1
If Not Evareact.NoMatch Then
While sigue
totpreg = totpreg + 1
reactivos.Index = "Llave0"
st = Evareact(2)
reactivos.Seek "=", st
If Not reactivos.NoMatch Then
If (reactivos(4) <> Null) Then
st = st + " - " + reactivos(4) + " - " + reactivos(5)
Else
st = st + " - " + reactivos(5)
End If
Form1.List1.AddItem st
Else Form1.List1.AddItem st
End If
Evareact.MoveNext
If Not Evareact.EOF Then
sigue = (Evareact(0) = Idevaluacion)
Else sigue = 0
End If
Wend
End If

```

End Sub

```

Sub DatosReactivo (idreactivo As String)
Dim i As Integer
Form2.Show
reactivos.Index = "Llave0"
reactivos.Seek "=", idreactivo
If reactivos.NoMatch Then
MsgBox ("Error, No existe el reactivo, inconsistencia con la Evaluación")
Else
Form2.Label5(0) = idreactivo
If reactivos(4) <> Null Then
Form2.Label5(1) = reactivos(4)
Else

```

```

Form2.Label5(1) = ""
End If
If reactivos(11) <> "" Then
Form2.Image1.Visible = True
On Error Resume Next

Form2.Image1.Picture = LoadPicture("c:\windows\" + reactivos(11))
If Err Then
MsgBox ("No existe el gráfico c:\windows\" + reactivos(11) + " asociado a este reactivo")
Form2.Image1.Visible = False
End If
End If
Select Case reactivos(6)
Case "1"
Form2.Label5(2).Caption = "1=Opciones generadas"
Case "2"
Form2.Label5(2).Caption = "2=Opciones múltiples"
Case "3"
Form2.Label5(2).Caption = "3=Valor específico"
Case "4"
Form2.Label5(2).Caption = "4=False/Verdadero"
End Select
Sesmater.Index = "Llave0"
Sesmater.Seek "=", reactivos(0)
If Sesmater.NoMatch Then
Form2.Label5(3).Caption = "No existe materia ??"
Else
Form2.Label5(3).Caption = reactivos(0) + " " + Sesmater(1)
End If
If reactivos(1) <> "" Then
panalitico.Index = "Llave0"
panalitico.Seek "=", reactivos(0), reactivos(1)
If panalitico.NoMatch Then
Form2.Label5(4).Caption = reactivos(1) + " no existe este tema en la materia"
Else
Form2.Label5(4).Caption = reactivos(1) + " " + panalitico(2)
End If
Else
Form2.Label5(4).Caption = "Tema no definido"
End If
If reactivos(3) <> "" Then
Autor.Index = "Llave0"
Autor.Seek "=", reactivos(3)
If Autor.NoMatch Then
Form2.Label5(5).Caption = reactivos(3) + " no existe en Autor "
Else
Form2.Label5(5).Caption = reactivos(3) + " " + Autor(1)
End If
Else
Form2.Label5(5).Caption = "Autor no definido"
End If
Form2.Label1.Caption = reactivos(5)
For i = 0 To 4
Form2.Label4(i).Visible = True
Form2.Command3D1(i).Visible = True
Next i
Select Case reactivos(6)

```

```

Case 2
Form2.Label4(0).Caption = reactivos(15)
Form2.Label4(1).Caption = reactivos(16)
Form2.Label4(2).Caption = reactivos(17)
Form2.Label4(3).Caption = reactivos(18)
Form2.Label4(4).Caption = reactivos(19)
Form2.Label6.Caption = "A)"
Case 3
For i = 0 To 4
Form2.Label4(i).Visible = False
Form2.Command3D1(i).Visible = False
Next i
Form2.Label6.Caption = reactivos(7)
Case 4
Form2.Label4(0).Visible = False
Form2.Label4(1).Caption = "Veradero"
Form2.Label4(2).Caption = "Falso"
Form2.Label4(3).Visible = False
Form2.Label4(4).Visible = False
Form2.Command3D1(0).Visible = False
Form2.Command3D1(1).Caption = "V)"
Form2.Command3D1(2).Caption = "F)"
Form2.Command3D1(3).Visible = False
Form2.Command3D1(4).Visible = False
If reactivos(14) Then
Form2.Label6.Caption = "Verdadero"
Else Form2.Label6.Caption = "Falso"
End If
End Select
End If
End Sub

Sub InformacionBotones (Index As Integer)
Dim dato, Dato1, Dato2 As Variant
Dim st As String
Dim X

Select Case Index
Case 0
dato = reactivos(8)
If Len(dato) > 0 Then
MsgBox (dato)
Else MsgBox ("No hay ayuda especificada")
End If

Case 1
dato = reactivos(9)
If Len(dato) > 0 Then
X = Shell(dato)
Else MsgBox ("No hay tutorial especificado")
End If

Case 2
dato = reactivos(23)
If Len(dato) > 0 Then
referencias.Index = "Llave0"
referencias.Seek "=", dato

```

```

If referencias.NoMatch Then
    MsgBox "No existe la referencia especificada"
Else
    st = referencias(1) + referencias(2)
    If st <> "" Then
        MsgBox (st)

        End If
    End If
Else
    MsgBox "No hay referencia especificada"
End If
Case 3
    Dato1 = reactivos(20)
    Dato2 = reactivos(21)
    If Len(Dato1) > 0 Then
        dato = "Recomendación si acierta:" + Dato1 + Chr(13)
    Else dato = "No hay recomendación si acierta " + Chr(13)
    End If
    If Len(Dato2) > 0 Then
        dato = dato + "Recomendación si falla:" + Dato2
    Else dato = dato + "No hay recomendación si falla"
    End If
    MsgBox (dato)
Case 4
    Dato1 = reactivos(12)
    Dato2 = reactivos(13)
    If Len(Dato1) > 0 Then
        dato = "Discriminación: " + Dato1 + Chr(13)
    Else dato = "No hay discriminación" + Chr(13)
    End If
    If Len(Dato2) > 0 Then
        dato = "Dificultad: " + Dato2 + Chr(13)
    Else dato = "No hay dificultad" + Chr(13)
    End If
    MsgBox (dato)
Case 5
    Unload Form2
End Select

End Sub

Sub Inicializa ()
Dim i As Integer
calificacion = -1
indarreglo = 0
For i = 1 To maxreactivos
    Arreglo(i).idreactivo = ""
    Arreglo(i).acierto = 0
Next i
indpregunta = 0
For i = 1 To maxpreg
    TablaP(i).preg = ""
    TablaP(i).valor_verdad = 0
Next i
End Sub

```

```

Sub RegistraReglas ()
If Esnuevo Then
  Reglas.AddNew
Else Reglas.Edit
End If
Reglas(0) = Idevaluacion
Reglas(1) = Form1.Text1.Text
Reglas.Update
End Sub

Sub ReglasEval ()
Idevaluacion = Form1.Combo1.Text
Reglas.Index = "Llave0"
Reglas.Seek "=", Idevaluacion
Esnuevo = 1
If Not Reglas.NoMatch Then
  Esnuevo = 0
  Form1.Text1.Text = Reglas(1)
End If
Form1.Text1.SetFocus

End Sub

Function StPeriodo (periodo, TipPer As Variant) As String
If TipPer = "S" Then
Select Case Mid$(periodo, 3, 1)
  Case "1"
    StPeriodo = "Enero-Mayo " + Mid$(periodo, 1, 2)
  Case "2"
    StPeriodo = "Verano " + Mid$(periodo, 1, 2)
  Case "3"
    StPeriodo = "Agosto-Dic. " + Mid$(periodo, 1, 2)
End Select
ElseIf TipPer = "T" Then
Select Case Mid$(periodo, 3, 1)
  Case "1"
    StPeriodo = "Enero-Abril " + Mid$(periodo, 1, 2)
  Case "2"
    StPeriodo = "Abril-Junio " + Mid$(periodo, 1, 2)
  Case "3"
    StPeriodo = "Julio-Agosto " + Mid$(periodo, 1, 2)
  Case "4"
    StPeriodo = "Sept-Dic. " + Mid$(periodo, 1, 2)
End Select
Else StPeriodo = " ??? "
End If
End Function

```

### **C1.5. Compilar.bas**

```
Option Explicit
```

```
Const Maxreglas = 500
Const Maxpremisas = 30
```

```
Dim indicearchivo, Longarchivo As Integer
```

```

Dim finarchivo, contlineas, atrib As Integer
Type Treglas
nombre_prop As String
valor_verdad As Integer 'valor de verdad en esta proposición
ul_valor_verdad As Integer 'valor de verdad combinado, en caso de aplicar
Visitado As Integer
concluido As Integer
negacion As Integer
Tipo As Integer '0= Hecho (si reactivo ok, si aciertos),
                '1= Proposición
                '2= Pregunta
End Type
Type Creglas
nombre_prop As String
valor_verdad As Integer
ul_valor_verdad As Integer 'valor de verdad combinado, en caso de aplicar
Visitado As Integer
concluido As Integer
conjuncion As Integer
TipoConc As Integer '0=Recomendación
                  '1=Proposición

valasig As Integer
negacion As Integer
End Type
Dim TablaR(1 To Maxpremisas, 1 To Maxreglas) As Treglas
Dim TablaC(1 To Maxreglas) As Creglas

Dim Xtabla, ytabla As Integer
Dim lookahead As String
Global e_error As Integer
Dim huborecomendaciones As Integer

Dim Entrada As Variant

Dim e, concluye, calculaVv, b, cambia As Integer
Dim v1, v2 As Integer
Dim valorv As Integer

Sub Aplicareglas ()
Dim i, j As Integer
e_error = False
compilaste = True
For i = 1 To Maxreglas
    TablaC(i).nombre_prop = "nil"
    For j = 1 To Maxpremisas
        TablaR(j, i).nombre_prop = "nil"
    Next j
Next i
indpregunta = 0
For i = 1 To Maxpreg
    TablaP(i).nombre_prop = ""
    TablaP(i).preg = ""
    TablaP(i).valor_verdad = 0
Next i
indicearchivo = 1
Entrada = Form1.Text1.Text

```

```

Longarchivo = Len(Entrada)
lookahead = Token()
ytabla = 0
contlineas = 1
huborecomendaciones = False
EscaneaReglas
End Sub

```

```

Function BuscaConclu (nombreprop As String) As Integer
Dim i, encuentra As Integer
encuentra = False
i = 0
While (i < ytabla) And Not (encuentra)
    i = i + 1
    encuentra = (TablaC(i).nombre_prop = nombreprop)
Wend
If encuentra Then
    BuscaConclu = i
Else
    BuscaConclu = 0
End If
End Function

```

```

Sub CicloAndF (indicey As Integer)
Dim iconclu, minvalorv, sigue, x, vv As Integer
v1 = 100
sigue = True
x = 1
While (Mid$(TablaR(x, indicey).nombre_prop, 1, 3) <> "nil") And (b) And (sigue)
    If (TablaR(x, indicey).Tipo = 1) Then
        iconclu = BuscaConclu(TablaR(x, indicey).nombre_prop)
        If iconclu > 0 Then
            If TablaC(iconclu).concluido Then
                If TablaR(x, indicey).negacion Then
                    minvalorv = -1 * TablaC(iconclu).ul_valor_verdad
                Else
                    minvalorv = TablaC(iconclu).ul_valor_verdad
                End If
            If minvalorv < v1 Then
                v1 = minvalorv
            End If
            If (v1 = -100) Then
                sigue = False
            End If
        Else
            b = False
            v1 = 0
            calculaVv = False
        End If
    Else
        MsgBox ("Warning, no se concluyó " + TablaR(x, indicey).nombre_prop)
    End If
ElseIf TablaR(x, indicey).Tipo = 2 Then 'si es una pregunta por hacer
    vv = Preguntar(TablaR(x, indicey).nombre_prop)
    If TablaR(x, indicey).negacion Then
        minvalorv = -1 * vv
    Else

```

```

    minvalorv = vv
End If
TablaR(x, indicey).Visitado = True
If minvalorv < v1 Then
    v1 = minvalorv
End If
If (v1 = -100) Then
    sigue = False
End If
ElseIf TablaR(x, indicey).Tipo = 0 Then 'si es hecho
    minvalorv = TablaR(x, indicey).valor_verdad ' aquí se puede insertar
        'para hacer preguntas al momento de proposiciones y
        'asignar el valor_verdad dependiendo de la respuesta
        'del usuario
    TablaR(x, indicey).Visitado = True
    If minvalorv < v1 Then
        v1 = minvalorv
    End If
    If (v1 = -100) Then
        sigue = False
    End If
Else
    b = False
    v1 = 0
    calculaVv = False
End If
x = x + 1
Wend
If (TablaC(indicey).valasig <> 100) And (v1 > 0) Then
    v1 = CInt(v1 * TablaC(indicey).valasig / 100)
End If
If v1 < 0 Then
    v1 = 0
End If
If TablaC(indicey).negacion Then
    v1 = -1 * v1
End If
End Sub

Sub CicloOrF (indicey As Integer)
Dim iconclu, maxvalorv, sigue, x, vv As Integer
v1 = -100
x = 1
sigue = True
While (Mid$(TablaR(x, indicey).nombre_prop, 1, 3) <> "nil") And (b) And (sigue)
    If (TablaR(x, indicey).Tipo = 1) Then
        iconclu = BuscaConclu(TablaR(x, indicey).nombre_prop)
        If iconclu > 0 Then
            If TablaC(iconclu).concluido Then
                If TablaR(x, indicey).negacion Then
                    maxvalorv = -1 * TablaC(iconclu).ul_valor_verdad
                Else
                    maxvalorv = TablaC(iconclu).ul_valor_verdad
                End If
            If maxvalorv > v1 Then
                v1 = maxvalorv
            End If
        End If
    End If
End While

```

```

    If (v1 = 100) Then
        sigue = False
    End If
Else
    b = False
    v1 = 0
    calculaVv = False
End If
Else
    MsgBox ("Warning, no se concluyó " + TablaR(x, indicey).nombre_prop)
End If
ElseIf TablaR(x, indicey).Tipo = 2 Then 'si es una pregunta por hacer
    vv = Preguntar(TablaR(x, indicey).nombre_prop)
    If TablaR(x, indicey).negacion Then
        maxvalorv = -1 * vv
    Else
        maxvalorv = vv
    End If
    TablaR(x, indicey).Visitado = True
    If maxvalorv > v1 Then
        v1 = maxvalorv
    End If
    If (v1 = 100) Then
        sigue = False
    End If
ElseIf (TablaR(x, indicey).Tipo = 0) Then
    maxvalorv = TablaR(x, indicey).valor_verdad
    TablaR(x, indicey).Visitado = True
    If maxvalorv > v1 Then
        v1 = maxvalorv
    End If
    If (v1 = 100) Then
        sigue = False
    End If
Else
    b = False
    v1 = 0
    calculaVv = False
End If
    x = x + 1
Wend
If (TablaC(indicey).valasig <> 100) And (v1 > 0) Then
    v1 = CInt(v1 * TablaC(indicey).valasig / 100)
End If
If v1 < 0 Then
    v1 = 0
End If
If TablaC(indicey).negacion Then
    v1 = -1 * v1
End If
End Sub

Function Combina (ByVal val1, val2 As Integer, nombre_prop As String) As Integer
' combina valores de certeza ya que existen 2 premisas con
' la misma conclusiøn
Dim minimo As Integer

```

```

If (val1 >= 0) And (val2 >= 0) Then '{ 2 factores positivos }
    val1 = val1 + CInt((val2 * (100 - val1)) / 100)
ElseIf (val1 < 0) And (val2 < 0) Then ' 2 factores negativos }
    val1 = -(Abs(val1) + ((Abs(val2) * CInt(100 - Abs(val1))) / 100))
Else '{ 1 negativo y otro positivo }
    minimo = Abs(val1)
    If Abs(val2) < minimo Then
        minimo = Abs(val2)
    End If
    If minimo = 100 Then
        MsgBox ("Hay una completa contradicción al concluir " + nombre_prop)
    End If
    val1 = CInt((val1 + val2) * 100) / (100 - minimo)
End If
Combina = val1
End Function

```

```

Function CualPregunta (nombre_prop As String, preg As String)
Dim i As Integer
i = 0
Do
    i = i + 1
Loop Until (i > indpregunta) Or (TablaP(i).nombre_prop = nombre_prop)
If (i <= indpregunta) Then
    CualPregunta = i
Else CualPregunta = 0
End If
If i <= indpregunta Then
    preg = TablaP(i).preg
Else preg = ""
End If
End Function

```

```

Function Cumpleaciertos (operador, valor As String, numaci As Integer) As Integer
Dim cvalor As Integer
cvalor = CInt(valor)
Select Case operador
    Case ">"
        Cumpleaciertos = (numaci > cvalor)
    Case ">="
        Cumpleaciertos = (numaci >= cvalor)
    Case "<"
        Cumpleaciertos = (numaci < cvalor)
    Case "<="
        Cumpleaciertos = (numaci <= cvalor)
    Case "="
        Cumpleaciertos = (numaci = cvalor)
    Case "<>"
        Cumpleaciertos = (numaci <> cvalor)
End Select

```

End Function

```

Function Cumplecalificacion (operador, valor As String) As Integer
Dim cvalor As Single

```

```

Dim s As String

```

```

cvalor = CDbI(valor)
If calificacion = -1 Then
  s = InputBox$("Dame la calificación: ")
  calificacion = CDbI(s)
End If
Select Case operador
  Case ">"
    Cumplecalificacion = (calificacion > cvalor)
  Case ">="
    Cumplecalificacion = (calificacion >= cvalor)
  Case "<"
    Cumplecalificacion = (calificacion < cvalor)
  Case "<="
    Cumplecalificacion = (calificacion <= cvalor)
  Case "="
    Cumplecalificacion = (calificacion = cvalor)
  Case "<>"
    Cumplecalificacion = (calificacion <> cvalor)
End Select
End Function

Sub Darecomendaciones ()
  Dim i As Integer
  Dim strecomen As String
  Form3.Show
  Form3.SetFocus
  strecomen = ""
  For i = 1 To ytabla
    If (TablaC(i).TipoConc = 0) And (TablaC(i).ul_valor_verdad >= 20) Then
      'snum = Format$(TablaC(i).ul_valor_verdad, 999)
      Form3.List1.AddItem "(" + Format$(TablaC(i).valor_verdad, "0##") + ")" + TablaC(i).nombre_prop)
      strecomen = strecomen + Chr(10) + Chr(13) + TablaC(i).nombre_prop + " (" +
        Format$(TablaC(i).valor_verdad) + ")"
    End If
  Next i
  If strecomen = "" Then
    If huborecomendaciones Then
      MsgBox ("No se obtuvieron Recomendaciones")
    Else
      MsgBox ("No existen reglas con Recomendaciones ")
    End If
  Else
    ' MsgBox ("Recomendaciones: " + Chr(10) + Chr(13) + strecomen)
  End If
End Sub

Sub EscaneaReglas ()
  If Not finarchivo Then
    Select Case LCase$(lookahead)
      Case "si"
        ParseaProposicion
        'If Not e_error Then
        EscaneaReglas
        'End If
      Case "pregunta"
        ParseaPregunta
        'If Not e_eror Then

```

```

    EscaneaReglas
  'End If
Case Else
  MError ("Error, espera Si o Pregunta ")
End Select
End If
End Sub

```

```

Function ExisteReactivo (idreact As String, acierto As Integer)
Dim i As Integer
i = 0
Do
  i = i + 1
Loop Until (i > indarreglo) Or (Arreglo(i).idreactivo = idreact)
ExisteReactivo = (i <= indarreglo)
If i <= indarreglo Then
  acierto = Arreglo(i).acierto
End If
End Function

```

```

Function ExisteSigProp (nomprop As String, ni As Integer) As Integer
Dim b As Integer
Dim i As Integer

b = False
i = 0
While Not (b) And (i < ytabla)

  i = i + 1
  If (TablaC(i).nombre_prop = nomprop) Then
    If Not TablaC(i).Visitado Then
      ni = i
      b = True
    End If
  End If
Wend
ExisteSigProp = b
End Function

```

```

Function Leecaracter () As String
Dim R As String * 1
Dim R2 As String * 1
R = ""
If indicearchivo <= Longarchivo Then
  R = LCase(Mid$(Entrada, indicearchivo, 1))
End If
indicearchivo = indicearchivo + 1
finarchivo = indicearchivo > Longarchivo
If indicearchivo > 2 Then
  R2 = LCase(Mid$(Entrada, indicearchivo - 2, 1))
Else R2 = ""
End If
If R2 = Chr(10) Then
  contlineas = contlineas + 1
End If
Leecaracter = R
End Function

```

```

Sub Match (actToken As String)
If Not (finarchivo) And Not (e_error) Then
  If (lookahead = actToken) Then
    '{ Despliega el número de líneas compiladas }
    'Form3.Print contlineas
    lookahead = Token()
  Else
    MError ("Espera " + actToken)
  End If
End If
End Sub

```

```

Sub MError (st As String)
  MsgBox (st + " en línea " + Format$(contlineas))
  e_error = -1
End Sub

```

```

Sub ParseaPregunta ()
Dim stpreg As String
Match ("pregunta")
If atrib <> 100 Then
  MError ("Espera una proposición")
ElseIf CualPregunta(lookahead, stpreg) Then
  MsgBox ("Ya está definida la pregunta para la proposición " + lookahead)
Else
  indpregunta = indpregunta + 1
  TablaP(indpregunta).nombre_prop = lookahead
  Match (lookahead)
  Match (""")
  stpreg = ""
  While (lookahead <> """) And Not (finarchivo)
    stpreg = stpreg + lookahead
    Match (lookahead)
  Wend
  TablaP(indpregunta).preg = stpreg
  Match (""")
End If
End Sub

```

```

Sub ParseaPremisa ()
Dim negacion, escorrecto, numaci As Integer
Dim operador, preg As String
If lookahead = "reactivo" Then
  Match ("reactivo")
  If atrib = 1 Or atrib = 100 Then
    TablaR(Xtabla, ytabla).nombre_prop = lookahead
    TablaR(Xtabla, ytabla).Visitado = 0
    TablaR(Xtabla, ytabla).concluido = 0
    TablaR(Xtabla, ytabla).Tipo = 0
    escorrecto = RespuestaReactivook(lookahead)
    Match (lookahead)
    Match ("es")
    If ((lookahead = "correcto") And (escorrecto = 1)) Or ((lookahead = "incorrecto") And (escorrecto = -1)) Then
      Match (lookahead)
      TablaR(Xtabla, ytabla).valor_verdad = 100
      TablaR(Xtabla, ytabla).ul_valor_verdad = 100
    End If
  End If
End Sub

```

```

ElseIf ((lookahead = "correcto") Or (lookahead = "incorrecto")) And ((escorrecto = -1) Or (escorrecto = 1)) Then
  Match (lookahead)
  TablaR(Xtabla, ytabla).valor_verdad = -100
  TablaR(Xtabla, ytabla).ul_valor_verdad = -100
ElseIf ((lookahead = "correcto") Or (lookahead = "incorrecto")) And (escorrecto = 0) Then
  Match (lookahead)
  TablaR(Xtabla, ytabla).valor_verdad = 0
  TablaR(Xtabla, ytabla).ul_valor_verdad = 0
Else
  MError ("Error, espera 'correcto' o 'incorrecto'")
End If
Else
  MError ("Error, espera número reactivo ")
End If 'if not reservada
ElseIf lookahead = "calificacion" Or lookahead = "calificación" Then
  Match (lookahead)
  operador = lookahead
  If atrib <> 255 Then
    MError ("Espera operando lógico (=,>, etc) ")
  Else
    Match (lookahead)
    If atrib <> 1 And atrib <> 2 Then
      MError ("Espera un número entero o real ")
    Else
      TablaR(Xtabla, ytabla).nombre_prop = "calificación"
      TablaR(Xtabla, ytabla).ul_valor_verdad = 0
      TablaR(Xtabla, ytabla).Visitado = 0
      TablaR(Xtabla, ytabla).concluido = 0
      TablaR(Xtabla, ytabla).Tipo = 0
      If Cumplecalificacion(operador, lookahead) Then
        TablaR(Xtabla, ytabla).valor_verdad = 100
      Else
        TablaR(Xtabla, ytabla).valor_verdad = -100
      End If
      Match (lookahead)
    End If
  End If
ElseIf lookahead = "aciertos" Then
  Match ("aciertos")
  Match ("")
  numaci = 0
  While (lookahead <> "") And Not (finarchivo)
    If RespuestaReactivook(lookahead) = 1 Then
      numaci = numaci + 1
    End If
    Match (lookahead)
    If lookahead = "," Then
      Match (",")
    End If
  Wend
  If finarchivo Then
    MError ("Fin de archivo inesperado, espera ")
  Else
    Match ("")
    operador = lookahead
    If atrib <> 255 Then
      MError ("Espera operando lógico (=,>, etc) ")
    End If
  End If

```

```

Else
  Match (lookahead)
  If atrib <> 1 Then
    MError ("Espera un número entero ")
  Else
    TablaR(Xtabla, ytabla).nombre_prop = "aciertos"
    TablaR(Xtabla, ytabla).ul_valor_verdad = 0
    TablaR(Xtabla, ytabla).Visitado = 0
    TablaR(Xtabla, ytabla).concluido = 0
    TablaR(Xtabla, ytabla).Tipo = 0
    If Cumpleaciertos(operador, lookahead, numaci) Then
      TablaR(Xtabla, ytabla).valor_verdad = 100
    Else
      TablaR(Xtabla, ytabla).valor_verdad = -100
    End If
    Match (lookahead)
  End If
End If
End If
ElseIf atrib = 100 Then ' if no es reactivo, es una var. alfanumérica
  negacion = 0
  If lookahead = "no" Then
    Match ("no")
    negacion = -1
  End If
  TablaR(Xtabla, ytabla).nombre_prop = lookahead
  TablaR(Xtabla, ytabla).valor_verdad = 0
  TablaR(Xtabla, ytabla).ul_valor_verdad = 0
  TablaR(Xtabla, ytabla).Visitado = 0
  TablaR(Xtabla, ytabla).concluido = 0
  If CualPregunta(TablaR(Xtabla, ytabla).nombre_prop, preg) > 0 Then
    TablaR(Xtabla, ytabla).Tipo = 2
  Else
    TablaR(Xtabla, ytabla).Tipo = 1
  End If
  TablaR(Xtabla, ytabla).negacion = negacion
  Match (lookahead)
  Else MError ("Espera nombre de una proposición")
  End If
End Sub

Sub ParseaProposicion ()
Dim cicloand As Integer
Dim strecomen As String
Xtabla = 0
ytabla = ytabla + 1
Match ("si")
cicloand = 1
If lookahead = "entonces" Then
  MError ("Espera una proposición")
Else
  While (lookahead <> "entonces") And Not (finarchivo) And Not (e_error)
    Xtabla = Xtabla + 1
    ParseaPremisa
    If Not (e_error) Then
      If (lookahead = "y") And ((cicloand = 1) Or (cicloand)) Then
        cicloand = -1

```

```

    Match ("y")
    ElseIf (lookahead = "o") And ((cicloand = 1) Or Not (cicloand)) Then
        cicloand = 0
        Match ("o")
    ElseIf ((lookahead = "y") And Not (cicloand)) Or ((lookahead = "o") And (cicloand)) Then
        MError ("No puede haber conjunciones y disyunciones en una prop.")
    ElseIf lookahead <> "entonces" Then
        MError ("Espera entonces, o, o y")
    End If
End If
Wend
Match ("entonces")
If (lookahead = "recomendación") Or (lookahead = "recomendacion") Then
    huborecomendaciones = True
    Match (lookahead)
    Match (""")
    While (lookahead <> """) And Not (finarchivo) And Not (e_error)
        strecomen = strecomen + lookahead + " "
        Match (lookahead)
    Wend
    If (finarchivo) And (lookahead <> """) Then
        MError ("Error, no encontró """)
    Else
        TablaC(ytabla).nombre_prop = strecomen
        TablaC(ytabla).valasig = 100
        TablaC(ytabla).valor_verdad = 0
        TablaC(ytabla).ul_valor_verdad = 0
        TablaC(ytabla).concluido = 0
        TablaC(ytabla).Visitado = 0
        TablaC(ytabla).conjuncion = cicloand
        TablaC(ytabla).TipoConc = 0 'recomendación
        Match (""")
    End If
Else ' si es una proposición
    If atrib = 100 Then
        TablaC(ytabla).negacion = 0
        If lookahead = "no" Then
            Match (lookahead)
            TablaC(ytabla).negacion = -1
        End If
        TablaC(ytabla).nombre_prop = lookahead
        TablaC(ytabla).valasig = 100
        TablaC(ytabla).valor_verdad = 0
        TablaC(ytabla).ul_valor_verdad = 0
        TablaC(ytabla).Visitado = 0
        TablaC(ytabla).concluido = 0
        TablaC(ytabla).conjuncion = cicloand
        TablaC(ytabla).TipoConc = 1 ' proposición
        Match (lookahead)
    Else
        MError ("Error, espera un nombre de proposición")
    End If
End If
If Not e_error Then
    If lookahead = "(" Then
        Match "("
        If atrib <> 1 Then 'si el lookahead no es numérico

```

```

    MError ("Espera un número")
Else
    TablaC(ytabla).valasig = CInt(lookahead)
    Match (lookahead)
    If lookahead = "%" Then
        Match ("%")
    End If
    Match ("")
End If
End If
End If
End Sub

Sub PonConcluidos (nomprop As String)
Dim i As Integer
For i = 1 To ytabla
    If (TablaC(i).nombre_prop = nomprop) Then
        TablaC(i).concluido = True
    End If
Next i
End Sub

Function Preguntar (nombre_prop As String) As Integer
Dim s, i As Integer
Dim preg As String
i = CualPregunta(nombre_prop, preg)
If (i > 0) And (TablaP(i).valor_verdad <> 0) Then
    Preguntar = TablaP(i).valor_verdad
ElseIf i > 0 Then
    s = MsgBox(preg, 4)
    If s = 6 Then
        TablaP(i).valor_verdad = 100
        Preguntar = 100
    ElseIf s = 7 Then
        TablaP(i).valor_verdad = -100
        Preguntar = -100
    End If
Else Preguntar = 0
End If
End Function

Sub RazonaForward ()
Dim ni As Integer
Dim i As Integer
calculaVv = False

While Not calculaVv

    cambia = False
    i = 0
    'concluye = False
    While (i < ytabla) 'And Not (concluye)
        i = i + 1
        b = True
        If Not TablaC(i).Visitado Then
            If TablaC(i).conjuncion Then

```

```

CicloAndF (i) ' recorre las proposiciones en and y va calculando el
'valor de certeza menor para deducir la proposici n
Else
CicloOrF (i) ' recorre las proposiciones en or y va calculando el
' valor de certeza menor para deducir la proposici n
End If
valorv = v1
v2 = v1
If b Then
    TablaC(i).Visitado = True
    TablaC(i).valor_verdad = valorv
    TablaC(i).ul_valor_verdad = valorv
    While (ExisteSigProp(TablaC(i).nombre_prop, ni)) And (b)
    ' ciclo para que busque todas las premisas que concluyen la
    ' proposici n que se est analizando. Si existe, calcula
    ' el valor de verdad de la nueva premisa y los combina
    If TablaC(ni).conjuncion Then
        CicloAndF (ni)
    Else
        CicloOrF (ni)
    End If
    If b Then
        TablaC(ni).valor_verdad = v1
        valorv = Combina(v1, v2, TablaC(ni).nombre_prop)
        TablaC(ni).ul_valor_verdad = valorv
        TablaC(ni).Visitado = True
    End If
    v2 = valorv
    Wend
End If
If b Then
    'si logr  concluir sobre esta proposici n }
    cambia = True
    PonConcluidos (TablaC(i).nombre_prop)
    TablaC(i).ul_valor_verdad = valorv
    TablaC(i).Visitado = True
Else
    TablaC(i).Visitado = False
End If
'If TablaC(i).Nombre_prop = Nombre_prop And TablaC(i).concluido Then
' concluye = True
'End If

End If
Wend
If (Not (cambia)) Then 'Or (concluye) Then
    calculaVv = True
End If
Wend
End Sub

Sub razonamiento ()
    If Not (e_error) Then
        RazonaForward
        Darecomendaciones
    Else
        MsgBox ("No se ha compilado, o hay error")
    End If
End Sub

```

```
End If
End Sub
```

```
Function RespuestaReactivook (idreact As String) As Integer
Dim s, acierto As Integer
If ExisteReactivo(idreact, acierto) Then
    RespuestaReactivook = acierto
Else
    indarreglo = indarreglo + 1
    Arreglo(indarreglo).idreactivo = idreact
    s = MsgBox("Es correcto el reactivo " + idreact + " ?", 3)
    If s = 6 Then
        Arreglo(indarreglo).acierto = 1
    ElseIf s = 7 Then
        Arreglo(indarreglo).acierto = -1
    Else
        Arreglo(indarreglo).acierto = 0
    End If
    RespuestaReactivook = Arreglo(indarreglo).acierto
End If
End Function
```

```
Function Token () As String
Dim T As String
Dim errg As Integer
Dim R As String
T = ""
While (Mid$(Entrada, indicearchivo, 1) = " ") Or (Mid$(Entrada, indicearchivo, 1) = Chr(13)) Or (Mid$(Entrada,
indicearchivo, 1) = Chr(10))
    indicearchivo = indicearchivo + 1
Wend
finarchivo = indicearchivo > Longarchivo
If Not finarchivo Then
    R = Leecaracter()
    Select Case R
        Case "a" To "z", "á", "é", "í", "ó", "ú", "Á", "É", "Í", "Ó", "Ú"
            While Not (finarchivo) And (((R >= "a") And (R <= "z")) Or (R = "_") Or (R = "á") Or (R = "é") Or (R = "í") Or
(R = "ó") Or (R = "ú") Or ((R >= "0") And (R <= "9")))
                T = T + R
                R = Leecaracter()
            Wend
            indicearchivo = indicearchivo - 1
            atrib = 100
            Case "0" To "9", "."
                atrib = 1
                errg = False
                While Not (finarchivo) And (((R >= "0") And (R <= "9")) Or (R = ".")) And Not (errg)
                    T = T + R
                    errg = (atrib = 2) And (R = ".")
                    If R = "." Then
                        atrib = 2
                    End If
                    If Not errg Then
                        R = Leecaracter()
                    End If
                Wend
                indicearchivo = indicearchivo - 1
```

```

Case "="
  T = T + R
  atrib = 255
Case ">"
  T = T + R
  R = Leecaracter()
  If R = "=" Then
    T = T + R
  Else
    indicearchivo = indicearchivo - 1
  End If
  atrib = 255
Case "<"
  T = T + R
  R = Leecaracter()
  If (R = ">") Or (R = "=") Then
    T = T + R
  Else
    indicearchivo = indicearchivo - 1
  End If
  atrib = 255
Case Else
  T = T + R
  atrib = -100
End Select
End If
Token = T
End Function

```

## C2. SISEVAL: Programa evaluador.

Este programa consta de 3 formas y de 2 programas de código:

```

Fprin.frm
Fpregun.frm
Ffin2.frm
archivos.bas
mrazona.bas

```

### C2.1. Fprin.frm.

Option Explicit

```

Sub Combo1_LostFocus ()
Dim Periodo, TipPer As String
Dim Sigue As Integer

While Combo2.ListCount > 0
  Combo2.RemoveItem 0
Wend
Periodo = Mid$(Combo1.Text, 1, 3)
TipPer = Mid$(Combo1.Text, 4, 1)
sesgrupo.Index = "Llave0"
sesgrupo.Seek ">=", Periodo, TipPer, "00000000", "00"
Sigue = 1
If Not sesgrupo.NoMatch Then

```

```

While (Not sesgrupo.EOF) And (Sigue)
  Combo2.AddItem sesgrupo(2) + sesgrupo(3)
  sesgrupo.MoveNext
  If Not sesgrupo.EOF Then
    Sigue = (Left$(Periodo, 3) = Left$(sesgrupo(0), 3)) And (Left$(TipPer, 1) = Left$(sesgrupo(1), 1))
  End If
Wend
End If
End Sub

Sub Combo2_LostFocus ()
Dim Periodo, TipPer, gpo, Clavemat As String
Dim Sigue As Integer
While Combo3.ListCount > 0
  Combo3.RemoveItem 0
Wend

Clavemat = Mid$(Combo2.Text, 1, 8)
gpo = Mid$(Combo2.Text, 9, 2)
Periodo = Mid$(Combo1.Text, 1, 3)
TipPer = Mid$(Combo1.Text, 4, 1)
Evaluacion.Index = "Llave0"
'Evaluacion.MoveFirst
Evaluacion.Seek ">=", Periodo, TipPer, Clavemat, gpo, ""
Sigue = 1
If Not Evaluacion.NoMatch Then
While Not Evaluacion.EOF And Sigue
  Combo3.AddItem Evaluacion(4)
  Evaluacion.MoveNext
  If Not Evaluacion.EOF Then
    Sigue = Periodo = Evaluacion(0) And TipPer = Evaluacion(1) And Clavemat = Evaluacion(2) And gpo =
Evaluacion(3)
  End If
Wend
End If
End Sub

Sub Command1_Click ()
IdEvaluacion = Combo3.Text
Cvegrupo = Combo2.Text
Periodo = Mid$(Combo1.Text, 1, 3)
TipPer = Mid$(Combo1.Text, 4, 1)
If ExisteEvaluacion(IdEvaluacion) Then

  Form2.Show ' Forma que muestra los reactivos de
' la evaluación
End If
End Sub

Sub Command2_Click ()
CierraArchivos
End
End Sub

Sub Command3D1_Click ()
IdEvaluacion = Combo3.Text
Cvegrupo = Combo2.Text

```

```

Periodo = Mid$(Combo1.Text, 1, 3)
TipPer = Mid$(Combo1.Text, 4, 1)
If ExisteEvaluacion(IdEvaluacion) Then

    Form2.Show    ' Forma que muestra los reactivos de
                  ' la evaluación
End If

End Sub

Sub Command3D2_Click ()
CierraArchivos
End
End Sub

Sub Form_Activate ()
Text1.SetFocus
End Sub

Sub Form_Load ()

Combo1.AddItem "963S Semestre Ago-Dic 96"
Combo1.AddItem "961S Semestre Ene-May 96"
Combo1.AddItem "953S Semestre Ago-Dic 95"
Combo1.AddItem "961T Trimestre Ene-Abr 96"
ABREARCHIVOS
seg = 0
min = 0
hrs = 0
End Sub

Sub Text1_KeyPress (keyAscii As Integer)
If keyAscii = Asc(Chr(13)) Then
If Text1.Text <> "" Then
SESALUM.Index = "LLAVE1"
SESALUM.Seek "=", Text1
If SESALUM.NoMatch Then
MsgBox "NO EXISTE LA MATRICULA "
Text1.SetFocus
Else
LABEL3.Caption = SESALUM(5)
End If
End If
End If
End Sub

Sub Text1_LostFocus ()
If Text1.Text <> "" Then
SESALUM.Index = "LLAVE1"
SESALUM.Seek "=", Text1
If SESALUM.NoMatch Then
MsgBox "NO EXISTE LA MATRICULA "
Text1.SetFocus
Else
LABEL3.Caption = SESALUM(5)
End If
End If
End Sub

```

End Sub

## C2.2. Fpregun.frm.

Option Explicit

'Dim acierto, Pond, naciertos As Integer

Sub Command1\_Click ()

Dim T As String

T = Text1.Text

If T <> "" Then

    AciertoPregunta Asc(T) - 64, T, 1

End If

End Sub

Sub Command2\_Click (Index As Integer)

    AciertoPregunta Index + 1, " ", 1

End Sub

Sub Command3\_Click (Index As Integer)

    InformacionBotones (Index)

End Sub

Sub Command3D1\_Click (Index As Integer)

Dim T As String

Dim nasc As Integer

If Index = 2 Then

    Finaliza

Else

    T = Text1.Text

    If T <> "" Then

        nasc = Asc(T)

    Else

        nasc = 0

    End If

    AciertoPregunta nasc - 64, T, (Index = 1)

End If

End Sub

Sub Form\_Load ()

seg = 0

min = 0

hrs = 0

totalseg = 0

If Not navegable Then

    Form2.Command3D1(0).Visible = False

    Form2.Command3D1(1).Visible = False

    Form2.Command3D1(2).Visible = False

Else

    Form2.Command3D1(0).Visible = True

    Form2.Command3D1(1).Visible = True

    Form2.Command3D1(2).Visible = True

End If

PonEncabezadoPregunta

Horainicio = Now

CargaPreguntaUno

```
'ABREARCHIVOS
'Resp = REACTIVOS(13)
'REACTIVOS.INDEX = "LLAVE1"
End Sub
```

```
Sub Text1_Change ()
Dim T As String
If Len(Text1.Text) >= 1 Then
    Text1.Text = Mid$(Text1.Text, 1, 1)
End If
If tipo_resp = 2 Then
    T = Text1.Text
    If Len(T) > 0 Then
        If Not (Left(T, 1) = "A" Or Left(T, 1) = "B" Or Left(T, 1) = "C" Or Left(T, 1) = "D" Or Left(T, 1) = "E") Then
            Text1.Text = ""
            Text1.SetFocus
        End If
    End If
End If
End Sub
```

```
Sub Timer1_Timer ()
    totalseg = totalseg + 1
    If seg = 59 Then
        seg = 0
        If min = 59 Then
            min = 0
            hrs = hrs + 1
        Else
            min = min + 1
        End If
    Else
        seg = seg + 1
    End If
    Label12.Caption = Format$(hrs) + ":" + Format$(min) + ":" + Format$(seg)
    If totalseg = TiempoMax Then
        MsgBox (" Disculpa, pero tu tiempo se agotó")
        Finaliza
    End If
End Sub
```

### **C2.3. Ffin2.frm**

No tiene código.

### **C2.4. Archivos.bas**

```
Option Explicit
Global BDSiseval As DATABASE
Global sesmater As Table
Global PANALITICO As Table
Global Sesalum As Table
Global Reactivos As Table
Global reglas As Table
Global evaluacion As Table
Global evareact As Table
```

```

Global AUTOR As Table
Global Bitacora As Table
Global DetBit As Table
Global Sesgrupo As Table
Global SESALGPO As Table
Global Referencias As Table
Global Const maxreactivos = 500
Global Const maxpreg = 100
Type Tarreglo
idreactivo As String * 10 ' identificador del reactivo
acierto As Integer 'si acertó el reactivo: 1=correcto, 0=no contestó, -1=incorrecto
evaluacion As String * 10 ' identificador de la evaluación
Pond As Single ' ponderación en % del reactivo en la evaluación
respuesta As String ' respuesta que eligió, si es "" no respondió
n1 As Integer ' opción (de 0 a 4) correcta en caso de que sea de opción múltiple
End Type
Type Tpreg
nombre_prop As String
preg As String
valor_verdad As Integer
End Type

Global TablaP(1 To maxpreg) As Tpreg
Global Idevaluacion As String * 10
Global npreg, naciertos, nfallas As Integer
Global arreglo(1 To maxreactivos) As Tarreglo
Global idreactivo As String * 10
Global NRespuesta As Integer
Global Ponderacion As Single
Global calif As Single
Global cvegrupo As String * 10
Global Periodo As String * 3
Global TipPer As String * 1
Global Ayuda, Idreferencia, Tutorial As String
Global tipo_resp As Integer
Global navegable As Integer
Global TiempoMax As Long
Global HoraInicio, HoraFin As Variant
Global totpreg As Integer
Global Seg, min, hrs As Integer
Global Totalseg As Long

Sub ABREARCHIVOS ()
Set BDSiseval = OpenDatabase("c:\vb\siseval\SISEVAL.MDB")
Set sesmater = BDSiseval.OpenTable("SESMATER")
Set Sesgrupo = BDSiseval.OpenTable("SESGRUPO")
Set PANALITICO = BDSiseval.OpenTable("PANALITICO")
Set Sesalum = BDSiseval.OpenTable("SESALUM")
Set Reactivos = BDSiseval.OpenTable("REACTIVOS")
Set evareact = BDSiseval.OpenTable("EVAL-REAC")
Set reglas = BDSiseval.OpenTable("REGLAS")
Set evaluacion = BDSiseval.OpenTable("EVALUACION")
Set AUTOR = BDSiseval.OpenTable("AUTOR")
Set Bitacora = BDSiseval.OpenTable("BITACORA")
Set DetBit = BDSiseval.OpenTable("DETBIT")
Set Referencias = BDSiseval.OpenTable("REFERENCIAS")
End Sub

```

```

Sub AciertoPregunta (Resp As Integer, Tresp As String, adelante As Integer)
Dim acierto As Integer

'Nopcion = Numopcion(Text1.Text)
If Tresp <> "" Then
    acierto = -1
    If tipo_resp = 2 Then
        If (Tresp = " ") Then
            form2.Text1.Text = Chr$(Resp + 64)
        End If
        If Resp = NRespuesta Then
            acierto = 1
        End If
    ElseIf tipo_resp = 3 Then
        If Tresp = Reactivos(7) Then
            acierto = 1
        End If
    ElseIf tipo_resp = 4 Then
        If Tresp = " " Then
            Select Case Resp
            Case 2
                form2.Text1.Text = "V"
            Case 3
                form2.Text1.Text = "F"
            End Select
        End If
        If (Resp = 2 And Reactivos(14)) Or (Resp = 3 And Not (Reactivos(14))) Then
            acierto = 1
        End If
    End If
    If acierto = 1 And evareact(6) Then
        MsgBox (Reactivos(20))
    ElseIf acierto = -1 And evareact(7) Then
        MsgBox (Reactivos(21))
    End If
    End If ' if contestó algo (if tresp<> "")
    arreglo(npreg).respuesta = form2.Text1.Text
    arreglo(npreg).idreactivo = idreactivo
    arreglo(npreg).acierto = acierto
    arreglo(npreg).evaluacion = Ievaluacion
    arreglo(npreg).Pond = Ponderacion
    If npreg = totpreg And adelante = 1 Then
        If MsgBox("Son todas las preguntas, terminas? ", 3) = 6 Then
            Finaliza
        End If
    ElseIf npreg = totpreg And adelante = -1 Then
        Finaliza
    Else
        If adelante = -1 Or adelante = 1 Then
            npreg = npreg + 1
            evareact.MoveNext
        Else
            npreg = npreg - 1
            evareact.MovePrevious
        End If
        idreactivo = evareact(2)
    End If
End Sub

```

```

Ponderacion = evareact(5)
Screen.ActiveForm.Text1.Text = ""
Screen.ActiveForm.Text1.SetFocus
PreguntaReactivo idreactivo
End If

```

```

End Sub

```

```

Sub Actualiza_Tabla (Nombre As Table)
Dim i As Integer
Dim Texto As String
Nombre(0) = Screen.ActiveForm.Text1
Texto = ""
Nombre(1) = Texto
Nombre(2) = Screen.ActiveForm.Text1
Nombre(3) = Screen.ActiveForm.Text1
Nombre.Update
End Sub

```

```

Function aleatorio (ByVal LI As Integer, ByVal LS As Integer) As Integer
aleatorio = Int((LS - LI + 1) * Rnd + LI)
End Function

```

```

Sub CargaPreguntaUno ()
Dim i As Integer
naciertos = 0
calif = 0
evareact.Index = "Llave0"
npreg = 0
evareact.Seek "=", Idevaluacion
If evareact(0) = Idevaluacion Then
If evaluacion(5) > 0 Then
npreg = npreg + 1
idreactivo = evareact(2)
Ponderacion = evareact(5)
PreguntaReactivo idreactivo
End If
End If
End Sub

```

```

Sub CierraArchivos ()
sesmater.Close
Sesgrupo.Close
PANALITICO.Close
Sesalum.Close
Reactivos.Close
evareact.Close
reglas.Close
evaluacion.Close
AUTOR.Close
Bitacora.Close
DetBit.Close
Referencias.Close
BDSiseval.Close
End Sub

```

```

Function ExisteEvaluacion (Idevaluacion As String) As Integer

```

```

Dim i As Integer
evaluacion.Index = "Llave1"
evaluacion.Seek "=", Ievaluacion
If evaluacion.NoMatch Then
    MsgBox "No existe la Evaluación"
    ExisteEvaluacion = False
Else
    ExisteEvaluacion = True
    totpreg = evaluacion(5)
    navegable = (evaluacion(11) = "S") Or (evaluacion(11) = "s")
    TiempoMax = evaluacion(10)
    For i = 1 To maxreactivos
        arreglo(i).idreactivo = ""
        arreglo(i).respuesta = ""
        arreglo(i).acierto = 0
    Next i
End If
End Function

```

```

Sub Finaliza ()
Dim scalif As String
Dim i As Integer
calif = 0
For i = 1 To totpreg
    If arreglo(i).acierto = 1 Then
        naciertos = naciertos + 1
        calif = calif + arreglo(i).Pond / 10
    ElseIf arreglo(i).acierto = -1 Then
        nfallas = nfallas + 1
    End If
Next i

    scalif = Str$(calif)
    HoraFin = Now
    Bitacora.AddNew
    Bitacora(0) = Periodo
    Bitacora(1) = TipPer
    Bitacora(2) = Sesalum(2)
    Bitacora(3) = Ievaluacion
    Bitacora(4) = naciertos
    Bitacora(5) = nfallas
    Bitacora(6) = calif
    Bitacora(7) = CInt(HoraFin - HoraInicio)
    Bitacora(9) = Now
    Bitacora.Update
    RegistraDetalleBitacora

    MsgBox ("Tu calificación fue " + scalif)
    AplicaReglas 'aplica las reglas, escanea y razona
                ' para dar recomendaciones
End Sub

```

```

Sub InformacionBotones (Index As Integer)
Dim st As String
Dim X
Select Case Index
Case 1

```

```

If Ayuda <> "" Then
    MsgBox (Ayuda)

End If
Case 2
If Tutorial <> "" Then
    X = Shell(Tutorial)

End If
Case 3
If Idreferencia <> "" Then
    Referencias.Index = "Llave0"
    Referencias.Seek "=", Idreferencia
    If evaluacion.NoMatch Then
        MsgBox "No existe la referencia especificada"
    Else
        st = Referencias(1) + Referencias(2)
        If st <> "" Then
            MsgBox (st)

        End If
    End If
End If
End Select

End Sub

Sub PonEncabezadoPregunta ()
Dim cvmateria As String * 8
Dim grupo As String * 2
sesmater.Index = "Llave0"
cvmateria = Mid$(cvegrupo, 1, 8)
sesmater.Seek "=", cvmateria
form2.Label7.Caption = Idevaluacion
If sesmater.NoMatch Then
    form2.Label8.Caption = "No registrada"
Else
    form2.Label8.Caption = sesmater(1)
End If
grupo = Mid$(cvegrupo, 9, 2)
Sesgrupo.Index = "Llave0"
Sesgrupo.Seek "=", Periodo, TipPer, cvmateria, grupo
If Sesgrupo.NoMatch Then
    form2.Label9.Caption = "No existe gpo"
Else
    form2.Label9.Caption = Sesgrupo(5)
End If
form2.Label11.Caption = Format$(totpreg)
If TiempoMax = 0 Then
    form2.Label14.Caption = "Sin Límite"
Else
    form2.Label14.Caption = Format$(CInt(TiempoMax / 3600)) + ":" + Format$(CInt((TiempoMax Mod 3600) / 60))
+ ":" + Format$(CInt(TiempoMax Mod 60))
End If
End Sub

Sub PreguntaReactivo (idreactivo As String)

```

```

Dim n1, i As Integer
Dim j As Integer

If navegable Then
If (npreg = 1) Then
    form2.Command3D1(0).Enabled = False
    form2.Command3D1(1).Enabled = True
ElseIf (npreg = totpreg) Then
    form2.Command3D1(1).Enabled = False
    form2.Command3D1(0).Enabled = True
Else
    form2.Command3D1(0).Enabled = True
    form2.Command3D1(1).Enabled = True
End If
End If
Reactivos.Index = "Llave0"
Reactivos.Seek "=", idreactivo
If Reactivos.NoMatch Then
    MsgBox "No existe el Reactivo"
Else
On Error Resume Next
Ayuda = Reactivos(8)
If Err Then Ayuda = ""
On Error Resume Next
Tutorial = Reactivos(9)
If Err Then Tutorial = ""
On Error Resume Next
Idreferencia = Reactivos(23)
If Err Then Idreferencia = ""
tipo_resp = Reactivos(6)
If Reactivos(11) <> "" Then
    On Error Resume Next
    form2.Image1.Picture = LoadPicture("c:\windows\" + Reactivos(11))
    If Err Then
        MsgBox ("No existe el gráfico c:\windows\" + Reactivos(11) + " asociado a este reactivo")
        form2.Image1.Picture = LoadPicture()
    End If
Else
    form2.Image1.Picture = LoadPicture()
End If
If arreglo(npreg).respuesta = "" Then
    n1 = aleatorio(0, 4)
    arreglo(npreg).n1 = n1
Else
    n1 = arreglo(npreg).n1
End If
NRespuesta = n1 + 1
If Reactivos(5) <> "" Then
    form2.Label1.Caption = CStr(npreg) + ".- " + Reactivos(5)
Else
    form2.Label1.Caption = CStr(npreg) + ".- ?"
End If
If tipo_resp = 2 Then
    form2.Command2(1).Caption = "B)"
    form2.Command2(2).Caption = "C)"
    form2.Label5.Caption = "Elige Opción:"
For i = 0 To 4

```

```

form2.Label4(i).Visible = True
form2.Command2(i).Visible = True
Next i
If Reactivos(15) <> "" Then
form2.Label4(n1).Caption = Reactivos(15)
Else
form2.Label4(n1).Caption = ""
End If
i = 0
j = 0
While i < 5
If i <> n1 Then
If Reactivos(j + 16) <> "" Then
form2.Label4(i).Caption = Reactivos(j + 16)
Else
form2.Label4(i).Caption = ""
End If
j = j + 1
End If
i = i + 1
Wend
ElseIf tipo_resp = 3 Then
form2.Label5.Caption = "Respuesta:"
For i = 0 To 4
form2.Label4(i).Visible = False
form2.Command2(i).Visible = False
Next i
ElseIf tipo_resp = 4 Then
form2.Command2(1).Caption = "V"
form2.Command2(2).Caption = "F"
form2.Label5.Caption = "Falso/Verdadero:"
form2.Label4(0).Visible = False
form2.Command2(0).Visible = False
For i = 1 To 2
form2.Label4(i).Visible = True
form2.Command2(i).Visible = True
Next i
For i = 3 To 4
form2.Label4(i).Visible = False
form2.Command2(i).Visible = False
Next i
form2.Label4(1).Caption = "Verdadero"
form2.Label4(2).Caption = "Falso"
End If
If arreglo(npreg).respuesta <> "" Then
form2.Text1.Text = arreglo(npreg).respuesta
End If
End If
End Sub

```

```

Sub RegistraDetalleBitacora ()
Dim i As Integer
For i = 1 To totpreg
DetBit.AddNew
DetBit(0) = Periodo
DetBit(1) = TipPer
DetBit(2) = Sesalum(2)

```

```

DetBit(3) = arreglo(i).evaluacion
DetBit(4) = arreglo(i).idreactivo
If arreglo(i).acierto = 1 Then
  DetBit(5) = "S"
ElseIf arreglo(i).acierto = -1 Then DetBit(5) = "N"
Else DetBit(5) = " "
End If
DetBit(6) = arreglo(i).Pond
DetBit(7) = Now
DetBit.Update
Next i
End Sub

```

## C2.5. Mrazona.bas

Option Explicit

```

Const Maxreglas = 500
Const Maxpremisas = 30
Dim indicearchivo, Longarchivo As Integer
Dim finarchivo, contlineas, atrib As Integer
Type Treglas
  nombre_prop As String
  valor_verdad As Integer 'valor de verdad en esta proposición
  ul_valor_verdad As Integer 'valor de verdad combinado, en caso de aplicar
  Visitado As Integer
  concluido As Integer
  negacion As Integer
  Tipo As Integer '0= Hecho (si reactivo ok),
                 '1= Proposición
End Type
Type Creglas
  nombre_prop As String
  valor_verdad As Integer
  ul_valor_verdad As Integer 'valor de verdad combinado, en caso de aplicar
  Visitado As Integer
  concluido As Integer
  conjuncion As Integer
  TipoConc As Integer '0=Recomendación
                    '1=Proposición
  valasig As Integer
  negacion As Integer
End Type
Dim TablaR(1 To Maxpremisas, 1 To Maxreglas) As Treglas
Dim TablaC(1 To Maxreglas) As Creglas
Dim Xtabla, ytabla As Integer
Dim lookahead As String
Dim e_error As Integer
Dim huborecomendaciones As Integer
Dim Entrada As Variant

Dim e, concluye, calculaVv, b, cambia As Integer
Dim v1, v2 As Integer
Dim valorv As Integer
Global indpregunta As Integer

```

```

Sub AplicaReglas ()
Dim i, j As Integer
reglas.Index = "Llave0"
reglas.Seek "=", Ievaluacion
If Not reglas.NoMatch Then
    For i = 1 To Maxreglas
        TablaC(i).nombre_prop = "nil"
        For j = 1 To Maxpremisas
            TablaR(j, i).nombre_prop = "nil"
        Next j
    Next i
    indpregunta = 0
    For i = 1 To Maxpreg
        TablaP(i).nombre_prop = ""
        TablaP(i).preg = ""
        TablaP(i).valor_verdad = 0
    Next i
    indicearchivo = 1
    Entrada = reglas(1)
    Longarchivo = Len(Entrada)
    lookahead = Token()
    ytabla = 0
    contlineas = 1
    huborecomendaciones = False
    EscaneaReglas
    If Not (e_error) Then
        RazonaForward
        Darecomendaciones
    End If
End If

End Sub

Function BuscaConclu (nombreprop As String) As Integer
Dim i, encuentra As Integer
encuentra = False
i = 0
While (i < ytabla) And Not (encuentra)
    i = i + 1
    encuentra = (TablaC(i).nombre_prop = nombreprop)
Wend
If encuentra Then
    BuscaConclu = i
Else
    BuscaConclu = 0
End If
End Function

Sub CicloAndF (indicey As Integer)
Dim iconclu, minvalorv, sigue, x, vv As Integer
v1 = 100
sigue = True
x = 1
While (Mid$(TablaR(x, indicey).nombre_prop, 1, 3) <> "nil") And (b) And (sigue)
    If (TablaR(x, indicey).Tipo = 1) Then
        iconclu = BuscaConclu(TablaR(x, indicey).nombre_prop)
    End If
End While

```

```

If iconclu > 0 Then
  If TablaC(iconclu).concluido Then
    If TablaR(x, indicey).negacion Then
      minvalorv = -1 * TablaC(iconclu).ul_valor_verdad
    Else
      minvalorv = TablaC(iconclu).ul_valor_verdad
    End If
    If minvalorv < v1 Then
      v1 = minvalorv
    End If
    If (v1 = -100) Then
      sigue = False
    End If
  Else
    b = False
    v1 = 0
    calculaVv = False
  End If
Else
  MsgBox ("Warning, no se concluyó " + TablaR(x, indicey).nombre_prop)
End If
ElseIf TablaR(x, indicey).Tipo = 2 Then 'si es una pregunta por hacer
  vv = Preguntar(TablaR(x, indicey).nombre_prop)
  If TablaR(x, indicey).negacion Then
    minvalorv = -1 * vv
  Else
    minvalorv = vv
  End If
  TablaR(x, indicey).Visitado = True
  If minvalorv < v1 Then
    v1 = minvalorv
  End If
  If (v1 = -100) Then
    sigue = False
  End If
ElseIf TablaR(x, indicey).Tipo = 0 Then 'si es hecho
  minvalorv = TablaR(x, indicey).valor_verdad ' aquí se puede insertar
  'para hacer preguntas al momento de proposiciones y
  'asignar el valor_verdad dependiendo de la respuesta
  'del usuario
  TablaR(x, indicey).Visitado = True
  If minvalorv < v1 Then
    v1 = minvalorv
  End If
  If (v1 = -100) Then
    sigue = False
  End If
Else
  b = False
  v1 = 0
  calculaVv = False
End If
x = x + 1
Wend
If (TablaC(indicey).valasig <> 100) And (v1 > 0) Then
  v1 = CInt(v1 * TablaC(indicey).valasig / 100)
End If

```

```

If v1 < 0 Then
    v1 = 0
End If
If TablaC(indicey).negacion Then
    v1 = -1 * v1
End If
End Sub

Sub CicloOrF (indicey As Integer)
Dim iconclu, maxvalorv, sigue, x, vv As Integer
v1 = -100
x = 1
sigue = True
While (Mid$(TablaR(x, indicey).nombre_prop, 1, 3) <> "nil") And (b) And (sigue)
    If (TablaR(x, indicey).Tipo = 1) Then
        iconclu = BuscaConclu(TablaR(x, indicey).nombre_prop)
        If iconclu > 0 Then
            If TablaC(iconclu).concluido Then
                If TablaR(x, indicey).negacion Then
                    maxvalorv = -1 * TablaC(iconclu).ul_valor_verdad
                Else
                    maxvalorv = TablaC(iconclu).ul_valor_verdad
                End If
            If maxvalorv > v1 Then
                v1 = maxvalorv
            End If
            If (v1 = 100) Then
                sigue = False
            End If
        Else
            b = False
            v1 = 0
            calculaVv = False
        End If
    Else
        MsgBox ("Warning, no se concluyó " + TablaR(x, indicey).nombre_prop)
    End If
ElseIf TablaR(x, indicey).Tipo = 2 Then 'si es una pregunta por hacer
    vv = Preguntar(TablaR(x, indicey).nombre_prop)
    If TablaR(x, indicey).negacion Then
        maxvalorv = -1 * vv
    Else
        maxvalorv = vv
    End If
    TablaR(x, indicey).Visitado = True
    If maxvalorv > v1 Then
        v1 = maxvalorv
    End If
    If (v1 = 100) Then
        sigue = False
    End If
ElseIf (TablaR(x, indicey).Tipo = 0) Then
    maxvalorv = TablaR(x, indicey).valor_verdad
    TablaR(x, indicey).Visitado = True
    If maxvalorv > v1 Then
        v1 = maxvalorv
    End If

```

```

    If (v1 = 100) Then
        sigue = False
    End If
Else
    b = False
    v1 = 0
    calculaVv = False
End If
x = x + 1
Wend
If (TablaC(indicey).valasig <> 100) And (v1 > 0) Then
    v1 = CInt(v1 * TablaC(indicey).valasig / 100)
End If
If v1 < 0 Then
    v1 = 0
End If
If TablaC(indicey).negacion Then
    v1 = -1 * v1
End If
End Sub

Function Combina (ByVal val1, val2 As Integer, nombre_prop As String) As Integer
' combina valores de certeza ya que existen 2 premisas con
' la misma conclusiøn
Dim minimo As Integer

If (val1 >= 0) And (val2 >= 0) Then '{ 2 factores positivos }
    val1 = val1 + CInt((val2 * (100 - val1)) / 100)
ElseIf (val1 < 0) And (val2 < 0) Then '{ 2 factores negativos }
    val1 = -(Abs(val1) + ((Abs(val2) * CInt(100 - Abs(val1))) / 100))
Else '{ 1 negativo y otro positivo }
    minimo = Abs(val1)
    If Abs(val2) < minimo Then
        minimo = Abs(val2)
    End If
    If minimo = 100 Then
        MsgBox ("Hay una completa contradicción al concluir " + nombre_prop)
    End If
    val1 = CInt((val1 + val2) * 100) / (100 - minimo)
End If
Combina = val1
End Function

Function CualPregunta (nombre_prop As String, preg As String)
Dim i As Integer
i = 0
Do
    i = i + 1
Loop Until (i > indpregunta) Or (TablaP(i).nombre_prop = nombre_prop)
If (i <= indpregunta) Then
    CualPregunta = i
Else CualPregunta = 0
End If
If i <= indpregunta Then
    preg = TablaP(i).preg
Else preg = ""
End If

```

End Function

```
Function Cumpleaciertos (operador, valor As String, numaci As Integer) As Integer
Dim cvalor As Integer
cvalor = CInt(valor)
Select Case operador
Case ">"
    Cumpleaciertos = (numaci > cvalor)
Case ">="
    Cumpleaciertos = (numaci >= cvalor)
Case "<"
    Cumpleaciertos = (numaci < cvalor)
Case "<="
    Cumpleaciertos = (numaci <= cvalor)
Case "="
    Cumpleaciertos = (numaci = cvalor)
Case "<>"
    Cumpleaciertos = (numaci <> cvalor)
End Select
```

End Function

```
Function Cumplecalificacion (operador, valor As String) As Integer
Dim cvalor As Single
cvalor = CDbI(valor)
Select Case operador
Case ">"
    Cumplecalificacion = (calif > cvalor)
Case ">="
    Cumplecalificacion = (calif >= cvalor)
Case "<"
    Cumplecalificacion = (calif < cvalor)
Case "<="
    Cumplecalificacion = (calif <= cvalor)
Case "="
    Cumplecalificacion = (calif = cvalor)
Case "<>"
    Cumplecalificacion = (calif <> cvalor)
End Select
End Function
```

Sub Darecomendaciones ()

Dim i As Integer

Dim strecomen As String

Form3.Show

Form3.SetFocus

strecomen = ""

For i = 1 To ytabla

If (TablaC(i).TipoConc = 0) And (TablaC(i).valor\_verdad >= 20) Then

Form3.List1.AddItem ("(" + Format\$(TablaC(i).valor\_verdad) + ")" + TablaC(i).nombre\_prop)

strecomen = strecomen + Chr(10) + Chr(13) + TablaC(i).nombre\_prop + " (" +

Format\$(TablaC(i).valor\_verdad) + ")"

End If

Next i

If strecomen = "" Then

If huborecomendaciones Then

MsgBox ("No se obtuvieron Recomendaciones")

```

Else
  MsgBox ("No existen reglas con Recomendaciones ")
End If
'Else
'MsgBox ("Recomendaciones: " + Chr(10) + Chr(13) + strecomen)
End If
End Sub

```

```

Sub EscaneaReglas ()
If Not finarchivo Then
Select Case LCase$(lookahead)
Case "si"
  ParseaProposicion
  'If Not e_error Then
  EscaneaReglas
  'End If
Case "pregunta"
  ParseaPregunta
  'If Not e_eror Then
  EscaneaReglas
  'End If
Case Else
  MError ("Error, espera Si o Pregunta ")
End Select
End If
End Sub

```

```

Function ExisteSigProp (nomprop As String, ni As Integer) As Integer
Dim b As Integer
Dim i As Integer

b = False
i = 0
While Not (b) And (i < ytabla)

  i = i + 1
  If (TablaC(i).nombre_prop = nomprop) Then
    If Not TablaC(i).Visitado Then
      ni = i
      b = True
    End If
  End If
Wend
ExisteSigProp = b
End Function

```

```

Function Leecaracter () As String
Dim R As String * 1
Dim R2 As String * 1
R = ""
If indicearchivo <= Longarchivo Then
  R = LCase(Mid$(Entrada, indicearchivo, 1))
End If
indicearchivo = indicearchivo + 1
finarchivo = indicearchivo > Longarchivo
If indicearchivo > 2 Then
  R2 = LCase(Mid$(Entrada, indicearchivo - 2, 1))

```

```

Else R2 = ""
End If
If R2 = Chr(10) Then
    contlineas = contlineas + 1
End If
Leecaracter = R
End Function

Sub Match (actToken As String)
If Not (finarchivo) And Not (e_error) Then
    If (lookahead = actToken) Then
        '{ Despliega el número de líneas compiladas }
        'Form3.Print contlineas
        lookahead = Token()
    Else
        MError ("Espera " + actToken)
    End If
End If
End Sub

Sub MError (st As String)
    MsgBox (st + " en línea " + Format$(contlineas))
    e_error = -1
End Sub

Sub ParseaPregunta ()
Dim stpreg As String
Match ("pregunta")
If atrib <> 100 Then
    MError ("Espera una proposición")
ElseIf CualPregunta(lookahead, stpreg) Then
    MsgBox ("Ya está definida la pregunta para la proposición " + lookahead)
Else
    indpregunta = indpregunta + 1
    TablaP(indpregunta).nombre_prop = lookahead
    Match (lookahead)
    Match (""")
    stpreg = ""
    While (lookahead <> """) And Not (finarchivo)
        stpreg = stpreg + lookahead
        Match (lookahead)
    Wend
    TablaP(indpregunta).preg = stpreg
    Match (""")
End If

End Sub

Sub ParseaPremisa ()
Dim negacion, escorrecto, numaci As Integer
Dim operador, preg As String
If lookahead = "reactivo" Then
    Match ("reactivo")
    If atrib = 1 Or atrib = 100 Then
        TablaR(Xtabla, ytabla).nombre_prop = lookahead
        TablaR(Xtabla, ytabla).Visitado = 0
        TablaR(Xtabla, ytabla).concluido = 0
    End If
End If

```

```

TablaR(Xtabla, ytabla).Tipo = 0
escorrecto = RespuestaReactivook(lookahead)
Match (lookahead)
Match ("es")
If ((lookahead = "correcto") And (escorrecto = 1)) Or ((lookahead = "incorrecto") And (escorrecto = -1)) Then
  Match (lookahead)
  TablaR(Xtabla, ytabla).valor_verdad = 100
  TablaR(Xtabla, ytabla).ul_valor_verdad = 100
ElseIf ((lookahead = "correcto") Or (lookahead = "incorrecto")) And ((escorrecto = -1) Or (escorrecto = 1)) Then
  Match (lookahead)
  TablaR(Xtabla, ytabla).valor_verdad = -100
  TablaR(Xtabla, ytabla).ul_valor_verdad = -100
ElseIf ((lookahead = "correcto") Or (lookahead = "incorrecto")) And (escorrecto = 0) Then
  Match (lookahead)
  TablaR(Xtabla, ytabla).valor_verdad = 0
  TablaR(Xtabla, ytabla).ul_valor_verdad = 0
Else
  MError ("Error, espera 'correcto' o 'incorrecto'")
End If
Else
  MError ("Error, espera número reactivo ")
End If 'if not reservada
ElseIf lookahead = "calificacion" Or lookahead = "calificación" Then
  Match (lookahead)
  operador = lookahead
  If atrib <> 255 Then
    MError ("Espera operando lógico (=,>, etc) ")
  Else
    Match (lookahead)
    If atrib <> 1 And atrib <> 2 Then
      MError ("Espera un número entero o real ")
    Else
      TablaR(Xtabla, ytabla).nombre_prop = "calificación"
      TablaR(Xtabla, ytabla).ul_valor_verdad = 0
      TablaR(Xtabla, ytabla).Visitado = 0
      TablaR(Xtabla, ytabla).concluido = 0
      TablaR(Xtabla, ytabla).Tipo = 0
      If Cumplecalificacion(operador, lookahead) Then
        TablaR(Xtabla, ytabla).valor_verdad = 100
      Else
        TablaR(Xtabla, ytabla).valor_verdad = -100
      End If
      Match (lookahead)
    End If
  End If
ElseIf lookahead = "aciertos" Then
  Match ("aciertos")
  Match ("(")
  numaci = 0
  While (lookahead <> ")") And Not (finarchivo)
    If RespuestaReactivook(lookahead) = 1 Then
      numaci = numaci + 1
    End If
    Match (lookahead)
    If lookahead = "," Then
      Match (",")
    End If
  End If

```

```

Wend
If finarchivo Then
  MError ("Fin de archivo inesperado, espera ")
Else
  Match ("")
  operador = lookahead
  If atrib <> 255 Then
    MError ("Espera operando lógico (=,>, etc ")
  Else
    Match (lookahead)
    If atrib <> 1 Then
      MError ("Espera un número entero ")
    Else
      TablaR(Xtabla, ytabla).nombre_prop = "aciertos"
      TablaR(Xtabla, ytabla).ul_valor_verdad = 0
      TablaR(Xtabla, ytabla).Visitado = 0
      TablaR(Xtabla, ytabla).concluido = 0
      TablaR(Xtabla, ytabla).Tipo = 0
      If Cumpleaciertos(operador, lookahead, numaci) Then
        TablaR(Xtabla, ytabla).valor_verdad = 100
      Else
        TablaR(Xtabla, ytabla).valor_verdad = -100
      End If
      Match (lookahead)
    End If
  End If
End If
ElseIf atrib = 100 Then ' if no es reactivo, es una var. alfanumérica
  negacion = 0
  If lookahead = "no" Then
    Match ("no")
    negacion = -1
  End If
  TablaR(Xtabla, ytabla).nombre_prop = lookahead
  TablaR(Xtabla, ytabla).valor_verdad = 0
  TablaR(Xtabla, ytabla).ul_valor_verdad = 0
  TablaR(Xtabla, ytabla).Visitado = 0
  TablaR(Xtabla, ytabla).concluido = 0
  If CualPregunta(TablaR(Xtabla, ytabla).nombre_prop, preg) > 0 Then
    TablaR(Xtabla, ytabla).Tipo = 2
  Else
    TablaR(Xtabla, ytabla).Tipo = 1
  End If
  TablaR(Xtabla, ytabla).negacion = negacion
  Match (lookahead)
Else MError ("Espera nombre de una proposición")
End If
End Sub

Sub ParseaProposicion ()
Dim cicloand As Integer
Dim strecomen As String
Xtabla = 0
ytabla = ytabla + 1
Match ("si")
cicloand = 1
If lookahead = "entonces" Then

```

```

MError ("Espera una proposición")
Else
While (lookahead <> "entonces") And Not (finarchivo) And Not (e_error)
  Xtabla = Xtabla + 1
  ParseaPremisa
  If Not (e_error) Then
    If (lookahead = "y") And ((cicloand = 1) Or (cicloand)) Then
      cicloand = -1
      Match ("y")
    ElseIf (lookahead = "o") And ((cicloand = 1) Or Not (cicloand)) Then
      cicloand = 0
      Match ("o")
    ElseIf ((lookahead = "y") And Not (cicloand)) Or ((lookahead = "o") And (cicloand)) Then
      MError ("No puede haber conjunciones y disyunciones en una prop.")
    ElseIf lookahead <> "entonces" Then
      MError ("Espera entonces, o, o y")
    End If
  End If
Wend
Match ("entonces")
If (lookahead = "recomendación") Or (lookahead = "recomendacion") Then
  huborecomendaciones = True
  Match (lookahead)
  Match (""")
  While (lookahead <> """) And Not (finarchivo) And Not (e_error)
    strecomen = strecomen + lookahead + " "
    Match (lookahead)
  Wend
  If (finarchivo) And (lookahead <> """) Then
    MError ("Error, no encontró """)
  Else
    TablaC(ytabla).nombre_prop = strecomen
    TablaC(ytabla).valasig = 100
    TablaC(ytabla).valor_verdad = 0
    TablaC(ytabla).ul_valor_verdad = 0
    TablaC(ytabla).concluido = 0
    TablaC(ytabla).Visitado = 0
    TablaC(ytabla).conjuncion = cicloand
    TablaC(ytabla).TipoConc = 0 'recomendación
    Match (""")
  End If
Else ' si es una proposición
  If atrib = 100 Then
    TablaC(ytabla).negacion = 0
    If lookahead = "no" Then
      Match (lookahead)
      TablaC(ytabla).negacion = -1
    End If
    TablaC(ytabla).nombre_prop = lookahead
    TablaC(ytabla).valasig = 100
    TablaC(ytabla).valor_verdad = 0
    TablaC(ytabla).ul_valor_verdad = 0
    TablaC(ytabla).Visitado = 0
    TablaC(ytabla).concluido = 0
    TablaC(ytabla).conjuncion = cicloand
    TablaC(ytabla).TipoConc = 1 ' proposición
    Match (lookahead)
  End If
End If

```

```

Else
  MError ("Error, espera un nombre de proposición")
End If
End If
If Not e_error Then
  If lookahead = "(" Then
    Match "("
    If atrib <> 1 Then 'si el lookahead no es numérico
      MError ("Espera un número")
    Else
      TablaC(ytabla).valasig = CInt(lookahead)
      Match lookahead
      If lookahead = "%" Then
        Match "%"
      End If
      Match ""
    End If
  End If
End If
End If
End Sub

```

```

Sub PonConcluidos (nomprop As String)
Dim i As Integer
For i = 1 To ytabla
  If (TablaC(i).nombre_prop = nomprop) Then
    TablaC(i).concluido = True
  End If
Next i
End Sub

```

```

Function Preguntar (nombre_prop As String) As Integer
Dim s, i As Integer
Dim preg As String
i = CualPregunta(nombre_prop, preg)
If (i > 0) And (TablaP(i).valor_verdad <> 0) Then
  Preguntar = TablaP(i).valor_verdad
ElseIf i > 0 Then
  s = MsgBox(preg, 4)
  If s = 6 Then
    TablaP(i).valor_verdad = 100
    Preguntar = 100
  ElseIf s = 7 Then
    TablaP(i).valor_verdad = -100
    Preguntar = -100
  End If
Else Preguntar = 0
End If
End Function

```

```

Sub RazonaForward ()
Dim ni As Integer
Dim i As Integer
calculaVv = False

```

```

While Not calculaVv

```

```

cambia = False
i = 0
'concluye = False
While (i < ytabla) 'And Not (concluye)
i = i + 1
b = True
If Not TablaC(i).Visitado Then
  If TablaC(i).conjuncion Then
    CicloAndF (i) 'recorre las proposiciones en and y va calculando el
      'valor de certeza menor para deducir la proposición
  Else
    CicloOrF (i) 'recorre las proposiciones en or y va calculando el
      'valor de certeza menor para deducir la proposición
  End If
  valorv = v1
  v2 = v1
  If b Then
    TablaC(i).Visitado = True
    TablaC(i).valor_verdad = valorv
    TablaC(i).ul_valor_verdad = valorv
    While (ExisteSigProp(TablaC(i).nombre_prop, ni)) And (b)
      ' ciclo para que busque todas las premisas que concluyen la
      ' proposición que se está analizando. Si existe, calcula
      ' el valor de verdad de la nueva premisa y los combina
      If TablaC(ni).conjuncion Then
        CicloAndF (ni)
      Else
        CicloOrF (ni)
      End If
      If b Then
        TablaC(ni).valor_verdad = v1
        valorv = Combina(v1, v2, TablaC(ni).nombre_prop)
        TablaC(ni).ul_valor_verdad = valorv
        TablaC(ni).Visitado = True
      End If
      v2 = valorv
    Wend
  End If
  If b Then
    'si logró concluir sobre esta proposición }
    cambia = True
    PonConcluidos (TablaC(i).nombre_prop)
    TablaC(i).ul_valor_verdad = valorv
    TablaC(i).Visitado = True
  Else
    TablaC(i).Visitado = False
  End If
  'If TablaC(i).Nombre_prop = Nombre_prop And TablaC(i).concluido Then
  ' concluye = True
  'End If

End If
Wend
If (Not (cambia)) Then 'Or (concluye) Then
  calculaVv = True
End If
Wend

```

End Sub

Function RespuestaReactivook (idreact As String) As Integer

Dim encontro, numr, i As Integer

encontro = False

For i = 1 To totpreg

If Arreglo(i).idreactivo = idreact Then

encontro = True

numr = i

End If

Next i

If Not encontro Then

MsgBox ("Warning: El reactivo " + idreact + " no existe o no fue contestado")

RespuestaReactivook = 0

Else

RespuestaReactivook = Arreglo(numr).acierto

End If

End Function

Function Token () As String

Dim T As String

Dim errg As Integer

Dim R As String

T = ""

While (Mid\$(Entrada, indicearchivo, 1) = " ") Or (Mid\$(Entrada, indicearchivo, 1) = Chr(13)) Or (Mid\$(Entrada, indicearchivo, 1) = Chr(10))

indicearchivo = indicearchivo + 1

Wend

finarchivo = indicearchivo > Longarchivo

If Not finarchivo Then

R = Leecaracter()

Select Case R

Case "a" To "z", "á", "é", "í", "ó", "ú", "Á", "É", "Í", "Ó", "Ú"

While Not (finarchivo) And (((R >= "a") And (R <= "z")) Or (R = "\_") Or (R = "á") Or (R = "é") Or (R = "í") Or (R = "ó") Or (R = "ú") Or ((R >= "0") And (R <= "9")))

T = T + R

R = Leecaracter()

Wend

indicearchivo = indicearchivo - 1

atrib = 100

Case "0" To "9", "."

atrib = 1

errg = False

While Not (finarchivo) And (((R >= "0") And (R <= "9")) Or (R = ".")) And Not (errg)

T = T + R

errg = (atrib = 2) And (R = ".")

If R = "." Then

atrib = 2

End If

If Not errg Then

R = Leecaracter()

End If

Wend

indicearchivo = indicearchivo - 1

Case "="

T = T + R

atrib = 255

```
Case ">"
  T = T + R
  R = Leecaracter()
  If R = "=" Then
    T = T + R
  Else
    indicearchivo = indicearchivo - 1
  End If
  atrib = 255
Case "<"
  T = T + R
  R = Leecaracter()
  If (R = ">") Or (R = "=") Then
    T = T + R
  Else
    indicearchivo = indicearchivo - 1
  End If
  atrib = 255
Case Else
  T = T + R
  atrib = -100
End Select
End If
Token = T
End Function
```

W26