

30
2FEJ



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA EXPERTO TUTORIAL
PARA LA CAPACITACIÓN EN
CONTROL DE TRÁNSITO AÉREO

FASE DE APROXIMACIÓN

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

P R E S E N T A:

FLOR SELENE GPE. COSS Y SALAS



DIRECTOR DE TESIS: DR. FELIPE LARA ROSANO
ASESOR: C.T.A. LUIS MANUEL FLORES SOLÓRZANO

Ciudad Universitaria México, D. F.

1995

FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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

Hoy decidí finalmente escribir la dedicatoria de mi tesis. Un buen tiempo me quedé mirando la pantalla, intentando dar un orden a mi corazón. No lo encontré y sin embargo fue preciso escribir el nombre de alguien primero....

A Dios, porque Dios me lo ha dado todo:

Mis abues (los que nacieron primero, los que me amaron y me dieron los últimos años de su vida).

Mis papis, a quienes les debo su vida, su entrega, su amor y por qué no decirlo, también su dolor.

Mi NZto, Fer, mi amor, mi vida.

Mis hermanos (si yo pude, ¿por qué ustedes no?)

Mis tías Raquel, Chela y Rosita, quienes me han escuchado, abrazado y querido.

Mi tío Luis, quien hizo realidad una idea.

Mis amigos.

Oigan..... los quiero mucho!

3 de septiembre de 1995

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México, a la cual le debo mi formación profesional.

A la Facultad de Ingeniería, mi segunda casa.

Al Instituto de Ingeniería, por todas las facilidades que me brindó para desarrollar esta tesis.

A mi director de tesis y maestro, el Dr. Felipe Lara Rosano, por su entusiasmo, conocimientos y por ser el maestro que despertó en mi, el interés hacia la IA

A mi asesor, el C.T.A. Luis Manuel Flores Solórzano, por compartir sus conocimientos de CTA y experiencia en la instrucción.

Al coordinador del Laboratorio de IA, M.I. Nicolás Kemper Valverde, por sus recomendaciones y disponibilidad.

A mis compañeros de Laboratorio de IA, por su apoyo y comentarios

SISTEMA EXPERTO TUTORIAL
PARA LA CAPACITACIÓN EN
CONTROL DE TRÁNSITO AÉREO

FLOR SELENE GPE. COSS Y SALAS

1995

ÍNDICE

1. Introducción	1
<hr/>	
2. Capacitación en el Control de Tránsito Aéreo	5
Aspectos Generales del Control de Tránsito Aéreo (CTA)	5
Manejo del Espacio Aéreo	7
Historia	7
Estándares Internacionales	8
Servicios de Tránsito Aéreo	10
Control de Aeródromo	12
Control de Aproximación	12
Control de Área	13
Licencias al Personal	14
Ayudas de Navegación y Comunicación	15
Automatización y Control de Tránsito Aéreo	16
Procesamiento de datos de vuelo	17
Procesamiento de datos de radar	17
La Capacitación de Controladores en México	17
Lugares de capacitación	17
Plan de cursos	18
Los Controladores de Tránsito Aéreo en la práctica	20
<hr/>	
3. Inteligencia Artificial y Sistemas Expertos	22
Inteligencia Artificial (IA)	23
Definición	24
Tipos de experiencia humana que maneja la IA	24
Historia	26
Áreas de aplicación y desarrollo	28
Sistemas Expertos (SE)	29

Definición	30
Historia	31
Tipos de Sistemas Expertos	35
Aplicaciones y ventajas	36
Cualidades de los SE	37
SE vs. expertos humanos	39
SE vs. programas tradicionales	40
Estructura general de un SE	41
Base de Conocimientos	42
Motor de Inferencias	43
Adquisición de conocimientos	44
Componente para la explicación	45
Interfaz del usuario	45
Representación del conocimiento	46
Reglas de producción	47
Redes semánticas	48
Frames o Marcos	49
Cálculo de predicados	50
Tipos de inferencia	50
Encadenamiento hacia adelante	50
Encadenamiento hacia atrás	52
Inferencia Mixta	52
Búsqueda aleatoria	53
Búsqueda heurística	53
Niveles de conocimiento	53
Conocimiento de decisión	54
Conocimiento de soporte	54
Meta-conocimiento (reglas acerca de reglas)	54
Lenguajes y Herramientas para SE	54
Lenguajes imperativos	56
Lenguajes funcionales	56
Lenguajes orientados a objetos	56
Lenguajes declarativos	57
Los Shells	57

4. Aprendizaje y Sistemas Expertos Tutoriales	59
Acerca del aprendizaje	59
La psicología conductista	60
La psicología cognitiva	60
La psicología evolutiva	60
El ciclo de aprendizaje	61
Aspectos neurológicos	62
Reaprendizaje	63
La evaluación	63
Instrumentos para la evaluación	64
El pizarrón	65
Aprendizaje individualizado	66
Aprendizaje por computadora (tutoriales)	68
Ventajas del uso de las computadoras	69
Presentando a los tutoriales inteligentes	70
Software educacional como contenedor de información	72
Sistemas Expertos Tutoriales: codificación del conocimiento	73
Disciplinas que interaccionan con los SET	74
Comunicación del conocimiento	75
Consideraciones para elaborar los SET	76
Módulos: experto, del estudiante y de la evaluación	76
La forma de comunicación con el usuario final	77
<hr/>	
5. Desarrollo, aplicación y evaluación de SECATA	78
El desarrollo de un SET	78
Proceso para impletar un SE	79
Inicialización del proyecto	80
Evaluación de necesidades	81
Implementación	82
Depuración del SE	82
Desarrollo de SECATA	82
Inicialización del proyecto	83

Evaluación de necesidades	83
El shell elegido	83
Tipo de inferencia utilizada	85
Fuentes de conocimiento utilizadas	85
Requerimientos del sistema	85
Implementación	86
Diseño de la Interfaz del usuario	86
Estructura de SECATA	87
Aplicación y evaluación de resultados	90
Operación general de SECATA	90
Validación de SECATA	96
Depuración de SECATA	98
Ciclo de vida de SECATA	98

6. Conclusiones

100

Glosario
 Bibliografía
 Apéndice I
 Apéndice II

INTRODUCCIÓN

Ya desde finales de la década de los ochenta, se ha trabajado en el Instituto de Ingeniería de la UNAM, bajo la dirección del Dr. Felipe Lara Rosano en la elaboración de sistemas expertos con títulos diversos.

Todos los sistemas expertos comienzan con la percepción de un problema cuya resolución, por medio de los métodos de programación tradicionales, o basados en algoritmos, es difícil. Pues bien, así se originó la idea de SECATA, una aplicación desarrollada con técnicas de la Inteligencia Artificial para capacitar en la fase de aproximación no radar al personal que presta sus servicios como controladores de aeródromo.

Esta tesis pretende mostrar las facilidades que brindan las computadoras personales como herramientas educativas debido a factores como:

- La motivación que despierta una computadora en los estudiantes que no tienen contacto continuo con ellas o por la novedad que presente un sistema tutorial.
- La individualización que proporciona una computadora al dedicarse a un solo estudiante y que proporciona una disminución del 30% del tiempo invertido en aprender.
- La información visual interactiva.
- La estimulación de ambos hemisferios del cerebro al estimular la parte creativa mediante gráficos y la lógica mediante conceptos textuales.

A lo que hay que agregar las ventajas de la tecnología de los sistemas expertos tutoriales:

- Presentan la información de una manera dinámica y especial para cada estudiante.
- Disponen de gran cantidad de conocimiento sobre el problema, fruto de la experiencia.
- Realizan una labor similar a la que realizaría un instructor experto en el tema pero dedicándose a un solo estudiante.
- Restringen su razonamiento a una parte determinada del problema.

El contenido de SECATA permitirá que cualquier alumno que haya cursado previamente la Licenciatura de Control de Tránsito Aéreo clase ^{III} y que después de cierto tiempo haya perdido contacto con la fase de Control de Aproximación no radar, pueda obtener incuestionablemente el mismo nivel de conocimientos usando el sistema experto tutorial, que un controlador que

emplee los métodos tradicionales para recordar o reaprender la teoría en Operaciones IFR.

Así pues, el controlador podrá desempeñar su labor en la fase de aproximación, aún en diferentes estaciones, de igual forma y reportando un mayor aprovechamiento en comparación a un alumno que haya tomado el curso bajo las mismas condiciones iniciales pero mediante el método tradicional de enseñanza.

Dado que el tutorial está diseñado para controladores, se obliga al usuario a contestar una evaluación previa al curso, pues es necesario cierto nivel de conocimientos para que se obtenga un buen aprovechamiento del tutorial, por lo cual es requisito la obtención de una calificación de 8, de una escala del 1 al 10, como mínimo en el examen previo que figura al inicio de la aplicación.

La tesis consta de seis capítulos, glosario de los términos usados y dos apéndices que contienen las siglas y abreviaturas usadas, así como el listado reducido del sistema en PRL. La bibliografía abarca tanto la parte de Inteligencia Artificial y Sistemas Expertos como la del Control de Tránsito Aéreo y el tema de Aprendizaje.

El capítulo 2 comprende la Capacitación en Control de Tránsito Aéreo y pretende dar una visión general de lo que significa el control de tránsito aéreo, una perspectiva histórica, la definición de las tres fases que lo comprenden (aproximación de aeródromo, control de aproximación y control de área), así como la forma en que se capacita a los controladores en México.

El capítulo 3 se dedica a lo que es la Inteligencia Artificial, especialmente a una de sus áreas: los Sistemas Expertos. Se describen los tipos de sistemas expertos así como sus ventajas y aplicaciones. Se explica a detalle su estructura, la manera en como se representa el conocimiento en ellos y la forma en como se realiza el razonamiento. Finalmente, se habla de los lenguajes y las herramientas para crearlos.

El capítulo 4 trata de los aspectos pedagógicos de la enseñanza-aprendizaje e introduce al lector al concepto de los Sistemas Expertos Tutoriales.

El capítulo 5 presenta el desarrollo de un Sistema Experto y la forma en como se desarrolló SECATA. En este capítulo podrá encontrar la forma en como está estructurado el sistema experto tutorial y un pequeño manual de operación. Al término del capítulo se describe la manera en como fue aplicado SECATA y los resultados que se obtuvieron de tal aplicación.

Finalmente, el capítulo 6 es una conclusión de la tesis que recalca los puntos principales de ésta.

2

CAPACITACIÓN EN EL CONTROL DE TRÁNSITO AÉREO

ASPECTOS GENERALES DEL CONTROL DE TRÁNSITO AÉREO (CTA)

En cualquier país la constitución de las comunicaciones incluye la *transportación aérea* y uno de los elementos esenciales de ésta es el *Control de Tránsito Aéreo (CTA)*.

El CTA es un servicio cuyo primer objetivo es el de otorgar seguridad a aquellos que viajan en el aire. Los prestadores de este servicio tienen que reconocer los diferentes requerimientos de los usuarios del aire.

Los usuarios del aire son tales como:

- transportes comerciales de pasajeros,
- defensa militar,
- aviación ejecutiva y privada.

Los controladores de tránsito aéreo deben identificar igualmente las características de rendimiento de diversas aeronaves, las cuales varían desde las aeronaves ligeras con una velocidad promedio de 160 nudos hasta las aeronaves Jet de fuselaje largo con velocidades hasta de 500 nudos y en ocasiones, las aeronaves supersónicas.

El servicio que se da a los usuarios del aire se complica cuando se presentan salidas y aterrizajes en exceso (de uno por minuto, por ejemplo), como es el caso de aeropuertos internacionales muy ocupados; y cuando varios aeropuertos se encuentran dentro de áreas terminales igualmente ocupadas, esto es, cerca de uniones de rutas aéreas que sirven para otros destinos y en proximidades cercanas a aeropuertos de capacidad similar.

Las bases aéreas militares también tienen problemas específicos, pues hay sectores, en los que transitan aeronaves civiles, que colindan con las áreas operacionales y de adiestramiento. Estos sectores, a menudo, están en conflicto con redes de rutas y áreas terminales que sirven, como ya se mencionó, a movimientos de transporte aéreo civil.

El arte del Control de Tránsito Aéreo, se fundamenta en la separación de una nave aérea de otra aeronave, y de que ese espacio que las separa nunca sea invadido. Las reglas que deben tomarse en cuenta para asegurarse de que se conserve dicha separación, durante su vuelo (desde el punto de partida hasta el aterrizaje), son conocidas como separaciones estándar o separaciones mínimas.

Deben observarse separaciones en tres dimensiones: tiempo (separación longitudinal), posición geográfica (separación lateral u horizontal) y altura (separación vertical).

Esto es un arte que, como hace alusión Arnold Field¹, puede considerarse como un juego de ajedrez a alta velocidad.

MANEJO DEL ESPACIO AÉREO

Historia

En forma por demás coincidente, casi todos los adelantos que se han dado en la Industria Aeronáutica se derivan del desarrollo tecnológico llevado a cabo por diferentes países para enfrentar conflictos bélicos, algunos locales, la mayoría mundiales; tal como sucedió después de la Segunda Guerra Mundial, en la que se descubrieron las potencialidades de la transportación aérea, trayendo como consecuencia el incremento del uso del espacio aéreo.

La necesidad inherente de establecer un orden y unas reglas o normas a seguir de manera mundial, precisó la formación de un organismo a este nivel, para regular la aviación civil y el establecimiento de las bases para la coordinación entre ésta y la aviación militar.

En Europa se comenzó con una forma de control de tránsito aéreo, inicialmente restringido a las vecindades del aeródromo², contándose con un sistema de asesoría mediante telégrafo fuera de esa área. Fue en el departamento Air Ministry de Inglaterra donde se comenzaron a construir los fundamentos de los sistemas del Control de Tránsito Aéreo.

Uno de los primeros sistemas de radioayudas electrónicas para la navegación aérea, y que fueron la base para establecer las primeras rutas aéreas, fue el NDB (Radiofaro No Direccional). Con este sistema, los pilotos podían establecer su posición con relación a la superficie de la tierra y reportar a los

¹ Arnold Field es el autor del libro International Air Traffic Control, management of the world's airspace

² Campo provisto de pistas e instalaciones necesarias para el despegue y aterrizaje de aeronaves

controladores, el tiempo estimado en que la aeronave llegaría a las proximidades del aeródromo; pero existía el inconveniente de que lejos de los emplazamientos de estas radioayudas, no se podía contar con ayuda para la navegación aérea.

Estos problemas eran comunes en todo el mundo a excepción de Estados Unidos, que por su posición geográfica y la necesidad de comunicarse a través de grandes distancias, se vieron obligados, no solamente a utilizar las aeronaves como medio de transporte, sino también a desarrollar un sistema de aerovías³ para regular y proteger el vuelo de una aeronave entre los puntos de partida y destino. Estas aerovías, fueron las primeras en aparecer en la red mundial de aerovías que eran, y siguen siendo, de diez millas de ancho y se extendían aproximadamente de 3 a 10 mil pies sobre el nivel de la superficie terrestre.

Al finalizar el año de 1944 en Chicago, nace la Organización de Aviación Civil Internacional OACI (ICAO por sus siglas en inglés) la cual, desde esa fecha a la actualidad, con la participación de treinta y tres estados miembros, ha venido dictando las normas, recomendaciones y procedimientos (estándares internacionales) que deben ser acatados por todos los estados.

Actualmente, el espacio aéreo está dividido en Regiones de Información de Vuelo (*Flight Information Region, FIR*) usualmente abajo de 20, 000 pies y en Regiones de Información de Vuelo Superiores (*Upper Flight Information Region, UIR*), sobrepuestas al nivel del FIR.

Estándares Internacionales

Como ya se mencionó, los estándares internacionales se refieren a las normas, recomendaciones y procedimientos que deben ser adoptados por los estados miembros de la OACI (entre los que se incluye México).

³ Ruta establecida para el vuelo comercial de las aeronaves

Estos estándares abarcan todos los aspectos dentro de la Industria Aeronáutica y son enmarcados en los Anexos de la OACI (1 al 18) que a continuación se mencionan:

1. Licencias al Personal

2. Reglamento del Aire

Fija las reglas del aire que rigen para pilotos y controladores.

3. Servicio Meteorológico para la Navegación Aérea Internacional

4. Cartas Aeronáuticas

5. Unidades de Medida que se Emplearán en la Operaciones Aéreas y Terrestres.

6. Operación de Aeronaves

Parte I. Transporte aéreo comercial internacional - aviones

Parte II. Aviación general internacional - aviones

Parte III. Operaciones internacionales - helicópteros

7. Marcas de Nacionalidad y de Matrícula de las Aeronaves

8. Aeronavegabilidad

9. Facilitación

10. Telecomunicaciones Aeronáuticas

Volumen I.

Parte I. Equipos y sistemas

Parte II. Radiofrecuencias

Volumen II. Procedimientos de comunicación

11. *Servicios de Tránsito Aéreo*
12. *Búsqueda y Salvamento*
13. *Investigación de Accidentes de Aviación*
14. *Aeródromos*
 - Volumen I. Diseño y operación de aeródromos*
 - Volumen II. Helipuertos*
15. *Servicios de Información Aeronáutica*
16. *Protección del Medio Ambiente*
 - Volumen I. Ruido de las aeronaves*
 - Volumen II. Emisiones de los motores de las aeronaves*
17. *Seguridad - Protección de la aviación civil internacional contra actos de interferencia ilícita*
18. *Transporte sin riesgos de Mercancías Peligrosas por Vía Aérea*

Servicios de Tránsito Aéreo

El punto 11 de los Anexos de la OACI, el cual se mencionó anteriormente, se refiere a los servicios de tránsito aéreo, éste indica a cada estado cuándo y cómo debe establecer estos servicios:

1. *Servicio de Información de Vuelo*
2. *Servicio de Alerta*
3. *Servicio de Asesoramiento de Tránsito Aéreo*
4. *Servicio de Control de Tránsito Aéreo*

- a) Servicio de control de aeródromo
- b) Servicio de control de aproximación
- c) Servicio de control de área

En adición a los anexos que comprenden los estándares internacionales dictados por la OACI, se establecieron y publicaron los Procedimientos para los Servicios de Navegación Aérea (PANS) que se citan a continuación:

- ABC - Abreviaturas y Códigos de la OACI
- OPS - Operación de Aeronaves

Volumen I. Procedimiento de vuelo

Volumen II. Construcción de procedimientos de vuelo visual y por instrumentos

- RAC - Reglamento del Aire y Servicios de Tránsito Aéreo
- Procedimiento para los servicios de navegación aérea. Establece cómo prestar estos servicios fijando las reglas que se deben cumplir (Separaciones estándar o mínimas).

La simple lectura de los títulos de los anexos, así como de los procedimientos para los servicios de navegación aérea (PANS), permite ver que todos los posibles aspectos de la industria aeronáutica son tratados y consecuentemente normados. Cada anexo está íntimamente relacionado con los restantes, ya sea ligera o fuertemente.

Por la variedad de las disciplinas que abarcan, también es casi imposible que una sola persona pueda dominar todas ellas, pero es imprescindible para las personas que se dediquen a una u otra actividad en esta industria, conocer perfectamente su especialidad y las ligas con las demás especialidades.

Ninguna de las disciplinas o especialidades en esta industria puede sustituirse, por lo que todas ellas se deben aprobar a un mismo nivel tomando cada una de ellas una importancia preponderante, cuando las demás han cumplido con su cometido en tiempo y lugar; así, producto de la necesidad de que las operaciones aéreas (civiles y militares) se realicen con orden, seguridad y expeditas, se establecen las normas para los servicios de tránsito aéreo.

Control de aeródromo

Una unidad de control de aeródromo provee servicios de control de tránsito aéreo principalmente, a las aeronaves que vuelan con referencia visual a la superficie dentro de las inmediaciones del aeropuerto (zona de tránsito del aeródromo) y también, al área de maniobras (pistas y calles de rodaje).

La unidad se encuentra en una cabina que permite el control visual. Aunque es una unidad separada, puede combinarse de manera temporal o permanente con la unidad de control de aproximación.

Control de aproximación

En un aeródromo ocupado (con continuos despegues y aterrizajes), es usual que el servicio de control de aproximación sea dirigido desde un cuarto separado ya que no se requiere ver físicamente a la aeronave bajo control, ni estar asociado, físicamente con el control de aeródromo. Este cuarto de operaciones suele encontrarse situado en la torre, en el piso inmediato a la cabina del control de aeródromo. Debe haber cierta flexibilidad para combinar la unidad de control de aproximación no radar con las unidades de control de aproximación radar y con el control de aeródromo o incluso, compartir responsabilidades con la del control de área.

Las áreas en las que se desempeña la labor de control de aproximación radar, anteriormente requerían de una iluminación especial, la cual se daba por medio de lámparas individuales, colocadas cuidadosamente para que la luz no se reflejara en la pantalla del radar. Con el paso del tiempo, las capacidades de estas pantallas han mejorado, permitiendo mayor iluminación en el cuarto de control.

Apoyando al controlador de aproximación radar o no radar, se encuentra un asistente de control de tránsito aéreo (controlador de aproximación) quien estará tratando con llamadas telefónicas de salida y entrada, preparando la información en las tiras de progreso de vuelo, informando al control de aeródromo de la actual secuencia de arribo de aeronaves y, cuando es aplicable, operando la terminal de computadora que está conectada al centro de control de área.

En muchos aeródromos, civiles y militares, el radar es usado para asistir el control de aproximación. No es necesario que las naves reciban el servicio de radar, pero en circunstancias de tránsito denso o requerimientos de aproximación asistida por radar (SRA, PAR), el control de aproximación no radar delega sus responsabilidades a la aproximación radar.

La aplicación de los mínimos de separación, señalados anteriormente, pueden variar dependiendo del aeródromo en el cual se está proporcionando el control de aproximación, ya sea que el aeródromo esté fuera o no de un espacio aéreo controlado con reglas especiales. La razón de las diferentes responsabilidades del control de aproximación fuera del espacio aéreo controlado y sus reglas especiales, es que no hay requerimientos legales para vuelos conducidos en el espacio aéreo que está fuera de zonas de tránsito del aeródromo y las instrucciones son solamente consultivas.

Control de área

Provee servicios de control de tránsito aéreo en las regiones FIR y UIR, área que se extiende algunas veces más allá de sus territorios; el control de área es responsable del control de la red de rutas aéreas y de las áreas

terminales. Adicionalmente, es responsable de servicios de alerta, peligro y búsqueda, y de la provisión de servicios de información de vuelos a aeronaves que vuelan dentro de la totalidad de este espacio aéreo.

Sin embargo, el control de área no está desligado y, como ya se mencionó, da apoyo al control de aproximación absorbiendo los retardos del arribo de aeronaves y regulando el flujo de las aeronaves que se encuentran partiendo.

Licencias al Personal

Establece los requerimientos mínimos de conocimiento así como los privilegios y responsabilidades del personal técnico aeronáutico.

Las personas que presten los servicios de control de tránsito aéreo deben estar apropiadamente calificadas y tener la licencia respectiva para el servicio que vaya a proporcionar; la licencia de control de tránsito aéreo, normalmente, trae los grados de control de aeródromo, control de aproximación, control de área y la capacitación radar. Para que un controlador pueda responsabilizarse de una posición, debe ser validado mediante la supervisión de la autoridad aeronáutica.

Para su desempeño, el controlador de tránsito aéreo debe tener conocimientos de:

- a) Meteorología aeronáutica.
- b) Cartas aeronáuticas.
- c) Unidades de medida utilizables.
- d) Rendimiento y/o características de las diferentes aeronaves.
- e) Matrículas de las aeronaves.
- f) Navegación aérea (aeronavegabilidad).

- g) Facilidades para las aeronaves y pasajeros en el aeropuerto (facilitación).
- h) Telecomunicaciones aeronáuticas (herramientas indispensables para proporcionar los servicios).
- i) Como auxiliar a los servicios de búsqueda y salvamento.
- j) Investigar o conocer los resultados de investigación de incidentes y accidentes de aviación así como su prevención.
- k) Aeropuerto(s) a los que sirve.
- l) Información aeronáutica, publicada por el estado, en donde se incluyen datos importantes cómo, dónde y quién proporciona:
 - información meteorológica aeronáutica,
 - ubicación y frecuencia de operación de las radio ayudas para la navegación,
 - facilidades de migración, aduana, combustible, hoteles y servicios diversos en cada uno de los aeropuertos,
 - diferencias que el estado ha manifestado a la OACI de las normas y recomendaciones,
 - forma en como está establecido el servicio de búsqueda y salvamento.
- m) Procedimientos de abatimiento de ruido

AYUDAS DE NAVEGACIÓN Y COMUNICACIÓN

Las ayudas de navegación pueden definirse de dos formas: terrestres, las cuales, como su nombre lo dice, son instalaciones en la superficie de la tierra que tienen una posición geográfica conocida y publicada, y de abordó

que se refieren al equipo montado en la aeronave, el cual permite al piloto interrogar y obtener información de las instalaciones terrestres.

Existe una tercera categoría, llamada sistema de navegación autónomo o "a bordo", el cual permite a la aeronave navegar sin recurrir a la radio ayuda terrestre, tal como un satélite de navegación o la navegación inercial (INS).

Sin embargo, se requiere de facilidades adicionales para pilotos y controladores que les permitan llevar a cabo sus tareas asignadas. En términos generales, éstas son conocidas como ayudas de comunicación que usan la radio telefonía para la comunicación entre el aire y la tierra.

Por otra parte, se ha incrementado el uso de las computadoras como un método rápido de comunicación y tienen un papel especial en el Control de Tránsito Aéreo.

AUTOMATIZACIÓN Y CONTROL DE TRÁNSITO AÉREO

Generalizando, hay tres elementos en la aplicación de la automatización en esta disciplina en particular, y son:

- Procesamiento de datos de vuelo
- Procesamiento de datos de radar
- La correlación entre el procesamiento de datos de vuelo y el procesamiento de datos de radar

Dentro de cada uno de los puntos anteriores hay variantes, pues dependen del sistema de control de tránsito aéreo que se esté utilizando.

Procesamiento de datos de vuelo

El procesamiento de datos de vuelo es la función que está concernida con la entrada a la computadora de los datos de vuelo, que han sido extraídos originalmente del plan de vuelo y guardados posteriormente como información de computadora para ser usada como referencia de una situación "viva" o "activa".

Procesamiento de datos de radar

El término radar es un acrónimo de la frase *radio detection and ranging*, y el rango y señales de relación recibidas por la antena de forma análoga, por lo cual se necesita que sean digitalizadas para ser procesadas por una computadora y, posteriormente, pueda ser desplegada la información directamente a la pantalla del radar.

LA CAPACITACIÓN DE CONTROLADORES EN MÉXICO

LUGARES DE CAPACITACIÓN

En 1953 se inició la capacitación en forma de controladores de tránsito aéreo, al realizarse en ese año, un convenio de cooperación entre el gobierno de México y la Organización de Aviación Civil, con el cual se formó el Centro Internacional de Adiestramiento de Aviación Civil (CIAAC), comenzando con los cursos de preparación para el personal técnico aeronáutico como: meteorólogos aeronáuticos, despachadores de aeronaves, mecánicos en motores, mecánicos en planeadores y controladores de tránsito aéreo; así como de pilotos aviadores privados.

El CIAAC se mantuvo como la única institución en México dedicada a la capacitación de controladores de tránsito aéreo hasta hace 10 años.

En 1985, la Secretaría de Comunicaciones y Transportes, autoriza a Servicios a la Navegación en el Espacio Aéreo Mexicano (SENEAM), órgano descentralizado de la misma, a impartir cursos de preparación, de refresco y de promoción bajo la supervisión de la Dirección General Aeronáutica Civil (específicamente, del CIAAC).

La razón de que surgiera SENEAM como órgano de capacitación de controladores, obedece a que CIAAC no podía proporcionar la cantidad de personal necesaria para prestar el servicio de CTA. ¿Por qué? CIAAC preparaba la cantidad de personal requerida por SENEAM más un porcentaje adicional para cubrir las eventualidades; sin embargo, existía un déficit año con año que no podía cubrirse, el problema se presentaba debido a que el personal preparado provenía del Distrito Federal y al ser mandado a provincia a prestar sus servicios, desertaba al poco tiempo, por tal motivo se iniciaron cursos en provincia en las áreas donde se requería el personal; así, el personal capacitado, era originario del lugar donde se capacitaba.

PLAN DE CURSOS

Los cursos de preparación de SENEAM abarcan un total de 19 meses, al final de los cuales los estudiantes terminan capacitados como controladores de tránsito aéreo para aeródromo, aproximación y área.

La capacitación comprende un total de 32 materias (1 305 horas de instrucción):

- Matemáticas aplicadas
- Física Aplicada
- Ortografía Básica y Redacción - Teoría y Práctica
- Inglés Técnico Parte I

- Aerodinámica Básica
- Medicina de Aviación
- Navegación Aérea
- Relaciones Humanas
- Principios de Computación - Teoría y Práctica
- Comunicaciones Aeronáuticas
- Servicios de Tránsito Aéreo
- Reglamento del Aire
- Ley de Vías Generales de Comunicación y sus Reglamentos
- Características de Aeronaves
- Meteorología General - Teoría
- Climatología Básica Clase IV - Teoría y Práctica
- Técnicas de observación Sensorial - Teoría y Práctica
- Instrumentos Meteorológicos - Teoría y Práctica
- Operación de Instrumentos - Teoría y Práctica
- Codificación de Informes - Teoría y Práctica
- Observación Aerológica - Teoría y Práctica
- Operaciones Aeronáuticas
- Procedimientos de Control - Teoría de Aeródromo (TWR-T)
- Procedimientos de Control - Laboratorio de Aeródromo (TWR-L)
- Inglés Técnico Parte II
- Publicación de Información Aeronáutica (Manual PIA)
- Procedimientos de Control - Teoría de Aproximación (APP-T)

- Procedimientos de Control - Laboratorio de Aproximación (APP-L)
- Entrenador Sintético de Vuelo por Instrumentos - Práctica
- Procedimientos de Control - Teoría de Área (ACC-T)
- Procedimientos de Control - Laboratorio de Área (ACC-L), (no radar)
- Prácticas en Torre de Control

Teóricamente, un controlador recién egresado, puede llegar a un aeropuerto y desempeñar el puesto de control de aeródromo, a una dependencia de control de aproximación y prestar su servicio; o a un centro de control de área, para desempeñarse como controlador de área.

LOS CONTROLADORES DE TRÁNSITO AÉREO EN LA PRÁCTICA

Ninguno de los servicios de CTA que prestan los controladores tiene mayor o menor responsabilidad, pero difieren en el número de aeronaves que se controlan por hora, día y aún en diferentes horas de un mismo día. Por tal razón, un controlador recién egresado se le es asignado el puesto de controlador de aeródromo, para que con el tiempo sea ascendido de categoría y emolumentos a controlador de aproximación; así, el controlador de aproximación ascenderá a controlador de área.

En algunos aeropuertos, los menos (cuatro) existe la posibilidad de lograr el desarrollo completo de un controlador, pues tiene las tres opciones de control. En otros aeropuertos (veintisiete) existe la posibilidad de lograr el desarrollo hasta el control de aproximación y en los aeropuertos restantes (cincuenta y tres) no hay posibilidad alguna de ascenso, por no haber más que el servicio de control de aeródromo. Es por esto, que las posibilidades de un ascenso rápido son muy pocas y en ésta, como otras actividades, lo que no se practica se va olvidando o la habilidad en la práctica disminuye.

Así, un controlador de aeródromo que desee o sea requerido para ocupar el puesto de control de aproximación, necesita ser nuevamente capacitado para que ocupe dicho puesto eficientemente.

Actualmente, SENEAM cubre estos requerimientos de diversas maneras, como el reunir a los aspirantes al ascenso en un lugar específico y conjunta a un grupo de instructores (cuando menos dos, si estos pueden impartir de 2 a 3 materias diferentes o en su defecto, cuatro instructores). El movilizar a este personal, implica un gasto muy alto pues temporalmente todo este personal debe ser cubierto mediante jornadas y pagos extra en sus lugares de origen, además, son otorgados viáticos en el sitio del curso.

3

INTELIGENCIA ARTIFICIAL Y SISTEMAS EXPERTOS

El campo de la Inteligencia Artificial ha progresado de una disciplina académica a una tecnología comercial en un tiempo relativamente corto. Particularmente una de sus áreas, los Sistemas Expertos, ha generado simpatizantes debido a que ésta ofrece la oportunidad de organizar la experiencia humana dentro de una representación que puede ser manipulada por la computadora.

INTELIGENCIA ARTIFICIAL (IA)

La investigación de la IA se define generalmente como la búsqueda de modelos computacionales generales para emular la inteligencia humana. Desafortunadamente, nadie conoce realmente qué es la inteligencia o cómo puede uno estar seguro de que alguien se está comportando inteligentemente. El nombre que se le ha dado a esta disciplina ha desatado polémicas, aumentadas por el hecho de que se ocupe el término "inteligencia artificial" para promover ciertos aparatos electrodomésticos como las videocaseteras.

Con el transcurso del tiempo, se han delineado varias corrientes de enfoque de la IA, principalmente:

- La IA como disciplina que intenta reproducir el comportamiento inteligente que se aprecia en la naturaleza. Las disciplinas ligadas fuertemente a este enfoque son la neurología y la psicología, entre otras.
- La IA como disciplina de desarrollo de técnicas para resolver problemas prácticos con un enfoque ingenieril.

Sin embargo, las corrientes de la IA no están claramente separadas y es común encontrar trabajos de IA que manejan ambas corrientes.

La investigación básica, concierne áreas de comportamiento inteligente como el entendimiento del lenguaje, razonamiento, aprendizaje, etc., y no se limita al hombre.

Así pues, el término de Inteligencia Artificial es usado para referirse a toda una clase de problemas dignos de solución dentro de las áreas mencionadas anteriormente. Y, bajo el punto de vista de la ingeniería, lo importante no es la inteligencia en general, sino la forma en que ciertas técnicas computacionales, desarrolladas en el medio de la IA, pueden resolver problemas reales.

DEFINICIÓN

La Inteligencia Artificial estudia el comportamiento inteligente, dedicándose principalmente al razonamiento lógico que utiliza el hombre para adquirir conocimientos. La IA trata de simular el razonamiento mediante operaciones lógicas (aplicación de la lógica formal⁴) para la toma de decisiones y la retroalimentación, tal como lo haría un ser humano.

La IA se aplica comúnmente en el diseño de programas que proporcionan soluciones mediante la inferencia de conocimientos o que imitan la forma de tomar decisiones del hombre en un área en particular (uso de máquinas para manipular informaciones simbólicas).

Una de las primeras definiciones, por estudiosos del tema en el año de 1956, es citada a continuación:

*"La Inteligencia Artificial es el conjunto de técnicas que se aplican en el diseño de programa para computador que tengan capacidad de razonar, en el sentido de inferir nueva información, y que por la dificultad del problema a resolver requieren una solución con un grado de inteligencia."*⁵

TIPOS DE EXPERIENCIA HUMANA QUE MANEJA LA IA

Hace varias décadas, el psiquiatra suizo Carl Jung, describió una manera de clasificar los tipos de conocimiento o de experiencia humana y la manera en cómo expresamos esa experiencia. Las características generales de estas

⁴ En la antigüedad, los griegos inventaron la lógica formal para manipular símbolos de tal manera que el resultado pudiera predecir el estado real del mundo. La lógica estudia la estructuración del razonamiento válido mediante el cual, a partir de premisas verdaderas, se obtienen conclusiones verdaderas.

⁵ Definición por John McCarthy, Marvin Minsky, Allen Newell y Herbert Simon entre otros en la Conferencia de Dartmouth de 1956. Citada por R. Galán L.

experiencias pueden describir muy bien los tipos de conocimientos que se han deseado representar en la computadora.

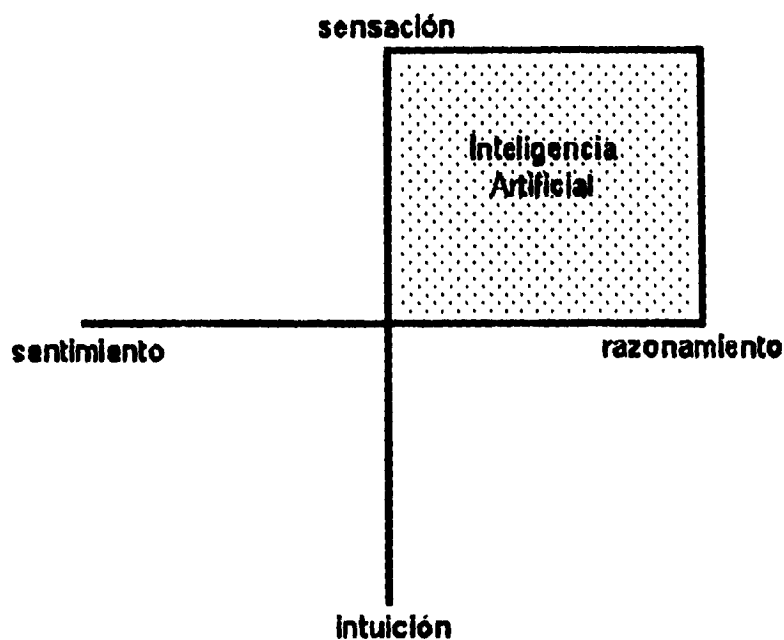


Figura 3 - 1 Tipos de Experiencia Humana

En la figura 3-1 se muestran los tipos de experiencia humana donde la función sensitiva está implementada por nuestros cinco sentidos, los cuales usamos para percibir eventos en el mundo físico. La función de razonamiento organiza las entradas sensoriales de acuerdo a uno o más paradigmas lógico-deductivos. Esta información puede ser pasada a la función de los sentimientos cuya tarea es poner en juicio la información para decidir si es correcta o incorrecta, buena o mala; por otra parte, la función intuitiva extiende la experiencia pasada permitiendo mirar la situación presente para saber qué puede presentarse en el futuro.

La IA está limitada a la percepción de entradas físicas y a su procesamiento mediante un proceso de razonamiento lógico-deductivo.

HISTORIA

Las inquietudes respecto a la IA comenzaron antes de los años cincuenta, pero fue en esa década cuando se comenzó a tratar formalmente este campo de estudio, pues comenzaban a realizarse programas basados en símbolos⁶ (llamados de tipo simbólico) y a desarrollarse las Redes Neuronales, cuyo inicio surgió del deseo de construir un cerebro electrónico.

El hecho de que la psicología y la lógica formal se establecieran como ciencias antes de la Segunda Guerra Mundial fue un punto clave para el desarrollo de la IA, pues a partir de 1945 los científicos trataron estos temas como la inquietud de añadir este tipo de características a las computadoras. Con el desarrollo de la Psicología del Procesamiento de Información, se inicia la investigación de la IA como tal (década de los años sesenta).

A principios de la década de los años cincuenta se publicó la revista *Computing Machinery and Intelligence* por A. M. Turing, quien planteaba una prueba para el reconocimiento de un comportamiento inteligente, basada en que al poner a una persona a dialogar con una computadora (sistema bajo estudio) sin saber con quién dialoga, tuviera la sensación de estar en contacto con un interlocutor desconocido.

En 1955 se desarrolla el primer lenguaje de procesamiento de listas, IPL II (*Information Processing Language II*) en la Carnegie Mellow University, tomado como el primer lenguaje de IA. Sus autores: J. C. Shaw, Alled Newell, y Herbert Simon (premio Nobel). Estos dos últimos publican a principios de los años sesenta el libro *Human Problem Solving*.

⁶ Para entender lo que es un símbolo, tomemos el ejemplo de la escritura: La escritura es un conjunto de símbolos. Primeramente tenemos a las letras, que son símbolos que representan ciertos sonidos y forman parte de las palabras que representan el habla, las palabras, a su vez, son símbolos de objetos en el mundo real.

A mediados de 1956, año en que es considerado el nacimiento de la IA, se establece la primera definición de Inteligencia Artificial en la conferencia de Dartmouth sobre IA (en el Dartmouth College en Hanover, New Hampshire) que en algunos libros se le atribuye a John McCarthy, del mismo colegio. Y es McCarthy, inspirado en el trabajo de Simon, quien en 1958 elabora el que sería el lenguaje dedicado a implementaciones de IA más popular en Estados Unidos: LISP.

Entre 1961 y 1970 comienza el desarrollo de la IA en la solución general de problemas. En 1969 se realiza el primer Congreso Internacional de Inteligencia Artificial y en 1970 se publica la revista *Artificial Intelligence*. En ese año comienzan a desarrollarse proyectos de investigación hasta finales de década. Las actividades de la IA se distinguieron por estar vinculadas a aspectos tales como la traducción automática de textos de un idioma a otro, resolución general de problemas y la percepción visual.

A partir de 1970, se desarrolla el lenguaje Prolog (*PROgramming in LOGic*) por Alain Colmera y sus colegas de la Universidad de Marsella, cuyas bases fueron planteadas en 1965 por J. Alan Robinson, convirtiéndose en el lenguaje de programación utilizado por los estudiosos de la IA en Europa. A finales de ese año, salen versiones de Prolog para microcomputadora. La versión más popular fue micro-Prolog. El proyecto japonés de la quinta generación está basado en este lenguaje.

Hasta 1981, es cuando empiezan a desarrollarse proyectos a nivel comercial.

La IA comprende varios temas de investigación que se han "abierto" en diferentes épocas, dado que los avances logrados en ella dependen grandemente de la evolución de las computadoras y la electrónica. Entre los primeros temas que marcan la historia de la IA, por el impulso que le han dado a ésta, se encuentran:⁷

⁷ Foreyth 1986. Citado por J.P. Sánchez

- Redes Neuronales (1950)
- Búsqueda Heurística (1960)
- Sistemas Expertos (1970)
- Aprendizaje de las Máquinas (1980)

ÁREAS DE APLICACIÓN Y DESARROLLO

La IA, en su evolución, ha encontrado diversas áreas de aplicación. Así como un ser vivo percibe, procesa e identifica patrones, resuelve problemas y controla sus acciones; ha sido deseable el lograr que una máquina posea esas cualidades. Esto ha llevado a la IA a expandirse en diversas áreas, como son:

- Lógica Difusa
- Algoritmos Genéticos
- Reconocimiento de Patrones y Visión Artificial
- Sistemas Expertos
- Robótica
- Redes Neuronales
- Lenguaje Natural

En la actualidad aún siguen programándose en lenguajes, tales como FORTRAN, C, PASCAL, y orientando estos programas hacia el hardware tradicional. Estas herramientas no fueron elaboradas para la simulación de un comportamiento inteligente y por lo mismo, no se obtiene, mediante ellas, todo el provecho que se pudiera de una computadora.

La diferencia entre lenguajes dedicados a la IA y los tradicionales, es que en estos últimos, los programas trabajan únicamente con datos, mismos que son procesados mediante métodos algorítmicos. En lenguajes de programación para la IA, los programas procesan conocimientos mediante ciertos métodos de búsqueda, aunque también pueden procesar datos.

La investigación en el campo de la IA no le concierne únicamente a la informática, pretender esto es cerrarse las puertas hacia el entendimiento del comportamiento humano dirigido a la solución de problemas. Así pues, es necesario que diversas disciplinas como la filosofía, psicología, lingüística y, desde luego, la informática interaccionen para el desarrollo de la IA.

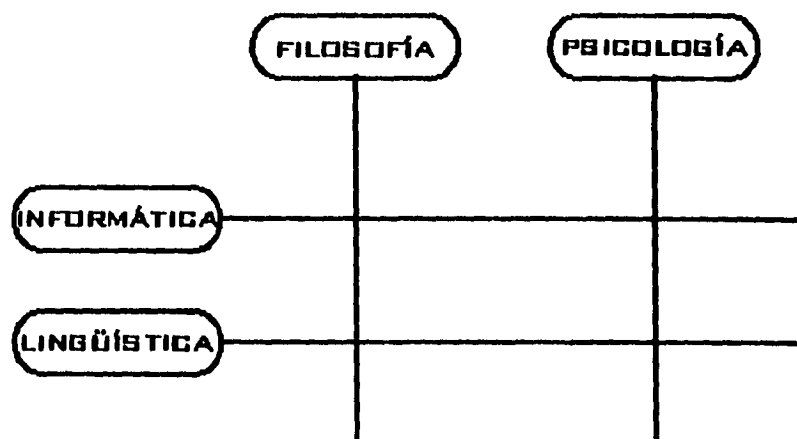


Figura 3 - 2 Interacción entre las áreas de estudio de la IA

SISTEMAS EXPERTOS (SE)

El área de los SE investigan los métodos y técnicas que hacen posible la construcción de sistemas especializados en dar soluciones expertas a problemas prácticos específicos. Por lo tanto, todo SE comienza con la percep-

ción de un problema que es difícil de resolver usando los métodos de computación tradicionales.

Un SE es solamente una extensión de un arreglo de herramientas que provee una computadora a un usuario específico.

DEFINICIÓN

Los Sistemas Expertos⁸ (SE) son programas dedicados a imitar el comportamiento humano con el fin de sustituir a un experto humano o especialista; para esto, almacenan conocimientos muy específicos de expertos en un área en particular, utilizan técnicas de la IA de manera que pueden solucionar un problema por medio de deducciones lógicas y son capaces de explicar y justificar sus conclusiones. De esta manera, podemos contar con un "experto" siempre a la mano, o bien permitir que el experto humano se dedique a labores más importantes, mientras que el SE realiza tareas un tanto rutinarias pero que, sin lugar a dudas, requieren del conocimiento de un especialista.

Por lo que:

*"Los Sistemas Expertos son sistemas computarizados que proporcionan soluciones a problemas en temas específicos, debido a que incorporan, en una base de conocimientos, conocimiento extraído de expertos humanos, del cual obtienen conclusiones válidas mediante mecanismos de inferencia"*⁹

⁸ En algunos libros se les denomina como "Sistemas Basados en el Conocimiento" o *Knowledge Based Systems* pues no son capaces de actualizar sus conocimientos mediante la práctica, como lo haría un experto humano.

⁹ Lara Rosano, 1991

HISTORIA

En 1951, John McCarthy propuso un algoritmo, denominado alfa-beta, que permitía abreviar los árboles explorados por programas de la IA. Su fracaso se debió a que se creía en la posibilidad de elaborar un programa de solución general de problemas (GPS, por sus siglas en inglés), sin la utilización adecuada de conocimientos de especialistas en los temas tratados; por lo que algunos investigadores cambiaron el enfoque de ese problema, restringiendo sus ambiciones a un dominio específico e intentando imitar el comportamiento de un humano con amplia experiencia en el dominio, dando así paso al nacimiento de los SE.

En 1974, se crea la teoría que concibe a los SE y se desarrollan las computadoras, así como los lenguajes de programación, siendo el IPL el primer lenguaje de tipo simbólico dedicado a implementar programas de IA. Posteriormente (1958-1981) surgen, cronológicamente: LISP, Prolog, SmallTalk (Versión inicial, 72, por Alan Kay del Palo Alto Research Center de Xerox, realizada a partir de LISP y Plasma. Comercializado a partir de la versión 80) y OPS (*Official Production System*, creado por C. L. Forgy de la Universidad de Carnegie Mellon a mitad de los años setenta).

La década de los setenta se caracterizó por la elaboración de SE; al principio trataban de resolver problemas generales pero esto los convertía en programas débiles, debían entonces ser programas para problemas específicos o delimitados. Los primeros sistemas tenían alcances muy limitados y eran dedicados a temas académicos muy idealizados, entre ellos se encuentran:

- DENDRAL; dedicado a deducir información analizando los datos del espectrógrafo de masas sobre estructuras químicas de compuestos orgánicos, utilizando reglas de producción como forma de representación del conocimiento. El hecho de que este sistema poseyera una gran cantidad de conocimientos especializados integrados de una forma clásica, traía grandes problemas al momento de modificar

su base de conocimientos. Elaborado por la Universidad de Stanford en 1965-1967 por Feigenbaum, Buchanan y Lederberg.

- MYCIN; realiza diagnósticos médicos para enfermedades infecciosas de la sangre, por medio de la introducción de los síntomas del paciente, su conocimiento comprende aproximadamente de 400 reglas que relacionan posibles condiciones a interpretaciones asociadas. Iniciado por Edward A. Feigenbaum y desarrollado por E. Shortliffe y sus colaboradores en 1972-1976 en Stanford.
- XCON; SE desarrollado por la Digital Equipment Corporation para configuraciones de redes de computadoras VAX.
- PROSPECTOR; dirigido hacia la exploración mineral y herramientas de identificación. Incluye una base de conocimientos para cada tipo de depósito mineral.
- El sistema se caracteriza por utilizar una combinación de reglas de producción y redes semánticas para representar el conocimiento. Por Duda, Gaschnig y Hart en Stanford Research Inst.¹⁰
- CADUCEUS; originalmente llamado INTERNIST, herramienta de diagnóstico para medicina interna. Su conocimiento de medicina interna es mayor al de cualquier humano, pudiendo diagnosticar casos complejos, consiste en una extensa red semántica de interrelaciones entre enfermedades y síntomas. Por Pople, Myers y Miller en la Carnegie-Mellon University, Pittsburgh 1975.
- R1; configura sistemas de computadoras VAX y PDP para la DEC (Digital Equipment Corporation), con el cual la empresa estima que ahorra 70 millones de dólares anuales. Por Carnegie-Mellon, 1978.

Estos sistemas dieron lugar a que algunos investigadores (Obrow y Winograd, Minsky, Brachman)¹¹, intentaran desarrollar una teoría general de

¹⁰ El año de desarrollo es marcado como 1972 en algunos libros y como 1979 en otros.

¹¹ Mencionados por Chatain y Dussauchoy.

representación del conocimiento con resultados limitados, dado que el conocimiento es demasiado amplio para ser explotado por un solo formalismo, se dice,

"La potencia de un SE procede del conocimiento que posee y no de un formalismo en particular ni del motor de inferencia que utilice".¹²

En esa década también se construyen las primeras computadoras simbólicas en Estados Unidos (Proyecto DARPA, 1973).

En los ochenta comienzan a divulgarse lenguajes especializados, herramientas y shells, de los cuales se hablará posteriormente; incrementándose cada vez más el desarrollo de los SE con gran variedad de aplicaciones, por lo que en aquella época se aguardaban los años noventa como la década de la difusión generalizada de esta técnica.

Con el nacimiento de los SE, se plantea la necesidad de una nueva rama de estudio: la Ingeniería del Conocimiento como parte de la IA.

¹² Felgenbaum en la *International Joint Conference on Artificial Intelligence*, 1977. Citado por Chatain y Dussauchoy.

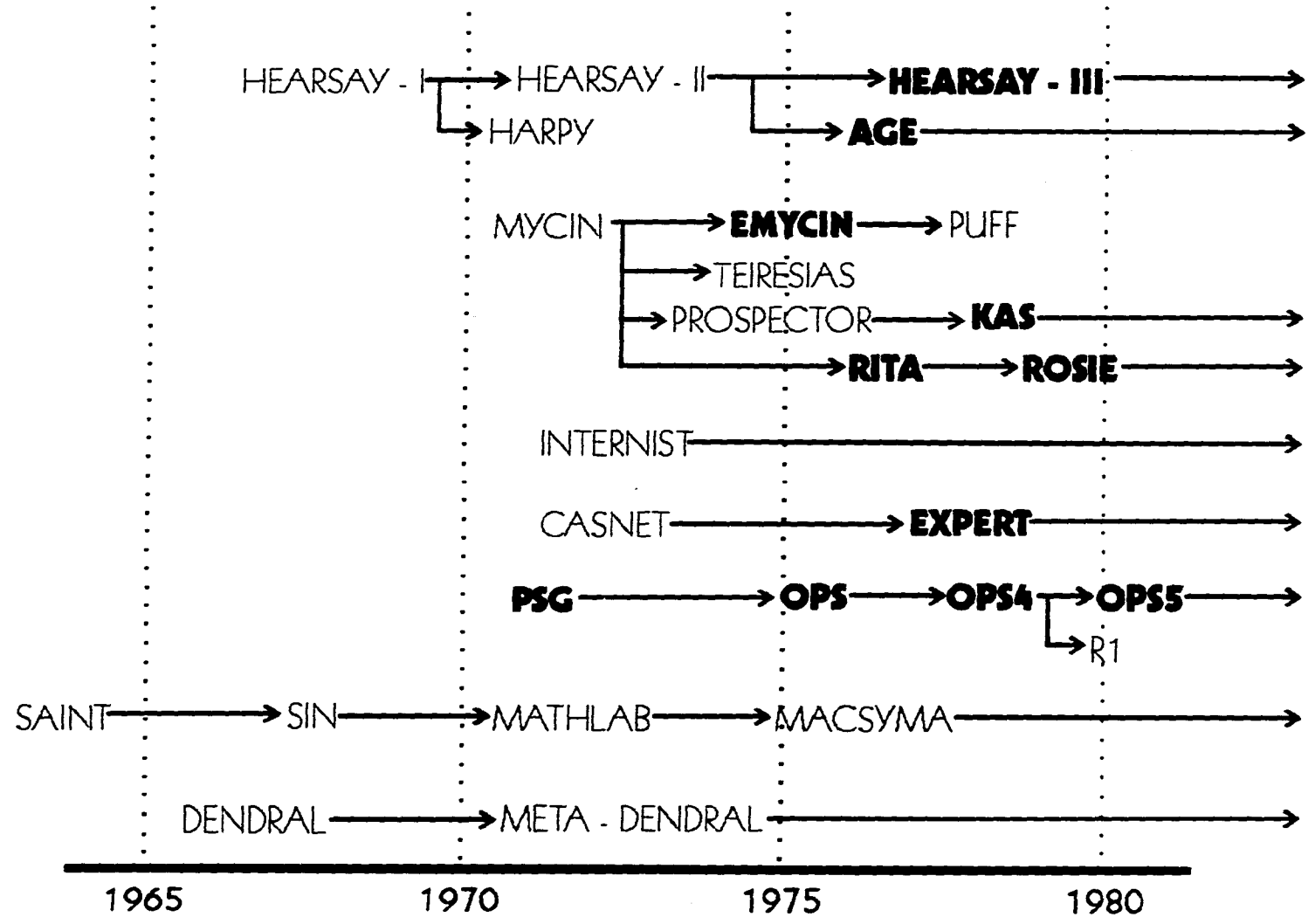


Figura 3 - 3 Evolución de SE y shells (los shells aparecen con negritas)

TIPOS DE SISTEMAS EXPERTOS

Según el problema que vayan a realizar, se puede clasificar a los SE según la labor que realizan:

- a) **Interpretación.** A partir de datos de entrada obtenidos, ya sea por medio de sensores o introducidos por el usuario terminal, dan una explicación a estados y situaciones que engloban los datos, asignando posteriormente un significado simbólico para describirlos.

Existen dos tipos de interpretación: la *síntesis* que se basa en la combinación de los datos para obtener un resultado y, el *análisis* que utiliza la separación de los datos para dar una interpretación.

- b) **Predicción.** Infieren probables consecuencias de situaciones dadas.
- c) **Diagnóstico.** Infieren las causas que provocan un mal funcionamiento, basándose en los síntomas consecuentes del problema. Seleccionan, según su construcción, el mejor diagnóstico de los que se tienen ya definidos o lo construyen.
- d) **Diseño.** Elaboran configuraciones, modificándolas cíclicamente hasta obtener el diseño final, que resuelve un problema con ciertas restricciones.
- e) **Planeación.** Utilizando ciertos recursos y sujetándose a restricciones, infieren acciones estrechamente relacionadas para lograr un objetivo.
- f) **Monitoreo.** Analizan el comportamiento de un sistema, comparando continuamente datos de entrada con los estándares para detectar anomalías e informar del estado del sistema.
- g) **Reparación.** Desarrollan y ejecutan (de ser posible) las acciones correctoras óptimas necesarias para eliminar fallas del área específica para la cual se elabore el SE. Es necesario que estos SE cuen-

ten con un diagnóstico, ya sea que lo elaboren especialistas, otros Sistemas Expertos o el propio SE.

- h) **Formación o Instrucción.** Capacitan en una rama específica, evalúan a cada alumno y dan orientación e información personalizada.

Cuando el usuario es un alumno, busca desarrollar pericia personal, de la manera más rápida, relacionada con el área específica mediante la recuperación de conocimientos organizados y condensados del sistema.

Este tipo de SE incorporan diagnósticos y depuran subsistemas que específicamente dirigen al estudiante como el sistema de interés. Típicamente, estos sistemas comienzan con la construcción de una descripción hipotética del conocimiento del estudiante que interpreta su conducta, posteriormente, diagnostican las deficiencias en el conocimiento del estudiante e identifican un remedio apropiado, que podría ser la aplicación de un tutorial para transmitir el conocimiento al estudiante.

- i) **Control.** Guían un proceso mediante la interpretación de una situación dada comparándola con la norma establecida para detectar las desviaciones ocurridas (diagnóstico), formulan un plan de corrección y monitorean su ejecución.
- j) **Depuración.** Muestran posibles soluciones para los casos en que programas de computadora o equipos cualquiera muestran condiciones anómalas a su comportamiento.

APLICACIONES Y VENTAJAS

Con la descripción anterior de los tipos de SE, es fácil ver que la aplicación de éstos es muy extensa. Sin embargo, es necesario cumplir con ciertos

requisitos para elaborar un SE, pues puede presentarse el caso de que con un programa tradicional se resuelva el problema.

Así, la aplicación de un SE será adecuada si la solución al problema no está previamente fijada, es decir, la solución depende del razonamiento del SE de acuerdo a las diferentes circunstancias en las que se encuentre y además:

- El problema no es reducible a un algoritmo numérico, ya que si no, no merecería la pena construir el SE.
- El problema tiene perfectamente definido su dominio y la nomenclatura para su descripción, está ya definida.
- Existen expertos humanos que orientarán al ingeniero del conocimiento en la elaboración del sistema experto.

Los sectores en los cuales pueden elaborarse aplicaciones de SE son: industria, comercio, servicios, actividades gubernamentales y en el sector financiero.

Cualidades de los SE

Todos los SE tienen ciertas características en común, pero no se puede decir que tengan esas características en la misma proporción. La dimensión que adquieran estas cualidades hacen especial a cada SE.

- Grado de especialidad. La meta más importante en el trabajo del SE es dar el alto nivel de desempeño que puede dar un humano especializado en el área. Entre más actúe como el especialista, un SE producirá alta calidad de resultados en un tiempo mínimo.
- Manipulación de símbolos. Los SE trabajan con cierto tipo de información con la cual razonan. Específicamente, los SE emplean un ra-

zonamiento simbólico, por esta razón, la representación del conocimiento es uno de los puntos más importantes para la creación de un programa.

- **Inteligencia.** Esta cualidad sigue del énfasis al razonamiento que un SE debe exhibir. Un SE puede ser más o menos inteligente dependiendo del alcance de sus principios básicos y de la calidad de su proceso de razonamiento de propósito general.
- **Complejidad o dificultad.** El razonamiento requerido para resolver problemas en un área en particular puede, no envolver bastantes pasos o bastantes alternativas en un punto de bifurcación.
- (Los problemas que se complican demasiado requieren de una persona dedicada a esa área en particular).
- **Reformulación.** Una de las tareas críticas que realizan los especialistas y que algunos SE realizan aproximadamente, es el tomar inicialmente un problema de una forma arbitraria y convertirla en una forma apropiada para que sea procesada por reglas especializadas. El problema debe ser tal que sea razonablemente manipulable por un especialista, su forma puede ser simplemente una colección de datos con la demanda de encontrar patrones apropiados y actuar de acuerdo al área específica que cubre el experto humano.
- **Razonamiento acerca de sí mismo.** Una de las habilidades requeridas en un SE es el permitir el razonamiento acerca de su propio proceso de razonamiento. Esta habilidad puede no ser una parte necesaria en algunos tipos de sistemas.
- (El sistema observa su razonamiento como si estuviera afuera, asocia principios fundamentales apropiados con cada regla experta).
- **Tipo de tarea.** Las diferentes tareas que se pueden realizar por medio de los SE pueden cambiar el diseño de la arquitectura del SE general (cuyo esquema se presenta en este capítulo). La dimensión de un SE variará dependiendo del tipo de tarea que realice.

SE vs. expertos humanos

Un experto humano es aquella persona que resuelve problemas en un área específica (especialista). Sus conocimientos crecen en profundidad pues saben más de una sola cosa y menos de otras. Las soluciones que obtiene ya no son pensadas, las infiere de manera tan natural para él que no se percata del procedimiento y a una persona común le resulta incomprensible el comportamiento del experto.

Los conocimientos que tienen han sido adquiridos a través de un largo período de tiempo, motivo por el cual son caros y escasos. Siempre se llevan con ellos los conocimientos.

Por esto,

- Al tener un SE completo éste se convierte en un sistema autónomo, ya que no es necesario tener al experto humano para tomar decisiones; por lo consiguiente, los gastos que genera una persona, en cuanto a honorarios e incluso viáticos, se eliminan.
- Se puede reproducir el SE, como expertos se necesitan y llevarlos a cualquier lugar pues las necesidades de hardware se limitan a un equipo de PC.
- Un SE no necesita descanso, por lo cual siempre está disponible para realizar su trabajo y no tiene distracciones que lo atrasen.
- Un SE puede poseer grandes cantidades de información, la cual puede provenir de más de un experto, haciéndolo más efectivo.
- Al tener un SE trabajando en el lugar de un experto, se aumenta la productividad del conocimiento del especialista, pues se releva a éste de tareas repetitivas, pudiendo así aprovechar su capacidad intelectual en labores que requieren de creatividad.

SE vs. programas tradicionales

Un programa tradicional, basado en datos numéricos, corre en forma secuencial y necesita de conocimientos precisos que raramente se modifican. La solución a la que llegan se basa en la combinatoria y es óptima. Este tipo de programas no pueden explicar ni justificar los pasos que siguen, ni la solución a la que llegan; y, por último, no poseen la capacidad de realizar pequeños prototipos rápidamente.

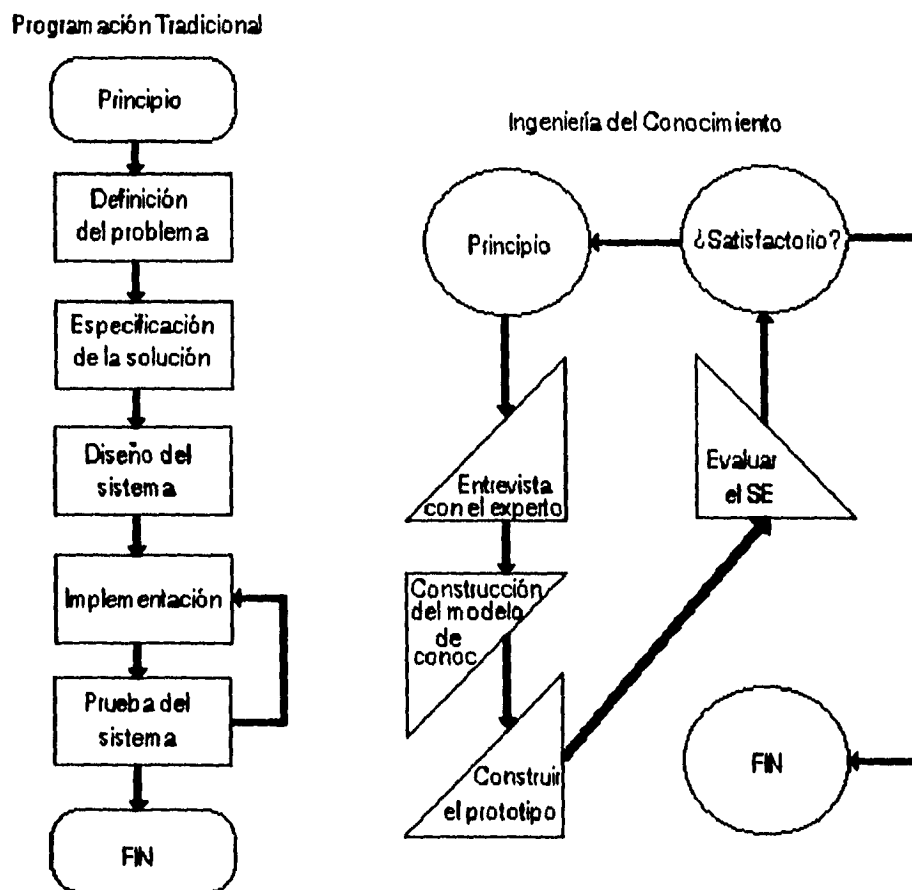


Figura 3 - 4 Programación convencional y simbólica

En contraste con los sistemas tradicionales de procesamiento de datos, las aplicaciones de la IA envuelven, generalmente, características distintivas como: representación simbólica, inferencia simbólica y búsqueda heurística.

ESTRUCTURA GENERAL DE UN SE

Un SE se caracteriza por tener en su interior gran capacidad de conocimientos dirigidos hacia un área en particular, técnicas de búsqueda, capacidad para inferir nuevos conocimientos, a partir de los ya existentes, procesamiento de tipo simbólico y habilidad para explicar su razonamiento.

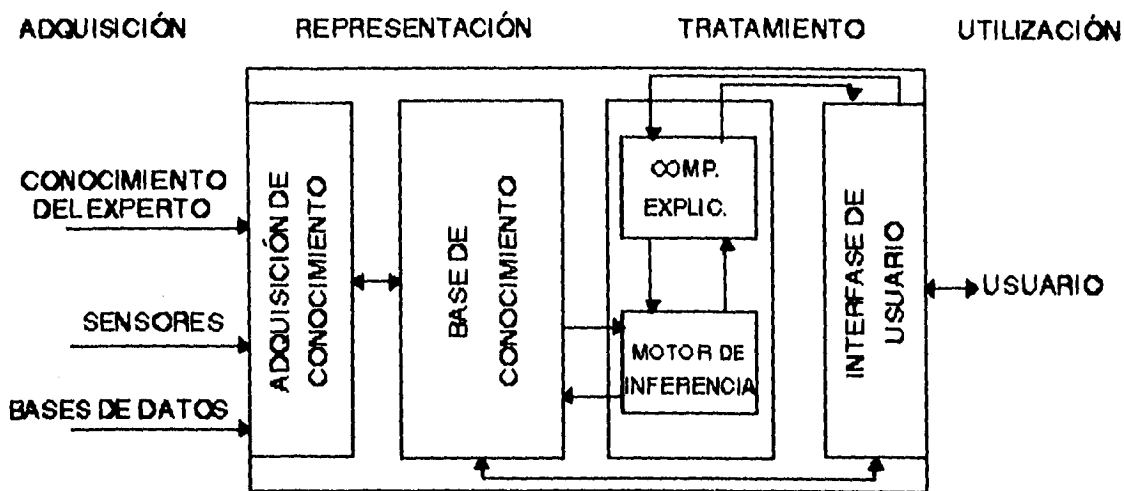


Figura 3 - 5 Estructura de un SE

La composición de un SE varía dependiendo de la aplicación que se va a realizar. Los elementos que lo constituyen deben ser independientes; entre los más comunes están:

- Base de Conocimientos
- Motor de Inferencia
- Adquisición de Conocimientos

- Componente para la Explicación
- Interfaz del Usuario

Base de Conocimientos

La base de conocimientos es la parte especializada del sistema y es el conjunto de conocimientos reunidos por el Ingeniero del Conocimiento; en el que se incluye la experiencia del experto humano orientada hacia un área en particular, en otras palabras, es el medio a través del cual el conocimiento de un experto se establece en cierto formato para estar disponible para la computadora. Los elementos del conjunto de conocimientos son estructurados de tal forma, que puedan ser interpretados por el motor de inferencia, cabe remarcar que son independientes de éste, y así el sistema puede trabajar como un experto. Entre estos elementos se encuentran: objetos, relaciones entre ellos, hechos y reglas con las que se procesa la información y se crea nueva a partir de la original (heurística).

La forma en que se represente el conocimiento en la base de conocimientos debe ser sencilla, de tal modo que sea manipulable y que ocupe poco espacio. Es deseable que se encuentre en módulos y que estos sean independientes para que al quitar uno de ellos no afecte al resto y se pueda así modificar la base de conocimientos. Otra característica que deben tener los módulos es que sean relacionales para que se establezcan vínculos entre los conocimientos. La base de conocimientos puede asociarse a una memoria permanente.

El proceso de construcción de la base de conocimientos comienza y continúa observando a los expertos hacer decisiones y filtrando fuera de ese estilo, los factores que van dentro de la toma de decisiones.

Motor de Inferencia

Representa el núcleo del SE y su construcción no requiere de los conocimientos de un experto. Es llamado también Mecanismo de Inferencia o Estructura de Control. Como lo sugiere su nombre, infiere nuevos conocimientos a partir de la información ya existente. Por esto, la selección de la información apropiada para obtener nuevos conocimientos, es el punto más importante del proceso que lleva a cabo el SE.

El motor de inferencia es en sí un programa lógico que sigue cierta estrategia para producir nuevos conocimientos (en él se construye la forma de razonamiento); dicha estrategia se basa en el razonamiento que sigue el experto del área que se está tratando y cuyo conocimiento ya está contenido en la base de conocimientos.

El motor de inferencia tiene diferentes formas de evaluar el conocimiento, de tal manera que se le puede catalogar en:

- Determinístico, cuando las condiciones que se introducen afectan totalmente al resultado.
- Probabilístico, cuando la probabilidad de las hipótesis depende de las probabilidades de los hechos que las sostienen.
- Aproximado, cuando la certeza de los resultados está en función de la certeza de los hechos.
- De analogía, cuando un hecho adquiere propiedades de otro por su similitud.
- De herencia, cuando un hecho adquiere las propiedades de su antecesor.

El motor de inferencia tiene dos estrategias fundamentales de búsqueda: el encadenamiento hacia atrás (retroencadenamiento) y el encade-

namiento hacia adelante (encadenamiento frontal), de los cuales se hablará más adelante.

Adquisición de Conocimientos

Las tareas inteligentes que realiza un humano, envuelven el uso hábil de una gran cantidad de experiencia obtenida de la vida diaria, y aprendiendo de estas experiencias, que generalmente ocurre de manera inconsciente.

Nosotros aprendemos a hacer ciertas tareas intentándolas (actuando), es decir, a prueba y error. En muchos casos el conocimiento que obtenemos de estas experiencias no es firme, pero algunas de estas experiencias están en forma de reglas que almacenamos en algún lugar para usarlas cuando una situación similar se presente. Estas reglas son llamadas heurísticas y ellas representan el conocimiento que tratamos de almacenar para ser usado por el SE.

Al proceso de obtener tales heurísticas de una persona (generalmente un experto) y almacenarlas en la computadora, se llama adquisición de conocimiento.

Este componente del SE permite al experto o al ingeniero del conocimiento la introducción de conocimientos, ya sea mediante de transductores, bases de datos o el intercambio dinámico de datos con otro software.

En muchas áreas, con el paso del tiempo, la información cambia y crece dejando al sistema con la misma base de conocimientos, por esto, el SE debe presentar facilidades para la actualización de conocimientos.

Es deseable que los elementos que conforman la base de conocimientos puedan introducirse fácilmente al sistema y que puedan visualizarse. También es preferible que el shell presente corrección de sintaxis al punto.

Componente para la Explicación

Proporciona los motivos por los cuales es sistema llega a una conclusión final, es decir, explica el por qué ha tomado ciertas decisiones, tal y como lo haría el experto humano. El componente se basa en la explicación o justificación de las inferencias efectuadas, pudiendo mostrar el flujo que han seguido éstas.

Este componente requiere de cierto nivel de conocimiento que proporcione el soporte a las conclusiones que proporciona el SE. Por ejemplo, supóngase que la base de conocimientos cuenta con una regla como la siguiente:

IF proposición de aeronave = operar IFR inicialmente

AND proposición de aeronave = operar VFR después

THEN autorizar hasta el fijo en que termina la parte IFR del vuelo

Y se cumplen las dos premisas de la regla, la conclusión a la que llegaría el SE requeriría de una explicación: ¿Por qué autorizar hasta el fijo en que termina la parte IFR del vuelo?, pues para el usuario puede no estar clara tal decisión; es entonces cuando se recurre a un conocimiento llamado "de soporte".

Interfaz del Usuario

Denominado también Sistema de Consulta. Es el medio mediante el cual el usuario se comunica con el SE. La interfaz del usuario debe ser "amigable", esto es, debe poder aceptar información del usuario para después convertirla en información para ser usada por el propio sistema; así como presentarle al usuario, en un lenguaje ordinario, información proveniente del sistema.

La comunicación se realiza mediante preguntas y respuestas con un formato accesible, en donde la información puede ser desplegada en forma textual o gráfica; usando pantallas de alta resolución, ventanas múltiples, iconos, color, animación, ayuda en línea, diálogos, bibliotecas, etc.; dependiendo de las necesidades habituales del usuario.

REPRESENTACIÓN DEL CONOCIMIENTO

En la vida diaria para adquirir un conocimiento se necesita obtener una representación de la realidad con la cual identifiquemos a un objeto o a un proceso entre objetos. Esta representación del conocimiento se realiza de forma simbólica, ya sea por medio de imágenes o a través del lenguaje natural del hombre.

A través del tiempo, el hombre ha encontrado diversas formas de representar su realidad, ya sea por medio de modelos matemáticos o por medio de la determinación de la veracidad de conclusiones obtenidas de axiomas o hechos, esto es, por medio de la lógica.

La obtención del conocimiento para un SE se realiza por medio de entrevistas entre el ingeniero del conocimiento y el experto, quien describe su actividad y los casos que presenta. La representación de dichos conocimientos se realiza mediante lógica de predicados, lógica difusa, lógica modal o lógica multivaluada; siguiendo:

- Reglas de Producción (*rules*)
- Redes Semánticas (*nets*)
- Frames o Marcos
- Cálculo de Predicados

Las formas en las que se represente el conocimiento deben describir a la realidad de manera fácil de tal forma que se aumente su poder expresivo y por

consiguiente disminuya su eficacia de cálculo. Para aumentar la rapidez de acceso y obtener conocimientos completos, el conocimiento tiene que estar ordenado y relacionado.

Reglas de Producción

Un tipo de regla de producción o regla de inferencia es la llamada *modus ponens* y es conocida desde el tiempo de los griegos; tiene la forma:

Si se sabe que siempre que **X** es cierto

entonces **Y** es cierto.

y se sabe que actualmente **X** es cierto

Entonces

Y es cierto.

Cada regla es una parte independiente del conocimiento que puede referirse a una relación, información o acción que dependen de ciertas restricciones. Las reglas de producción se formulan por medio del principio de causa-efecto usando una función lógica ("si..., entonces"), de modo que actúen como filtros.

Tienen como ventajas la modularidad, independencia del resto de las reglas y la sencillez con la que se expresan. Pero es difícil establecer relaciones y es probable que se introduzcan reglas repetidas o que dos de ellas se contradigan.

Esta forma de representación del conocimiento fue usada, dadas sus ventajas, en SE como: MYCIN, DENDRAL, R1.

Redes Semánticas¹³

Representan el conocimiento en forma de gráfica que permite ver las relaciones y dependencias de un área de conocimientos en particular, en donde cada nodo corresponde a un objeto y los arcos, entre nodos, muestran la relación entre ellos, ya sea de herencia o de descripción.

Las redes semánticas pueden transformarse en listas de objetos con sus relaciones que son mencionadas fuera de la red. Una red semántica no contiene información para procesarla, pero su estructura permite que el experto verifique fácilmente como ha quedado representado el conocimiento.

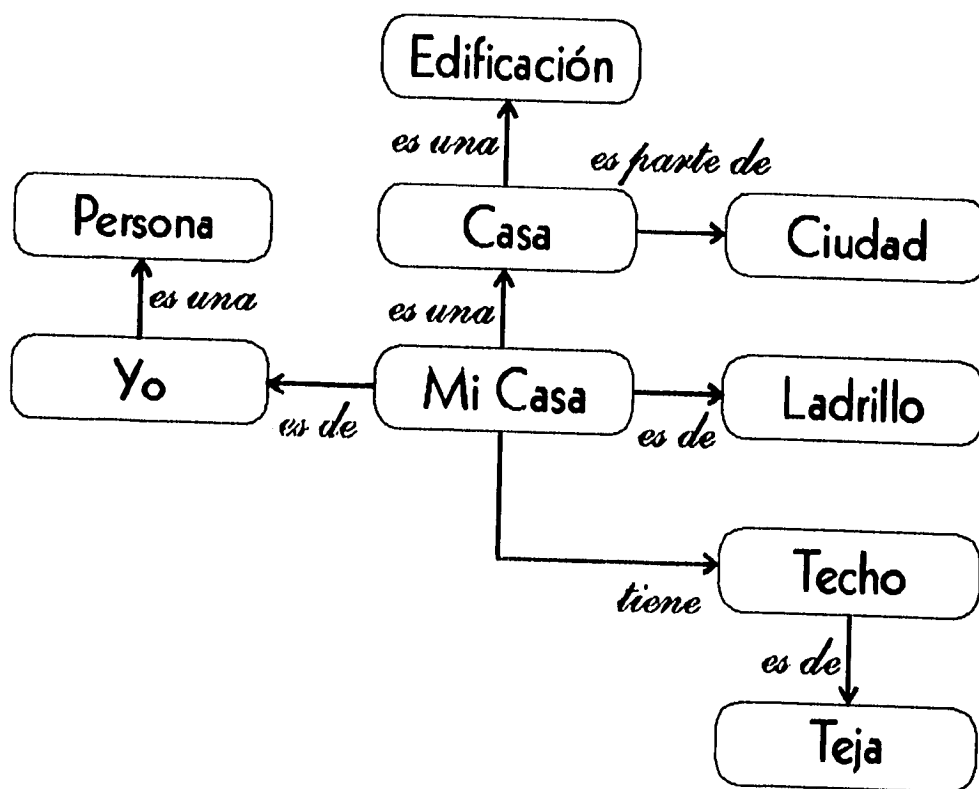


Figura 3 - 6 Red semántica del concepto "Mi Casa"

¹³ Atribuidas a Quillian, 1968

Esta representación del conocimiento fue usada en la construcción de PROSPECTOR.

Frames o Marcos

La introducción de los marcos o frames (objetos estructurados) la llevó a cabo Marvin Minsky en 1974, quien los definió como:

"... estructura de datos que sirve para representar una situación estereotipada, como estar en algún tipo de salón o ir a la fiesta de cumpleaños de un niño. Añadido a cada frame hay varios tipos de información. Parte de esta información hace referencia a cómo utilizar el frame; otra se refiere a lo que uno puede esperar de que suceda en segundo lugar. Y otra, a su vez, indica qué hacer si tales esperanzas son confirmadas."¹⁴

Los frames son modulares y reciben el conocimiento de forma declarativa. El conocimiento, ya sea referente a un objeto o a una situación, es dividido en componentes que describan procedimientos, valores, reglas o herencias que describen la realidad a representar. Los componentes son introducidos en slots o ranuras y por medio de estos se procesa al frame.

Los frames se ocupan únicamente para desarrollo de aplicaciones de la Inteligencia Artificial, en otras palabras: los frames son estructuras exclusivas de la IA.

Podemos imaginarnos a un frame como una tarjeta dividida en renglones, cada renglón representa una ranura. De tal forma, podemos escribir los frames como listas.

Dadas las facilidades que presentan los frames, comúnmente se usan en las bases de datos relacionales.

¹⁴ Citado por Dieter Nebendahl en *Sistemas Expertos, Introducción a la técnica y a la aplicación*.

Cálculo de predicados

Se refiere a la deducción lógica de los resultados por medio de la verificación de las condiciones establecidas.

La representación del conocimiento se realiza mediante enunciados que siguen una gramática y sintaxis definida (WFF, *Well Formed Formulas*), los cuales son evaluados para inferir conclusiones que crean nuevas expresiones.

El cálculo de predicados proporciona un medio natural de representar el conocimiento de manera declarativa, con el cual se pueden deducir nuevos conocimientos a partir de los ya establecidos. Este modo de representación es usado por el lenguaje Prolog.

TIPOS DE INFERENCIA

Los tipos de inferencia son usados por el motor de inferencia y se refieren al modo en el cual se realizará la búsqueda de una solución, la forma en cómo se seleccionará un camino dentro de múltiples alternativas o la forma en la cual el motor de inferencia razonará. El éxito de un SE dependerá en gran parte del tipo de inferencia que se seleccione.

Las reglas son la forma más común de almacenar el conocimiento del experto y es el encadenamiento de reglas el modo usual que se sigue para alcanzar decisiones; sus dos formas más comunes son el encadenamiento hacia adelante y el encadenamiento hacia atrás.

Encadenamiento hacia adelante

Mecanismo de búsqueda ordenada, llamado también *forward chaining* o encadenamiento frontal, cuyo proceso de razonamiento es ascendente y guiado por los datos o deductivo. Utilizado cuando se cuenta con datos bási-

cos como punto de partida; esto es, parte de los hechos para llegar a los resultados.

La búsqueda se realiza seleccionando las reglas que verifican las condiciones necesarias para llevar a cabo el efecto, es decir, se aplica la regla para la cual la condición sea cierta según los datos con los que se cuente. El proceso se sigue hasta que no existan más reglas aplicables o se encuentre la solución.

Explicándolo de otra forma, imaginemos que somos el motor de inferencia. Sabemos que la base de conocimientos está formada por una lista de reglas que usamos para tomar decisiones. Cuando usamos el encadenamiento hacia adelante queremos encontrar una solución basada en hechos, entonces nos referimos a la base de conocimientos como si no supiéramos nada y nos damos a la tarea de encontrar algo. Primeramente tenemos que tener algunos hechos acerca del problema y podemos preguntar por ellos explícitamente.

El método elemental del encadenamiento hacia adelante nos indica que comencemos de manera arbitraria con la primera regla de la base de conocimientos y tratemos de usarla, para lo cual debemos conocer como ciertas sus premisas. En algún lugar vacío inicialmente, al que llamaremos "lista de certezas", almacenamos todas las cosas que conocemos como ciertas. Cuando la certeza de una premisa no es conocida y no puede ser determinada por otra regla, preguntamos al usuario o a una base de datos para obtener tal información.

Una vez que se usa una regla, adicionamos todas sus conclusiones a la lista de certezas y vamos por la siguiente regla en la base de conocimientos, que use la primera conclusión de la regla anterior como una de sus premisas y tratamos de usarla. Esto es a lo que llamamos encadenamiento hacia adelante.

El encadenamiento hacia adelante, está basado en el "modus ponens". Se generan nuevos hechos y se tratan en profundidad o en anchura. Este tipo de inferencia activa todas las reglas aplicables aún cuando algunas de estas no sean importantes.

Encadenamiento hacia atrás

Mecanismo de búsqueda ordenada, llamado también *backward chaining* o retroencadenamiento, cuyo proceso de razonamiento es descendente y guiado por los objetivos o inductivo. Se aplica cuando el punto de partida es una hipótesis u objetivo; esto es, comprueba que una hipótesis es cierta en base a los hechos contenidos en la base de conocimientos.

Las reglas seleccionadas son las que verifican los consecuentes o efectos de acuerdo a la hipótesis inicial. Las condiciones o datos desconocidos existirán mientras existan subobjetivos por demostrar; cuando éstos hayan sido demostrados en su totalidad, se alcance la verificación de la hipótesis inicial o se termine la posibilidad de seleccionar otra regla, se habrá terminado la búsqueda.

Este tipo de encadenamiento de reglas difiere del encadenamiento frontal principalmente en que asumimos que una conclusión es cierta y usamos las reglas de la base de conocimientos para tratar de probarla, verificando la certeza de las premisas de la regla que contiene la conclusión a probar.

Para probar la certeza de una premisa, buscamos una regla que tenga esa premisa como una de sus conclusiones; si tal regla es encontrada, encadenamos hacia atrás esa regla y nos damos a la tarea de probar la certeza de las premisas de esa segunda regla de la misma forma que la inicial.

El encadenamiento hacia atrás se basa en el "modus tollens" de la lógica formal y emplea procesos recursivos. Las soluciones deben ser conocidas de antemano. Tiene como ventajas el hecho de que el árbol de búsqueda sea más corto que el del encadenamiento hacia adelante y que el sistema plantea cuestiones después de haber explorado todas las posibilidades.

Inferencia Mixta

Mecanismo de búsqueda más empleado por los SE. Busca un conjunto de soluciones mediante el encadenamiento hacia adelante que guía la búsqueda

de soluciones, y con el encadenamiento hacia atrás se verifican, de modo que se convierte en un mecanismo de dos direcciones.

Puede también utilizarse una doble búsqueda, tomando datos iniciales e hipótesis para igualar las soluciones en un punto intermedio.

En algunos casos puede ser preferible usar la búsqueda de dos direcciones, empleando encadenamiento hacia adelante, a partir de estado inicial y simultáneamente, usar encadenamiento hacia atrás, a partir de los estados objetivos. El problema es que los procesos de búsqueda puedan atravesarse sin encontrarse.

Búsqueda aleatoria

Búsqueda no ordenada de soluciones, recorre el árbol de búsqueda de forma exhaustiva. Este mecanismo de búsqueda es válido cuando la base de conocimientos es pequeña. La ventaja es que se puede elegir la solución óptima, dado que se encuentran todas las posibles soluciones.

Búsqueda heurística

Este mecanismo de búsqueda divide a la base de conocimientos en módulos, buscando sólo en alguna parte. Este tipo de búsqueda ahorra tiempo y se basa en una agenda.

NIVELES DE CONOCIMIENTO

En una base de conocimientos deben encontrarse tres niveles de conocimiento: conocimiento de decisión, conocimiento de soporte y meta-conocimiento. En el mundo real, la diferencia entre estos niveles de conocimiento raramente es clara.

Conocimiento de Decisión

Este tipo de conocimiento se refiere a las reglas usadas más o menos explícitamente por el experto en la búsqueda de una decisión. Este nivel es típicamente el único que puede ser especificado en un shell.

Conocimiento de Soporte

En un sistema basado en el conocimiento, no sólo es importante el hecho de tomar una decisión basada en un IF-THEN ("si..., entonces"), sino también las razones de por qué debe o no ser tomada tal decisión. Por tal motivo, es necesario un nivel de conocimiento que soporte las decisiones tomadas.

Meta-conocimiento (reglas acerca de reglas)

Este tipo de conocimiento afecta al control del proceso de la toma de decisiones más que a la decisión en sí. El meta-conocimiento se compone de meta-reglas, que son aquellas reglas que afectan la manera en cómo usa el motor de inferencia a las reglas de decisión (de las cuales forma parte el conocimiento de decisión), esto es, indica qué conocimiento debe ser elegido y aplicado en cada momento.

Parte de este conocimiento puede ser implementado dentro del motor de inferencia; en los shells este conocimiento ya se encuentra implementado de tal forma que no se encuentra disponible para incluirse alguna información adicional, que haga óptima la búsqueda de una solución.

LENGUAJES Y HERRAMIENTAS PARA SE

Es de esperarse que mientras se usen computadoras tradicionales (no diseñadas para aplicaciones de IA), el resultado final de un programa escrito

en cualquier lenguaje, será un programa en código máquina que se ejecutará de manera secuencial en una computadora.

Es cierto entonces que se pueden elaborar SE en cualquier lenguaje, pero unos presentan más facilidades que otros, al momento de programar, dependiendo de qué tan especializados estén y de para qué aplicación se desee realizar el SE.

Existen varios lenguajes y, dependiendo del nivel de programación, los podemos encontrar entre:

- Lenguaje ensamblador
- Lenguajes imperativos
- Lenguajes funcionales orientados a estructuras de datos
- Lenguajes imperativos y orientados a objetos
- Lenguajes declarativos y orientados a reglas
- *Shells*

Cabe remarcar que entre más especializadas sean los lenguajes, el diseño del SE es más rápido pero se pierde libertad al aceptar las condiciones del lenguaje ya fijadas.

Un lenguaje de programación de bajo nivel "habla" directamente a la computadora en código máquina, en tanto que un lenguaje de alto nivel es escrito en palabras que deben ser traducidas posteriormente, al código máquina. Esas traducciones pueden ser hechas por un intérprete o un compilador. Un intérprete traduce el código en lenguaje de alto nivel al código numérico del lenguaje de bajo nivel; conforme se va introduciendo el programa, se realiza el proceso de interpretación. Un compilador es un programa que toma instrucciones de alto nivel y las traduce todas ellas, de una sola vez, en un código de máquina ejecutable.

Lenguajes imperativos

Este tipo de lenguajes se basan en secuencias o procedimientos, es decir, el flujo del programa pasa a la siguiente línea a menos que se pida lo contrario.

Estos lenguajes proporcionan gran flexibilidad, el tiempo para aprenderlos es corto. Desafortunadamente, los programas desarrollados son complejos y largos, por lo que no se pueden modificar fácilmente.

Ejemplos de estos lenguajes son: PASCAL, C, BASIC, FORTRAN.

Lenguajes funcionales

En este tipo de lenguajes, el flujo depende de la evaluación de una función cuya posición física no es importante. Su procesamiento es recursivo. Pueden trabajar de manera simbólica y la construcción del programa lógico para elaborar nuevos conocimientos puede realizarse de manera sencilla. Las desventajas de este tipo de lenguajes es su requerimiento de memoria y velocidad, además del hecho de que son entornos cerrados, es decir, que no permiten la interacción con otros programas.

Ejemplos de este tipo de lenguajes son: LISP, cuyas estructuras son listas; y LOGO, que usa cadenas.

Lenguajes orientados a objetos

En estos lenguajes ya no se declaran variables. Usan definiciones y relaciones de objetos para obtener resultados, lo que hace que no exista una diferencia entre procedimientos (llamados métodos) y datos (llamados facetas), pues ambos son objetos.

Los objetos envían mensajes que corresponden a las acciones que realiza el programa. Un objeto pertenece a una clase de la cual hereda propiedades y

puede tomar valores por defecto. Los objetos son clasificados jerárquicamente, dependiendo de la clase o subclase a la que pertenezcan.

La programación se empieza identificando los objetos que se encuentran en el problema y en la solución, clasificándolos y redactando los mensajes que interaccionan a los objetos para posteriormente implantar métodos en ellos.

Un ejemplo de este tipo de lenguaje es: SMALLTALK.

Lenguajes declarativos

Los programas elaborados en lenguajes declarativos, buscan demostrar un hecho para el cual se define un universo sobre el cual se realiza la demostración, controlada por el motor de inferencia.

La forma en cómo se representa el conocimiento se basa en reglas de producción. El motor de inferencia se basa en la búsqueda en profundidad, marcha atrás, comparación de patrones. Permite el encadenamiento hacia adelante y hacia atrás.

Los puntos débiles son la difícil programación de operaciones de entrada y salida, la representación del conocimiento es única y no permite la interacción con otros programas.

Algunos lenguajes de este tipo son: OPS, PROLOG.

Los Shells

Un *shell* es un medio ambiente construido en un lenguaje de alto nivel (la mayor parte de los shells fueron desarrollados en LISP y PROLOG). Cuando la tecnología de los SE estaba siendo usada por primera vez en aplicaciones comerciales, los shells eran adaptaciones de las herramientas de investigación de las universidades, por ejemplo, es SE de diagnóstico médico MYCIN, convertido en una herramienta independiente llamada EMYCIN.

Los *shells* son también llamados sistemas vacíos ya que para poder hacerlos operativos se necesita introducirles el conocimiento. Son sencillos y rápidos en la programación, pero no son muy flexibles (consecuencia de su alto grado de especialización). Los *shells* no son herramientas para la adquisición de conocimiento, las cuales sólo adquieren y organizan el conocimiento pero no resuelven problemas usando el conocimiento adquirido.

Para entender lo que es un *shell*, imaginemos primero a un SE ya creado. Si al SE le extraemos toda la información o base de conocimientos de la aplicación para el cual fue creado, obtendríamos un cascarón que podría recibir información de otra aplicación en particular y poder así crear otro SE.

Dicho cascarón se compone del motor de inferencia, del componente para la explicación y ocasionalmente de la interfaz del usuario. Este *shell* puede contener *frames* en los cuales sólo se introduce el nombre del objeto y sus características.

Es de entenderse que un *shell* no es apropiado para todas las aplicaciones pues los componentes de cada *shell* varían dimensionalmente, debe entonces buscarse el *shell* apropiado a cada aplicación para la cual se desea realizar un SE.

En los últimos años ha habido una proliferación de *shells* comerciales disponibles con un amplio rango de precios y variando desde sistemas preliminares basados en reglas simples hasta poderosas herramientas.

4

APRENDIZAJE Y SISTEMAS EXPERTOS TUTORIALES

ACERCA DEL APRENDIZAJE

Las teorías del aprendizaje están relacionadas con la pregunta de cómo trabaja el cerebro en el aprendizaje, bajo diversos puntos de vista. Por lo cual estas teorías suelen clasificarse en varias categorías que cambian su esquema conforme las investigaciones en la educación avanzan y se aplican.

LA PSICOLOGÍA CONDUCTISTA

La psicología conductista clásica mantiene que lo que ocurre en el cerebro es científicamente imposible de conocer y maneja una teoría que se ha denominado como de Estímulo/Respuesta, la cual establece que el hombre reacciona sólo a estímulos, los cuales introducen información al cerebro y en base a esta información produce respuestas.

B. F. Skinner, por ejemplo, manejaba una teoría conductista, muy influyente en los años cuarenta, enseñando a las palomas a picotear diseños complejos. El trabajo era dividido en pequeñas tareas que premiaba con comida y ocasionalmente castigaba los resultados negativos. Este criterio influyó, en relación al proceso de aprendizaje, la enseñanza programada con textos programados.

LA PSICOLOGÍA COGNITIVA

Aunque la psicología cognitiva no es nueva, es considerada como una de las teorías más avanzadas. Los psicólogos de esta teoría creen, contrariamente a los psicólogos conductistas, que es esencial entender el esquema que utiliza el cerebro para organizar el conocimiento interno.

Una de las ramas de la psicología cognitiva intenta imitar las actividades del cerebro a través de programas con computadoras por lo cual está muy relacionada con la Inteligencia Artificial. Pero cabe aclarar que el hecho de tener una idea de cómo funciona el cerebro no es sinónimo de tener una teoría del aprendizaje adecuada.

LA PSICOLOGÍA EVOLUTIVA

La teoría de la psicología evolutiva se dedica a comprender las etapas de la evolución intelectual en los seres humanos. El líder en este campo fue J. P.

Piaget quien marcó las etapas de la evolución en las cuales indica que a partir de los 15 años el hombre alcanza el nivel operativo formal (última de las etapas), pudiendo éste generalizar todas las situaciones.

Son criticados, por algunos investigadores, algunos de los aspectos de sus conclusiones y las etapas que menciona, entre otros detalles.

EL CICLO DEL APRENDIZAJE

Robert Karplus, de la Universidad de California, aportó una de las teorías más interesantes respecto al ciclo del aprendizaje, observó que el aprendizaje completo requería de tres fases:

1. Fase experimental. En esta fase se permite que el estudiante juegue con los fenómenos desarrollando la intuición y el discernimiento a través de experiencias relacionadas con el tema. Los datos son puestos en contacto con el estudiante, al cual no se le proporciona ayuda.
2. Fase de recepción de conceptos. El material con el que el estudiante tuvo contacto en la fase experimental, es manejado en esta fase para aprender conceptos, ofreciendo datos nuevos. En las escuelas tradicionales, la educación se basa solamente en el aprendizaje de conceptos.
3. Fase de aplicación. Esta etapa se orienta hacia las aplicación de los conocimientos adquiridos, donde se pueden incluir pruebas cuyos resultados propician la creación de nuevo material que refuerza los conceptos débiles.

ASPECTOS NEUROLÓGICOS

Como ya se mencionó anteriormente, aún no se conoce bien el funcionamiento del cerebro. Sus aspectos químicos y eléctricos son complejos y no se pretende relacionarlos con el aprendizaje.

El cerebro humano tiene dos hemisferios, izquierdo y derecho; los dos comparten el pensamiento y la regulación corporal: el hemisferio derecho regula al izquierdo y viceversa; debido a que una mitad puede compensar el daño que sufriera la otra. Cuando un niño sufre una lesión en alguna parte de un hemisferio, el otro hemisferio, el que queda ileso, cumple con las funciones del otro. La adaptabilidad disminuye conforme avanza la edad, pues los hemisferios cerebrales tienden a especializarse.

Cada hemisferio es diferente del otro, esta asimetría del cerebro es un indicio de la especialización peculiar de cada una; sin embargo, ambos hemisferios interaccionan.

En el aspecto neurológico, se ha prestado atención a los efectos de intensificar la conexión entre los dos hemisferios del cerebro; esto es, la lateralidad. En términos generales, el trabajo sobre la lateralidad muestra que en la mayoría de las personas, los dos hemisferios funcionan de manera distinta. Generalmente, para una persona diestra, el hemisferio izquierdo controla, al menos de manera parcial, el lenguaje escrito y hablado, la lógica, las aptitudes numéricas y los conocimientos científicos; en tanto que el hemisferio derecho se relaciona mayormente con la información visual (reconocimiento de formas y trazos y la relación que hay entre ellos), con la estructura general y con la organización, contribuye al discernimiento y la imaginación.

Lo anterior, nos sugiere que un buen material de aprendizaje debería estimular las dos partes del cerebro.

Roger Sperry, principal investigador de la lateralidad, comentaba en su discurso para el premio Nobel 1981:

"A pesar de las dudas que aún tenemos acerca de la lateralidad del cerebro, parece que hemos conseguido un resultado beneficioso, el de fomentar

una fuerte conciencia, tanto en la educación como en cualquier otro aspecto...¹⁵

REAPRENDIZAJE

La técnica del reaprendizaje intenta medir el tiempo que se ahorra al adquirir conocimientos de un material ya aprendido con anterioridad. Resulta más fácil y se ocupa menos tiempo en aprender materias completas que ya han sido estudiadas tiempo atrás, que aprender esas mismas materias por primera vez.

El reaprendizaje ocupa mucho tiempo pues esta técnica abarca el tiempo que se requiere para que un estudiante aprenda cierta información, la olvide conforme transcurre el tiempo y finalmente, se le presente al estudiante la misma información a ser adquirida.

La eficacia del reaprendizaje radica en la facilidad con la que una persona aprende cierto material.

LA EVALUACIÓN

Evaluar consiste en el conjunto de acciones que tienen como propósito determinar el valor de los logros alcanzados por los alumnos, con respecto a los objetivos planteados para el curso. La evaluación es parte primordial del proceso de enseñanza-aprendizaje, es parte del hecho educativo.

Es importante señalar que no se evalúa a la persona, sino al comportamiento que presenta ante una situación concreta:

¹⁵ *Science*, 24 de septiembre de 1982, p. 1623. Citado por A. Bork en *El Ordenador en la Enseñanza*.

- *Dominio de conocimientos:* Manejo adecuado de datos, conceptos, terminología, convenciones, clasificaciones, teorías, métodos.
- *Destrezas específicas:* Manejo de técnicas, procedimientos propios de cierto trabajo.

La evaluación, de acuerdo con el momento en el proceso de enseñanza-aprendizaje, tiene diferentes funciones y características. Así pues, encontramos tres tipos de evaluación:

- *Evaluación Diagnóstica.* Se aplica al inicio del proceso y su finalidad es conocer las características de entrada (requeridas para el curso) que tiene el alumno.
- *Evaluación Formativa.* Es también llamada evaluación continua y sirve para juzgar sistemáticamente los resultados parciales de la enseñanza. Se emplea en cualquier momento después de haber iniciado el curso con la finalidad de registrar el progreso del alumno. Esta evaluación motiva al alumno, pues se da cuenta de que se le sigue en el proceso.
- *Evaluación Sumativa.* Sirve para juzgar los resultados del aprendizaje después de determinado periodo o al final del curso, con el fin de comprobar el logro de los objetivos planteados.

Instrumentos para la evaluación

Algunos de los instrumentos para la evaluación más usados son:

- *Pruebas de ejecución.* Se dividen en observación directa de la conducta y registro de resultados.
- *Pruebas orales.* Se trata de observar si el alumno logró la comprensión de un tema, por medio de la explicación que de sobre él.

- Pruebas escritas. Son las más usuales y pueden ser de respuesta estructurada o no estructurada.

Los reactivos de respuesta no estructurada son:

- a) De complementación. Son frases incompletas que piden al alumno una palabra que les dé sentido.
- b) De respuesta breve. Son preguntas o indicaciones para que el alumno responda en pocas palabras.
- c) De ensayo. Se pide a los estudiantes que desarrollen un tema o resuelvan un problema que exige un desarrollo completo.

Los reactivos de respuesta estructurada son:

- a) Falso-verdadero. Consisten en un enunciado afirmativo que el alumno debe marcar como cierto o como falso.
- b) Jerarquización. Son una serie de conceptos que pueden ser ordenados bajo un criterio claro que se especifica en el reactivo.
- c) Correspondencia. Son aquellos que requieren de relacionar una columna de conceptos con otra columna de conceptos, sin embargo, puede sustituirse una columna por un esquema.
- d) Opción múltiple. Consiste en un encabezado y algunas opciones que pueden dar respuesta a éste.

EL PIZARRÓN

El pizarrón es el más antiguo recurso didáctico visual y sigue teniendo gran eficacia. Puede usarse:

- Para desarrollar problemas, fórmulas y ejemplos.

- Para elaborar cuadros sinópticos, resúmenes.
- Como medio ilustrativo.
- Cubierto, para ir descubriendo secciones previamente preparadas en cierta secuencia.

APRENDIZAJE INDIVIDUALIZADO

La enseñanza es importante sólo si conduce a aprender¹⁶.

La mayoría de los estudiosos en la teoría del aprendizaje, cualquiera que sea su corriente, están de acuerdo con la declaración de Sperry acerca de que el aprendizaje es un proceso muy individualizado.

Los estudiantes tienen habilidades e historiales académicos diferentes, y probablemente sus opiniones acerca del concepto "aprender" son diversas. El tiempo que cada estudiante requiere para aprender puede que sea distinto.

Uno de los problemas centrales en cualquier sistema educativo es el hallar la forma efectiva para llegar al estudiante y que éste capte la información, que entienda. Este problema raramente se trata de una forma adecuada en los sistemas de educación actuales. Así, de manera sorpresiva, la mayoría de los medios disponibles para la educación masiva no permiten en la actualidad una educación medianamente individualizada.

El ambiente en una institución educativa, donde asisten a clase grandes cantidades de alumnos, proporciona a cada estudiante un nivel de atención individual evidentemente bajo, pues cada uno recibe la misma materia y en la misma proporción. Quizás unos pocos estudiantes podrán consultar al ins-

¹⁶ A. Bork en *El Ordenador en la Enseñanza*.

structor fuera de las aulas, pero el mismo número de alumnos por clase impide que ésta sea una práctica común.

Por otra parte, un libro permite que los buenos lectores lo utilicen de una manera algo más individualizada: pueden pasar de una sección a otra hojeando y examinando, además de consultar los libros que se encuentran como referencias. De cualquier forma, muchos lectores no emplean tales estrategias, pocas personas aprenden a leer bien. Para los lectores menos adiestrados, el libro no es un medio más individualizado que una clase o conferencia. Esto es, los libros no ofrecen más que una individualización reducida.

*"Muy pocos estudiantes reflexionan sobre lo que perciben del texto y por consecuencia, no generan ideas nuevas, carecen de la creatividad y no son constructores de su propio conocimiento"*¹⁷

Algunas organizaciones educativas logran, en determinadas circunstancias, cierta individualización. La interacción entre el instructor y los alumnos en condición de uno a uno, es un ejemplo de cómo la instrucción puede ser individualizada, lo que abarca solamente una pequeña parte del proceso total del aprendizaje. En las típicas clases de escuela elemental o secundaria, la mayoría de los profesores abandonan la individualización debido en gran parte a la agobiante presión de tener un número de alumnos elevado. Aunque algunos profesores, realmente buenos, puedan proporcionar cierta individualización en clases numerosas, la carga es agotadora. Es difícil observar experimentalmente alguna diferencia entre un grupo compuesto por 10 alumnos y un grupo de 100 alumnos. La educación en masa no permite individualización.

Incluso, en grupos reducidos, muchos instructores no proporcionan individualizaciones. Cuando algún alumno consulta al profesor fuera del aula puede suceder que éste solamente se limite a repetir una parte de la lección. Disponiendo de buen material, una computadora puede proporcionar una dedicación individual adecuada a las necesidades de cada alumno; serviría en una gran serie de funciones educacionales, en algunas de las cuales un libro no resulta útil. Las funciones incluyen instrucción y práctica, uso de herra-

¹⁷ Paulo Freire, citado por Yolanda Argudín y María Luna en *Aprender a Pensar Leyendo Bien*.

mientas profesionales con las cuales se atacan problemas significativos, exploración de micromundos desarrollados con propósitos educacionales que simulan ciertas actividades donde el aprendizaje consiste en hacer y, la autoexaminación.

Los psicólogos, por otra parte, sostienen que es necesaria la internalización de la materia, lo cual se puede lograr a través de la participación activa de los estudiantes, es decir, debe considerarse al estudiante como un participante activo en el proceso de conocimientos.

Un ejemplo de lo que la interacción en el aprendizaje significaría es el de colocar un grupo reducido de estudiantes en la oficina del profesor, el cual sólo les dirige preguntas para que los propios alumnos obtengan respuestas por sí mismos de una forma activa. A esto suele llamársele "descubrimiento dirigido".

El concepto de Piaget acerca de la interacción incluye la acción sobre el conocimiento o actividad intensa de conocimiento

APRENDIZAJE POR COMPUTADORA (TUTORIALES)

*"La máquina no aísla al hombre de los grandes problemas de la naturaleza: lo sumerge en ellos"*¹⁸

Una computadora por sí sola no despierta interés a los educadores como medio de enseñanza. El interés surge a partir del desarrollo de herramientas educativas para computadoras.

Se debe tener claro que el uso de las computadoras en la educación, al igual que otros elementos y técnicas, por sí mismas no se pueden catalogar de buenas o malas. Dado su potencial, pueden inclinarse hacia cualquier lado,

¹⁸ Antoine de Saint-Exupéry en *Tierra de Hombres* citado en *Los Porqués de la Mente Humana*

por lo tanto, la labor radica en aminorar sus defectos y en dar énfasis sus cualidades.

VENTAJAS DEL USO DE LAS COMPUTADORAS

Primeramente, contamos con un elevado factor de motivación que despierta una computadora en los estudiantes, pues la gente recibe constantemente referencia de ellas en películas, televisión y medios de comunicación impresos. Los estudiantes que no tienen contacto con las computadoras ven en ellas máquinas apasionantes y ansían tener contacto con ellas.

Por sus experiencias en el Centro de Tecnología Educacional, A. Bork menciona que no ha encontrado estudiantes que no quisieran utilizar la computadora siempre que el uso de ellas no fuera prolongado. Así pues, la idea de estar frente a una máquina que "contesta" resulta atrayente para el estudiante, pues éste se siente como si estuviera unido a la aplicación. Aunado a esto, la computadora se presenta como un mecanismo que saca al estudiante de la rutina.

Dado que el grado de aprendizaje depende del tiempo que se dedique a estudiar un tema, es necesario poner énfasis a los factores motivadores, tales como la computadora, ya que inducen a que los estudiantes empleen más tiempo en aprender, por medio del repaso, al ser el tutorial un material adicional.

Con una computadora como material único de aprendizaje, los estudiantes adquieren los conocimientos invirtiendo menos tiempo del que les tomaría hacerlo en una clase convencional, debido a que la computadora está dedicada a un solo estudiante. A. Bork maneja una disminución de 30% del tiempo invertido gracias a la individualización: una vez que se verifican los conocimientos del estudiante, la aplicación puede pasar a otros temas, no antes. La cuestión del tiempo que se emplea para aprender es especialmente importante en los lugares donde el personal es apartado de su trabajo productivo en una compañía para estudiar algo.

Otra ventaja que poseen las computadoras es el que pueden proporcionar información visual interactiva. La interacción es en esencia, un diálogo entre el usuario final y la computadora, donde se usan datos como medio de comunicación; entre los datos se encuentran los gráficos que llaman mucho la atención ya que estimulan la parte creativa del cerebro, fomentando así la estimulación de ambos hemisferios del cerebro.

El uso de la computadora en la educación trae consigo la ventaja de poder presentar una lección como un juego en el que las repuestas incorrectas recibidas por la computadora hacen desplegar de alguna forma las correctas, fomentando así la instrucción-práctica.

Ahora veamos las posibilidades que ofrecen los hipertextos. Éstos proveen oportunidades significativas de expandir interacciones durante y acerca de la lectura; si son usados ventajosamente, pueden ayudar a los estudiantes a enterarse de los resultados fundamentales en la construcción del conocimiento, ofreciéndoles colaboraciones en línea. Los hipertextos pueden ser descritos como una base de datos que es organizada como una enciclopedia de referencia cruzada que permite saltar de un texto o gráfica desplegada a otro desplegado que proporciona más información sobre un punto, ya sea una palabra o parte de un gráfico, de la pantalla anterior.

Un tutorial nunca quedaría obsoleto, ya que éste se puede ir actualizando en información y en presentación. Además puede irse perfeccionando conforme se vayan presentando, mediante su uso, errores o bien se descubran puntos en los que se necesite remarcar un concepto.

PRESENTANDO A LOS TUTORIALES INTELIGENTES

Desde el surgimiento de la computadora como medio de fácil acceso o adquisición, ha surgido el interés en sistemas de instrucción basados en computadoras que desplieguen cierto grado de inteligencia. Estos sistemas tutoriales inteligentes, lo que hemos llamado Sistemas Expertos Tutoriales (SET), son bien avenidos por muchos educadores y especialistas de instrucción, quienes los ven como desafíos en la investigación.

Todos los SE son por si mismos herramientas para la enseñanza. Los SET se enfocan a la educación, son inteligentes no en cuanto a su contenido, sino a la forma en como el sistema enseña su contenido. Un SET proporciona apoyo a una tarea experta: la enseñanza. Es valioso económicamente pero técnicamente limitado. Los SET se desarrollan generalmente en una computadora personal y se considera asistente. Es importante que un sistema de este tipo no presente solamente pantallas cargadas de información.



Figura 4 - 1 Estructura de un SET

Los primeros intentos de creación de software educacional tuvieron resultados variados. Comúnmente los proyectos habían hecho más sacando a la luz problemas que cumpliendo las promesas, tanto de realizaciones técnicas como pedagógicas y su integración dentro de los fines de las instituciones educacionales. La dificultad de producir buen material dejó cierta incertidumbre, llegando a la desilusión; fue la Inteligencia Artificial quien dio nuevas esperanzas al viejo sueño. Dado el gran interés generado alrededor de la Inteligencia Artificial, ampliado por lo referente al gran potencial de la educación e instrucción, se ha motivado el desarrollo de varios proyectos de investigación en el área de los SET.

Mucho de la investigación sobre el tema de software educacional dentro de la Inteligencia Artificial ha sido conducido bajo el nombre de ICAI (*Intelligent Computer-Aided Instruction*, Instrucción Inteligente Apoyada en

una Computadora), más recientemente el acrónimo de ICAI ha sido reemplazado por el de ITS (*Intelligent Tutoring System*, Sistemas Tutoriales Inteligentes). En esta tesis se hace referencia a ellos como SET (Sistemas Expertos Tutoriales).

En los sistemas ICAI, la experiencia ha ser comunicada es contenida en bloques de presentación acumulados anteriormente, algunas veces llamados *frames* los cuales son diseñados por el maestro experto y son desplegados al estudiante bajo condiciones dadas. Entre la transición de ICAI a SET, se encuentra históricamente el primer aspecto en el que el conocimiento de la materia da como resultado la representación explícita en los sistemas de la experiencia del maestro.

En los sistemas que comunican el conocimiento, hay un módulo especial llamado comúnmente *experto* que contiene una representación del conocimiento a ser comunicado. En la mayoría de los casos esta representación del conocimiento no es sólo una descripción de varios conceptos y destrezas que el estudiante debe adquirir, el modelo actual provee al sistema con una forma dinámica de experiencia.

Software educacional como contenedor de información

Tradicionalmente, los programas educacionales eran receptáculos estructurados organizados estáticamente para incorporar un área en particular y el conocimiento pedagógico de maestros expertos. En este sentido, el uso de computadoras para el desarrollo de software educacional es evocador de la consideración de libros como contenedores del conocimiento de sus autores.

Los libros tienen herramientas para la presentación organizada de conocimientos e ideas. Sin embargo, uno no espera que un libro proporcione acceso dinámico al conocimiento que contiene, de manera tal que pueda contestar las preguntas del lector o mostrar nuevas deducciones relevantes. No esperamos que un libro sea capaz de modificar su presentación y adaptarse a las necesidades específicas de los lectores.

Un libro puede soportar varios niveles de lectura, invitando al lector avanzado a saltarse ciertas secciones o proveerle de un glosario y referencias entre secciones. Todo lo cual debe ser tomado en cuenta primeramente por el autor conjuntamente con el medio de impresión. De la misma manera, la tarea de crear material educacional, como un SET, puede ser un proceso extremadamente sofisticado al trasladar las decisiones pedagógicas de un maestro dentro del programa.

Sin embargo, todas las circunstancias en donde las decisiones son necesarias, deben de ser anticipadas, así que el programa debe ser escrito en un lenguaje apropiado para capturar tales decisiones. La fuerza del enfoque tradicional de los programas educacionales reside en la habilidad de tomar ventaja directamente de la experiencia pedagógica humana y reflejarla lealmente en el comportamiento del sistema.

Sistemas Expertos Tutoriales: Codificación del conocimiento

Para los investigadores sobre sistemas educacionales envueltos en la Inteligencia Artificial, el propósito no es proveer una armazón de ingeniería de software dentro del cual los expertos puedan componer interacciones instruccionales por representamiento de sus decisiones en la forma de programas. Más bien, el propósito es capturar el conocimiento que permite al experto componer una interacción instruccional, de tal manera que pueda ser usado en sistemas basados en computadoras. La responsabilidad es desarrollar programas para componer interacciones instruccionales dinámicamente, tomando decisiones por referencia al conocimiento con el cual han sido previstos.

Obviamente no hay una línea divisoria estricta entre los dos tipos de programas, estáticos y dinámicos. Los sistemas actuales se encuentran situados entre una continuidad entre la autonomía perfecta y la programación previa totalmente arreglada. Por un lado, los sistemas ICAI más sofisticados tienen algunas capacidades autónomas: por instancia, ellos pueden generar ejercicios o adaptar el nivel de dificultad, dependiendo del estudiante.

Por el otro lado, modelos de experiencia producidos por la Inteligencia Artificial varían grandemente en el grado en que se hacen explícitas decisiones específicas subyacentes a su conocimiento. De hecho, la investigación está aún lejos de la meta ideal de tener un sistema capaz de razonamiento pedagógico enteramente autónomo, puramente basado en principios primitivos, en el dominio del conocimiento tan bien como en experiencia pedagógica.

El estilo de enseñanza de los sistemas tutoriales inteligentes puede acabar difiriendo de lo que maestros humanos, en maneras significantes, tomarían ventaja de las características de las máquinas, como su capacidad de cálculo numérico.

Las computadoras no compiten con los maestros humanos en el sentido real, ni pueden reemplazar a la gente que instruye. La enseñanza, como un acto social, envuelve varias dimensiones en el exacto procesamiento de la información, para lo cual las computadoras están hechas.

Los libros son una gran fuente de información paciente, disponible para cuando un alumno así lo requiera. Ahora imaginemos un libro activo que pueda interactuar con los lectores para comunicar su conocimiento en un nivel apropiado, seleccionando la interconectividad y ramificaciones de temas, recalcando la información importante, probando la comprensión, explicando las áreas difíciles más profundamente....

Los sistemas expertos tutoriales son, por esto un sueño, atractivo.

DISCIPLINAS QUE INTERACCIONAN CON LOS SET

Con su interés en la inteligencia humana: la epistemología, la psicología y la ciencia cognitiva están ligadas a la investigación automatizada tanto como la Inteligencia Artificial lo está. Estas disciplinas buscan entender y construir modelos de cómo la gente conoce, como aprende. Los psicólogos han incorporado los propósitos y metodologías de los SET dentro de su investigación y han encontrado herramientas en el desarrollo de teorías cognitivas.

Obviamente la educación y la psicología educacional están muy ligadas en su intento de capturar los componentes de la experiencia pedagógica para acumular material de técnicas empíricas útiles y para producir herramientas educacionales útiles.

La antropología es otra disciplina relacionada; estudia el proceso central cognoscitivo para designar modelos explícitos de comunicación de conocimientos.

La noción de la comunicación de conocimiento brinda importancia a otras materias relacionadas con la comunicación. Por esto se incluye la lingüística que es relevante a dos niveles: primero, hay mucho interés en dotar a los sistemas expertos tutoriales con alguna capacidad para entender el uso del lenguaje natural; segundo, muchos de los actos, al hablar, pueden ser vistos como una forma de comunicación de conocimiento: lenguaje corporal. En los SET se observa cómo el uso de articulaciones, bases conversacionales o términos largos pueden causar modificaciones del estado del conocimiento de interlocutores.

COMUNICACIÓN DEL CONOCIMIENTO

Desde el punto de vista de la IA, los sistemas que comunican el conocimiento emplean actividades pedagógicas como una capacidad general de sistemas inteligentes. Desde el punto de vista de la educación, la comunicación del conocimiento no aporta las connotaciones sociales que instruyen.

La comunicación del conocimiento es la habilidad de causar y/o soportar la adquisición del conocimiento de alguien por alguien más mediante un conjunto restrictivo de operaciones de comunicación.

CONSIDERACIONES PARA ELABORAR LOS SET

El área en particular que cubrirá el SET necesita llegar a ser comprendida mediante preguntas profundas acerca de la naturaleza del conocimiento que abarca, la forma en cómo debe ser comunicado el conocimiento y la forma en como aprendizaje del área, es óptima. Los SET permanecen aún difíciles de construirse, la mayoría de los proyectos requieren años de trabajo de gente talentosa que son mejorados a través de varias generaciones de prototipos, algunos de los cuales se quedan en el laboratorio.

La enseñanza es difícil y el hecho de diseñar SET requiere de comprender todo el proceso que envuelve a una aplicación en particular, (No todos los problemas se atacan de la misma forma; así que no se puede sistematizar).

Módulos: Experto, del Estudiante y de la Evaluación

Un SET consta principalmente de dos módulos, el módulo experto del SET actuará como fuente del conocimiento que será presentado, generando explicaciones y respondiendo al estudiante. Este módulo sirve como estándar para evaluar el desarrollo de los estudiantes.

Si el SET va a guiar al estudiante en la solución de problemas, entonces el módulo experto debe generar trayectorias de soluciones sensibles, de manera que los pasos intermedios en esas trayectorias puedan ser comparados para la obtención de soluciones.

La adaptabilidad de un sistema educacional es determinada por la cobertura de la información contenida en el modelo del estudiante. La idea de tener un modelo del estudiante no es nueva, este modelo puede darse infiriendo aspectos no observables del comportamiento del estudiante, en orden para producir una interpretación de sus acciones y reconstruir la base de conocimiento que ha aumentado con esas acciones del estudiante inferidas.

El módulo de la evaluación conforma la parte inteligente del sistema, ya que es esta quien decide si un estudiante se encuentra apto para pasar al

siguiente tema o debe repetir el anterior. La manera en como se desarrolle este módulo dependerá de los fines para los cuales este pensada la aplicación.

La forma de comunicación con el usuario final

Para el buen éxito en la comunicación del conocimiento hacia el estudiante (usuario final), es importante que el sistema presente un tópico, donde la interfaz muestre la información de una manera entendible.

La interfaz es la forma final en la cual se presenta el SET, cuyas cualidades de fácil uso y atracción visual pueden ser cruciales en la aceptación del sistema por el estudiante. Por otra parte, el desarrollo tecnológico ha dado origen a la elaboración de herramientas sofisticadas las cuales cuentan con una comunicación poderosa, entre varias aplicaciones, con las cuales se puede diseñar el sistema entero.

Un ejemplo de enseñanza, es el usado por sistemas como el SCHOLAR, por Jaime Carbonell, 1970, considerado como un clásico. Este sistema fue diseñado para conducir diálogos tutoriales acerca de la geografía sudamericana, esto es, enseña mediante diálogos. El tutorial no enseña al sujeto mediante exposición directa, pero permite al estudiante hacer preguntas sucesivas para formular principios generales en base a casos individuales para examinar la validez de nuestra hipótesis, para descubrir contradicciones y finalmente, para atraer inferencias correctas de hechos conocidos.

5

DESARROLLO, APLICACIÓN Y EVALUACIÓN DE SECATA

EL DESARROLLO DE UN SET

Una vez identificado el problema que solucionará el SET, empieza la gran labor de desarrollarlo.

Para desarrollar un SET es necesario identificar a los expertos que participarán, y formar un conjunto de factores de decisión esenciales del área que abarca su experiencia; para esto, es necesario y de suma importancia, entrevistar a expertos en el área para la cual se elaborará la aplicación. Hay que poner atención en el hecho de que las decisiones que tome el experto durante las entrevistas, seguramente cambiarían si el experto se encontrara bajo presión.

También se deben crear algunas reglas generales para iniciar la base de conocimientos que contendrá el prototipo. El simular el proceso de decisión del experto es un trabajo difícil, para empezar, es necesario leer todo el material introductorio como si el elaborador de SET fuera un estudiante de la materia y posteriormente simular el proceso que realiza el experto en una máquina que no posee intuición.

PROCESO PARA IMPLEMENTAR UN SE

Todos los SE comienzan con la percepción de un problema que es difícil de resolver usando los métodos de programación tradicionales. Los SET solamente son una extensión del conjunto de herramientas que provee una computadora y tienen el potencial de asistir a los usuarios, de la manera en que un libro no podría hacerlo, presentando sólo las preguntas que necesitan ser contestadas y que se basan en respuestas anteriormente dadas por el usuario.

Dado que un SET requiere de experimentación para su buen desempeño, su desarrollo es gradual. El procedimiento para extraer el conocimiento del experto y codificarlo en forma de programa, se llama adquisición de conocimiento¹⁹. Esta transferencia de información es el corazón del proceso del desarrollo del SET.

La figura 5 - 1 muestra los pasos envueltos en la implementación de un SE, mismos que se encuentran en el desarrollo de un SET; están clasificados de acuerdo a las cuatro fases principales de la implementación.

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

¹⁹ En algunos libros se le llama Técnicas de Procesos de Desarrollo de Software



Figura 5 - 1 Proceso para implementar un SE

Inicialización del proyecto

Esta primera parte es una introducción general al proceso que conduce a los requerimientos de un SE. Comienza con la percepción de un problema, examinando la manera en la cual los expertos presentan una posible solución. De la necesidad percibida, el grupo de trabajo puede valorar la utilidad de un SE, desarrollando criterios con los cuales evaluar la factibilidad del proyecto. A través de este primer nivel, se asume que no se cuenta con el conocimiento implementado.

Durante la identificación del problema, el ingeniero del conocimiento y el experto trabajan juntos para identificar el área del problema y definir su alcance.

La decisión de elaborar un SE requiere de una evaluación previa del problema, el cual debe ser de dificultad razonable, además de no presentar una solución algorítmica posible o deseable, como es el caso de la elaboración de

tutoriales inteligentes (SET) los cuales contienen los conocimientos y habilidades de los pocos expertos existentes en el área.

Entrevistar a expertos es importante pero cabe remarcar que las decisiones que toman en una entrevista son diferentes a las que toman bajo presión. Hay que darse a la tarea de identificar expertos y hacer un conjunto de factores de decisión relevantes a dominio que abarca su experiencia.

Evaluación de Necesidades

En esta etapa, se identifican los participantes en el proceso de desarrollo (experto en el área en que se desarrollará el SE, especialista en IA, expertos adicionales y usuarios finales), se determinan los recursos necesarios (tiempo, facilidades de cómputo) y se deciden los objetivos de construcción del SET.

Una vez definido el problema que atenderá el SE, se localizan los conceptos claves y relaciones dentro de una representación formal sugerida por alguna herramienta de construcción de SE o lenguaje. El ingeniero del conocimiento debe seleccionar el lenguaje y, con ayuda del experto, representar los conceptos básicos y relaciones dentro del área de trabajo del lenguaje.

Para poder elegir un shell adecuado, debe tenerse en cuenta su costo y su capacidad de contener objetos y/o reglas, además de que la distribución del SE final sea adecuada a los requerimientos del problema. El shell debe presentar una interfaz del usuario agradable, con las facilidades que requiera el problema para el cual se desarrollará la aplicación.

Durante la conceptualización el experto y el ingeniero del conocimiento explican los conceptos clave, relaciones, reglas de inferencia, restricciones y métodos de razonamiento, características necesarias para describir el proceso de resolución del problema en el dominio dado. El experto y el ingeniero del conocimiento, trabajan en estrecha colaboración para conformar la base de conocimientos del SET.

Implementación

Durante la implementación, el ingeniero del conocimiento combina y reorganiza el conocimiento formalizado para hacerlo compatible con las características del flujo de información del problema. El conjunto resultante de reglas y la estructura de control asociada, define un programa prototipo que puede ejecutarse y ponerse a prueba.

Depuración del SE

Finalmente, la prueba envuelve la evaluación del prototipo y lo revisa para conformar a estándares de excelencia definida por expertos en el área (dominio del problema). Típicamente, el experto evalúa el programa y asiste al ingeniero del conocimiento en revisiones subsecuentes.

Estas etapas del desarrollo del SE no están bien definidas, sólo muestran una guía (no estricta) para la elaboración de un SE y las etapas pueden fundirse, volverse más o menos complejas, dependiendo de la aplicación para el cual se desarrolle el SE.

DESARROLLO DE SECATA

SECATA es un Sistema Experto Tutorial que capacita a controladores de tránsito aéreo en la fase de aproximación y que presenta las facilidades que ofrece una computadora como elemento para la enseñanza.

En su primera versión abarca las Operaciones IFR, parte correspondiente al Control de Aproximación, ya que el propósito es mostrar las bondades de los SET y motivar el interés por el desarrollo de SECATA.

INICIALIZACIÓN DEL PROYECTO

La mayor parte de los egresados como controladores de tránsito aéreo se dedican al control de aeródromo y con el paso del tiempo, sus conocimientos sobre el control de aproximación se van atenuando. Cuando un controlador de aproximación deja su puesto, se requiere un reemplazo lo más pronto posible; instructores son movilizados para capacitar al controlador de aeródromo en la fase de aproximación, lo que genera grandes gastos. Debido a esto, el tener un sistema capaz de capacitar a un controlador en temas que ya ha visto anteriormente y que sólo son explicados por expertos del tema, es una necesidad indiscutible.

El hecho de que el controlador estaría aprendiendo nuevamente conocimientos específicos, daría al sistema la ventaja del reaprendizaje. Por lo que, la capacitación resulta más fácil y el controlador ocupa menos tiempo en aprender materias completas que ya han sido estudiadas tiempo atrás.

Desde luego, el tutorial abarcaría únicamente la parte teórica y solamente sería necesario aplicar un examen práctico al final de éste.

Así, SECATA se concibe como el sistema experto tutorial para capacitar a controladores de tránsito aéreo en la fase de aproximación.

EVALUACIÓN DE NECESIDADES

El Shell elegido

El *shell* elegido para la elaboración de SECATA fue LEVEL5 OBJECT ya que es un ambiente de desarrollo de aplicaciones que combina la tecnología de Sistemas Expertos, programación orientada a objetos, modelos de bases de datos relacionales, capacidades de hipertexto, desarrollo de gráficos y herramientas de depurado. El ambiente que proporciona actúa como mediador gráfico entre el usuario y el programa en sí.

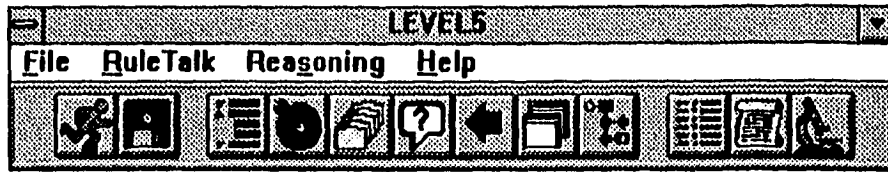


Figura 5 - 2 Barra de herramientas de LEVEL5 OBJECT 2.5

Las características principales de LEVEL5 son:

- a) LEVEL5 es lo que se conoce como un sistema híbrido, es decir, que puede manejar tanto objetos como reglas. De este modo el conocimiento puede ser conducido de manera muy similar a como es manejado por el ser humano; por un lado, se tienen un conjunto de objetos estructurados y jerarquizados dentro de clases, que constituye nuestra representación del conocimiento; y por el otro, un conjunto de reglas mediante las que podemos razonar sobre esa representación del conocimiento, y llegar a nuevas conclusiones que modifiquen los objetos anteriores y que nos permitan resolver problemas.
- b) Interfaz gráfica altamente desarrollada. No se requiere casi teclear nada ni cuando se programa ni cuando se corre el programa, ya que tanto clases, objetos y atributos como funciones, símbolos y comandos pueden ser obtenidos e integrados a reglas, demonios o métodos por medio del mouse. Además, permite desarrollar aplicaciones de gran atractivo para el usuario, manejando incluso dibujos animados.
- c) Capacidad de manejar la información de manera no secuencial y de acuerdo al nivel de conocimiento y a los requerimientos del usuario.
- d) Capacidad de interactuar con bases de datos relacionales.
- e) Confiabilidad. Existen ya un gran número de aplicaciones desarrolladas en este shell.

Tipo de inferencia utilizada

El tipo de inferencia que se usó fue la de *forward chaining* o encadenamiento hacia adelante, ya que se requiere de un razonamiento guiado por los datos o deductivo, pues cada estudiante es diferente (algunos repetirán lecciones mientras otros avanzarán rápidamente).

A partir de los datos de entrada (como la clave del usuario), el sistema decidirá qué conocimiento emplear, de tal manera que el resultado que arroje una decisión, corresponda a los datos de entrada requeridos para tomar la siguiente decisión.

Fuentes de conocimiento utilizadas

Para la adquisición del conocimiento se contó con la colaboración del instructor y experto en el tema, el controlador Luis Manuel Flores Solórzano quien fue la fuente de conocimientos tanto en la parte educativa, es decir, en la manera en que se imparten y evalúan las lecciones; como en la informativa o contenido de las lecciones. La información de cada una de las nueve secciones que comprenden el tutorial fue tomada, en su base, de los manuales de la OACI reeditados por el experto.

Este es un SET basado completamente en el conocimiento de un experto humano y por lo tanto el contenido de las lecciones es de gran valor. Los reactivos de la evaluación previa al tutorial fueron especialmente diseñados por el experto y constituyen también parte importante del tutorial.

Requerimientos del Sistema

Los requerimientos de software y hardware, tanto para la elaboración del SECATA como para su ejecución es la siguiente:

- Computadora personal con procesador 80386 o superior

- 4 Mbytes en RAM como mínimo
- MS-DOS 5.1 o superior
- Windows 3.0 o superior
- LEVEL5 OBJECT 2.5
- Espacio en disco duro para el SECATA y su ejecución de 15 Mbytes
- Mouse

IMPLEMENTACIÓN

Diseño de la Interfaz del usuario

La interfaz del usuario es completamente gráfica, del tal manera que la navegación por las lecciones se realiza completamente por medio de botones. Los reactivos de los exámenes que tiene SECATA se responden casi en su totalidad, haciendo click en la respuesta elegida con el mouse, a excepción de preguntas en las que se requiere la inserción de números o frases.

El diseño de la interfaz del usuario significó el gasto de mucho tiempo ya que la interfaz debió rediseñarse en varias ocasiones. Fueron sometidos a prueba los colores usados en los fondos de pantalla y botones; así como, el tipo de texto empleado en la Información y explicaciones, el tipo de botones y la ubicación de éstos conjuntamente con ventanas de texto, tablas y dibujos en la pantalla.

Los voluntarios que probaron la interfaz mostraron diversos gustos y opiniones; el resultado al conjuntar las recomendaciones fue la Interfaz del usuario que presenta SECATA.

Los botones, ilustraciones y fondos de pantalla fueron elaborados en Paintbrush con ayuda, en casos de animación y detalles, de Corel Draw! que permite la rotación de imágenes sin deformarlas.

Estructura de SECATA

El tutorial se compone de tres partes principales, conformadas por 24 bases de conocimiento o archivos KNB, las cuales se encuentran ligadas por medio de una base de datos con formato de dBASE III, debido a que es este el formato que trabaja LEVEL5:

- Recepción de datos
- Lecciones
- Evaluaciones

La recepción de datos se emplea tanto para guardar el récord del usuario y moverlo dentro de SECATA, así como para manejar la base de datos. El manejo de la base de datos es únicamente permitido por el supervisor, el cual debe introducir su clave al inicio del programa para que éste le permita el borrado de registros, reemplazo de datos de un registro determinado e inserción de registros, principalmente.

Es en la recepción de datos, donde SECATA decide a dónde mandar al usuario actual, por lo tanto, el tutorial siempre iniciará en esta parte y requerirá la clave del usuario para situarlo en la sección que le corresponda, según haya avanzado en el tutorial.

La parte de lecciones contiene todo el conocimiento en Operaciones IFR (parte del control de aproximación) y consta de texto informativo, tablas, dibujos con o sin animación (figura 5-3) y puntos importantes marcados como anotaciones en pizarrón. Las lecciones están divididas en nueve secciones a las cuales se tiene acceso de manera secuencial, esto es, el usuario sólo tendrá acceso a aquellas secciones anteriores al último examen acreditado, esto último con el fin de evitar que emplee material nuevo sin los antecedentes requeridos.

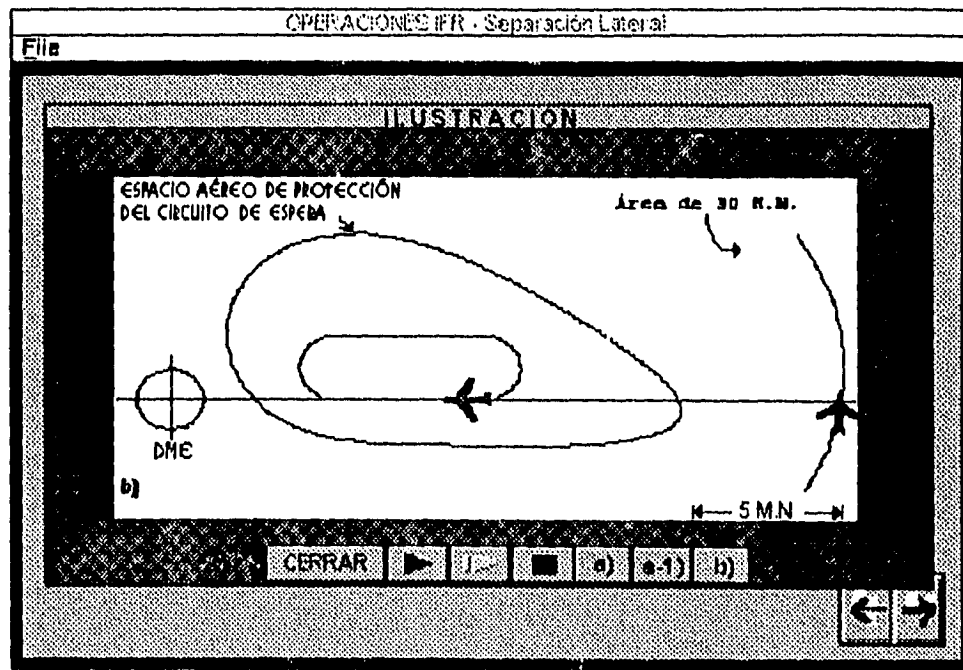


Figura 5 - 3 Muestra de un dibujo con animación en SECATA

Desde cada sección, el usuario puede salirse de la lección para ir al menú principal, el cual le muestra el título de cada sección, una ayuda para el manejo de las lecciones y salida del tutorial. Este menú será diferente para cada usuario, pues solamente se habilitarán los botones, es decir, el acceso a cada sección dependiendo de la acreditación de los exámenes (figura 5-4). La habilitación se muestra con el botón del número de la sección iluminado en un tono rojo, en tanto que el botón aparecerá en gris cuando éste se encuentre deshabilitado.

Para la acreditación de cada sección se cuenta con evaluaciones parciales las cuales debe contestar el usuario correctamente en su totalidad. Estas evaluaciones son de selección y en algunos casos se requiere de más de una respuesta para cada pregunta.

MENU PRINCIPAL			
1	GENERALIDADES	6	SEPARACIÓN LATERAL
2	AUTORIZACIONES	7	SEPARACIÓN LONGITUDINAL
3	SEPARACIÓN VERTICAL	8	ASIGNACIÓN DE RUTAS
4	ASIGNACIÓN DE ALTITUDES	9	COORDINACIÓN
5	VER DE ALTITUD Y REP. DE POSICIÓN		
REGRESAR			

Figura 5 - 4 Menú principal (Operaciones IFR)

Se cuenta también con una evaluación parcial, que consta de dos partes, cada parte debe acreditarse con una calificación igual o superior a ocho en una escala del 1 al 10 y no se tiene acceso a la segunda parte sin antes haber acreditado la primera. Esta evaluación previa, fue diseñada con el propósito de permitir la entrada al tutorial únicamente a aquellos usuarios que cuenten con los conocimientos suficientes en CTA para comprender el contenido del tutorial, ya que este no forma a principiantes, sino a controladores que han perdido contacto con la fase de aproximación (figura 5-5).

Al final del tutorial se aplica una evaluación sumativa, la cual evita que el estudiante abandone el tutorial sin los conocimientos requeridos. Esta evaluación incluye preguntas en inglés y los símbolos más comúnmente usados en CTA.

EVALUACIÓN PREVIA AL CURSO (parte 2)

Ella

19.-Cuál es el valor que deben llevar los dibujos que siguen?

REFERENCIA ART.66 DEL RTA.
Para realizar vuelos VFR, las aeronaves se mantendrán a una distancia de las nubes y en condiciones de visibilidad igual o superior a las indicadas a continuación.

DENTRO DE AEROVÍAS Y ESPACIOS
AÉREOS CONTROLADOS

visibilidad en vuelo kilómetros
 millas náuticas

kilómetros

millas náuticas

metros, o pies

Figura 5 - 5 Muestra de una evaluación del SECATA

APLICACIÓN Y EVALUACIÓN DE RESULTADOS

OPERACIÓN GENERAL DE SECATA

La operación de SECATA es muy fácil e intuitiva debido a su interfaz del usuario.

La forma en como operan los botones de las secciones se encuentra explicada en la parte de AYUDA del menú principal (figura 5-6). Se puede entrar a esta parte una vez acreditadas ambas partes del examen previo.

A continuación se explica la manera en cómo SECATA funciona pantalla por pantalla antes de entrar a las lecciones.

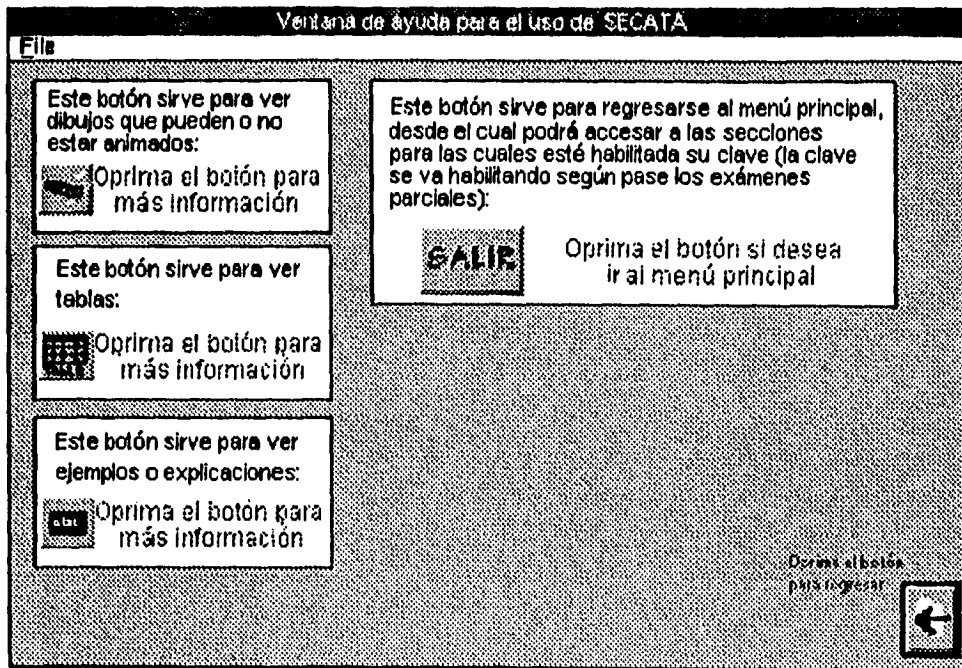


Figura 5 - 6 Muestra de la ayuda de SECATA



Figura 5 - 7 Presentación de SECATA

La primer pantalla que se presenta al correr el tutorial es únicamente la presentación que espera a que el usuario haga *click* con el mouse para continuar (figura 5-7).

Una vez que el usuario ha hecho *click* con el mouse, SECATA mostrará otra pantalla en la que el usuario sólo tendrá que contestar si es la primera vez que entra al tutorial o no, haciendo *click* en uno de los cuatro botones: SI, NO, AYUDA o SALIR, (figura 5-8).

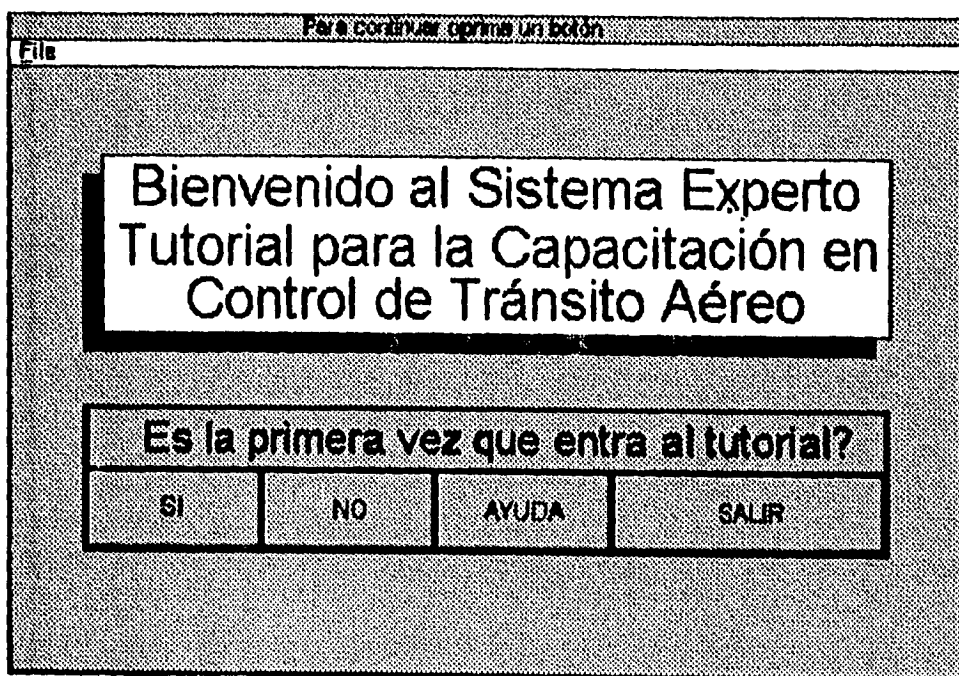


Figura 5 - 8 Pantalla de inicio al tutorial SECATA

Si el usuario oprime el botón SI, SECATA le pedirá sus datos (figura 5-9) y lo mandará a la evaluación previa al tutorial, la cual consta de dos partes; la pantalla inmediata que verá el usuario será la de las instrucciones correspondientes a la parte uno. El requisito para pasar cada una es la obtención a una calificación igual o superior a ocho, en una escala del uno al diez.

The screenshot shows a window titled 'BASE DE DATOS' with a 'File' menu. The main content area is titled 'INTRODUZCA SUS DATOS'. It contains a form with four input fields: 'NOMBRE', 'AP. PATERNO', 'AP. MATERNO', and 'CLAVE'. Below the form are three buttons: 'Datos Correctos', 'Ayuda', and 'Salir'.

Figura 5 - 9 Pantalla de recepción de datos del usuario

The screenshot shows a window titled 'BASE DE DATOS' with a 'File' menu. The main content area displays a large text box with the message: 'Bienvenido al Sistema Experto Tutorial para la Capacitación en Control de Tránsito Aéreo'. Below this, there is a label 'CLAVE' above a text input field. To the right of the input field is a question mark icon. Below the input field is an 'OK' button.

Figura 5 - 10 Inserción de la clave del usuario

En el caso de que el usuario oprima el botón NO, SECATA solicitará la clave de entrada para colocar al usuario en turno en la sección correspondiente (figura 5-10), en la parte del examen previo que no haya contestado o en el examen final. En el caso de que SECATA mande al usuario a una sección en especial, desplegará la pantalla del título de esta y el número al que corresponde y esperará a que el usuario haga click en el botón para continuar (figura 5-11).

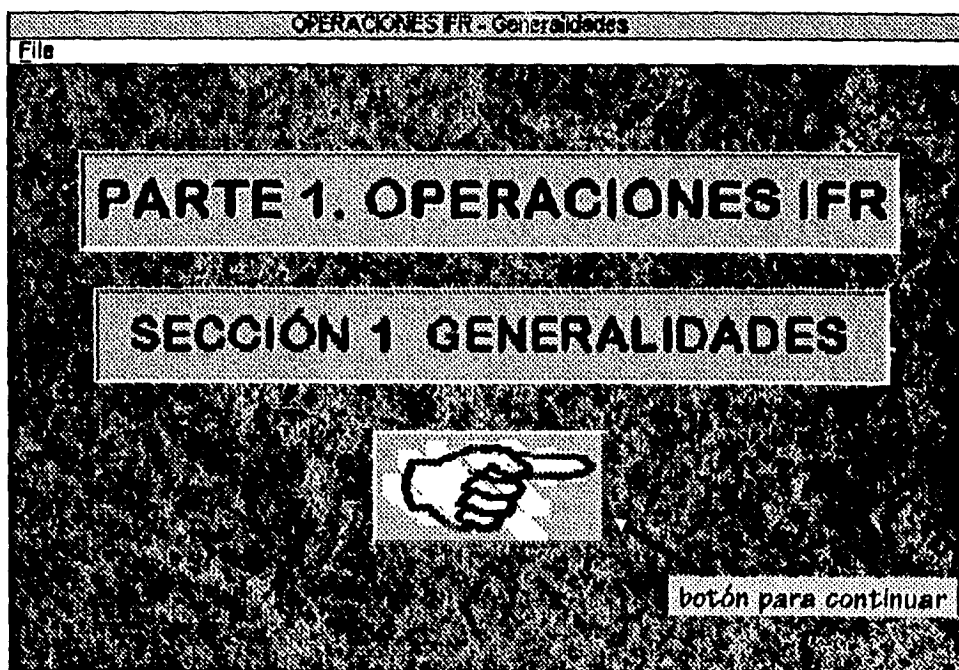


Figura 5 - 11 Pantalla de presentación de una sección

Es posible entrar al tutorial con una clave de supervisor, la cual le dará el privilegio de ver la base de datos y hacer cualquier modificación en ella (figura 5-12). Al introducir la clave de supervisor, SECATA lo enviará inmediatamente a la pantalla que muestra los registros de la base de datos.

En la pantalla de la base de datos para el supervisor se muestra un registro a la vez con los campos siguientes: nombre, apellido paterno y apellido materno, clave, calificaciones de cada examen (-1 cuando no ha pasado por el

examen), fecha de ingreso a SECATA y última fecha en que ocupó el tutorial. El campo marcado con rojo en el tutorial, al cual se le denominó "bandera", contiene el número de registro cuando este usuario está usando el tutorial, este campo se iguala a cero cuando el usuario sale de SECATA.

Los controles que se encuentran a la derecha de la pantalla permiten al supervisor moverse a lo largo del tutorial sin restricciones (S1-S9 para las secciones, E1-E9 para las evaluaciones parciales o formativas, P-P2 para la parte uno y dos de la evaluación previa y F para la evaluación final).

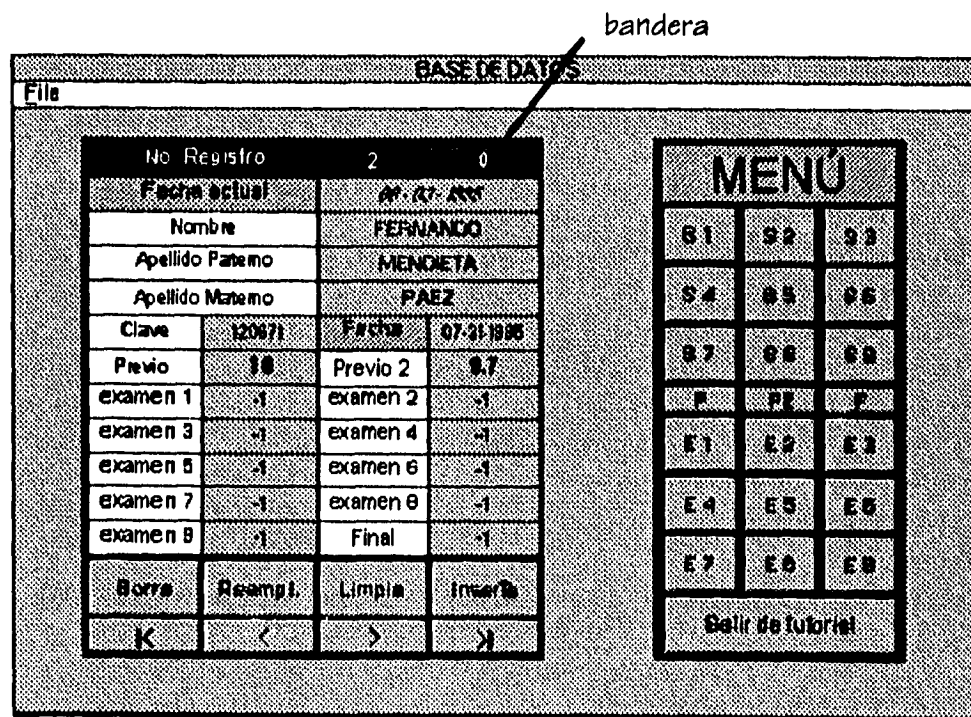


Figura 5 - 12 Base de datos de SECATA y controles del supervisor

Si el supervisor entra a cualquier parte del tutorial desde esta pantalla, será común encontrar una ventana con el título ERROR al moverse hacia un examen o sección, diciendo que se ha encontrado un error con la base de datos (figura 5-13); esta ventana se despliega porque no encuentra la "bandera" encendida y con esto no hará ningún cambio a la base de datos, como lo haría normalmente con la clave de un estudiante.

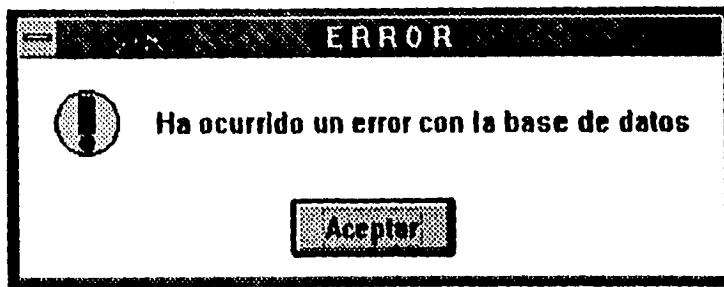


Figura 5 - 13 Ventana de error entrando a la base de datos de SECATA

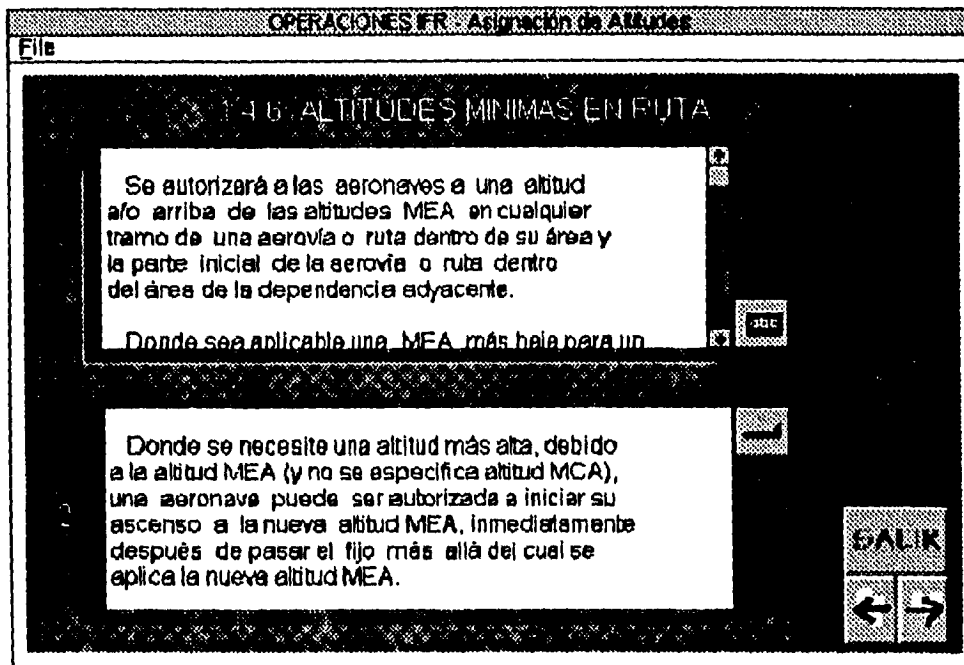


Figura 5 - 14 Pantalla de información de SECATA

VALIDACIÓN DE SECATA

La aplicación de SECATA se llevó a cabo bajo tres propósitos:

- Probar la respuesta del sistema a cualquier eventualidad
- Probar el diseño de la interfaz del usuario

- Probar al sistema como Sistema Experto Tutorial en sí.

Los dos primeros puntos se realizaron con la ayuda de voluntarios, de diferentes edades y áreas de desarrollo profesional, con excelentes resultados. Las únicas dificultades que se encontraron correspondían a la falta de lectura de las instrucciones así como a la falta del uso de la ayuda que proporciona el sistema.

El tercer punto, correspondiente a la aplicación del sistema bajo el punto de vista de SET, se llevó a cabo en dos partes. La primera parte correspondió a la aplicación de SECATA al experto; y la segunda, a su aplicación a un usuario con las características requeridas por el sistema.

El experto manejó el tutorial desde el principio hasta el final, tal y como lo haría un estudiante. Se cronometraron 3 horas y media en el recorrido completo, pero hay que tomar en cuenta que el experto ya conocía el material y lo maneja a la perfección, motivo por el cual no era regresado a sección alguna cuando contestaba los exámenes.

El usuario de la segunda parte, en cambio, tuvo que repetir la evaluación previa tres veces para poder acreditarla, lo cual le tomó dos días completos. Se le fue introduciendo sección a sección y, sin tomar en cuenta el tiempo invertido en las evaluaciones parciales, el usuario empleaba en promedio lo equivalente a un día común de trabajo, en rever cada sección, además de que requería revisar la última sección examinada nuevamente, aunque de manera más rápida, al siguiente día en que entraba a SECATA.

Por otra parte, el usuario se sintió a gusto trabajando en la computadora, pues repetía lecciones tantas veces como él sentía requerirlo, y mostró especial atracción por los gráficos animados, manifestando que aclaraban significativamente la teoría presentada con anterioridad.

El tiempo estimado, que le ocuparía a un usuario final el cubrir en su totalidad a SECATA sería de 2 meses como máximo, y este variará dependiendo de las habilidades del usuario final.

La aplicación de prueba al usuario final aún no termina, pues se requieren de muchas aplicaciones, las cuales llevan meses, para depurar al sistema en

su totalidad. En adición, un sistema nunca queda perfecto en su totalidad y día a día se presentan nuevas ideas para mejorarlo entre las cuales se incluyen la migración del sistema a nuevas versiones de LEVEL5 y la inclusión, a éste, de los multimedios.

DEPURACIÓN DE SECATA

Los SET son programas con características muy particulares y cuyo ciclo de vida difiere de los programas tradicionales.

Muy frecuentemente, el ciclo de vida de un proyecto de SET se descompone en cuatro fases, en cada cual la estructura del SET varía en magnitud y alcance:

- Puesta a punto de una maqueta o demostración, que sólo tiene en cuenta un subconjunto del problema estudiado. Este sistema permite demostrar que el problema es resoluble mediante técnicas de SE y evaluar los recursos necesarios para el conjunto del proyecto.
- Desarrollo de un prototipo que incluye una base de conocimientos que abarca todos los aspectos del problema, pero no forzosamente integrado en el medio real de explotación.
- Integración del producto final en su medio. En efecto, un SE va a integrarse por lo general, en un sistema existente.
- Explotación del sistema. Como el conocimiento suele ser de naturaleza evolutiva, el mantenimiento y la actualización de un SET son operaciones importantes.

Ciclo de vida de SECATA

SECATA aún se encuentra en el primer punto de su ciclo de vida pues es una demostración de la tecnología de los Sistemas Expertos Tutoriales como

medio para capacitar al personal de un área que consta con pocos conocedores del tema y que su enseñanza requiere de ciertas habilidades desarrolladas por un instructor experto en el tema del Control de Tránsito Aéreo.

Todavía hay que implementar mucho material en Control de Aproximación, correspondiente a Servicios de Control de Aproximación para que SECATA sea ya un prototipo. Aún con esta información, el sistema no pararía ahí, pues es posible extenderse a otras materias como meteorología o aeronáutica, o bien, crear un SET hermano ex profeso para el Control de Área.

Una vez integrado SECATA al medio para el cual está diseñado, los usuarios darían sus recomendaciones y con el tiempo surgirían más requerimientos al sistema, tal y como sucede con cualquier paquetería. Será necesario entonces, desarrollar nuevas versiones del sistema con un material más depurado y con tecnología reciente.

Al momento de terminar este trabajo se presenta la posibilidad de introducirlo a los multimedios, quedando así abierto suficiente trabajo para desarrollar la siguiente versión. SECATA tiene mucha vida por delante.

6

CONCLUSIONES

Es posible desarrollar el tutorial en Visual Basic; al igual que el sistema pudo haber sido desarrollado en C, Pascal, o hasta en lenguaje ensamblador; la diferencia de hacerlo en uno u otro radica en que existen herramientas que nos facilitan el trabajo, herramientas diseñadas exclusivamente para elaborar Sistemas Expertos, como es el caso de LEVEL5 OBJECT que ya nos proporciona un motor de inferencia, que tendría que ser programado en el caso de usar Visual Basic (no tomando en cuenta la existencia de librerías) o cualquier otro lenguaje. LEVEL5 proporciona un ambiente propicio para el fácil desarrollo de la aplicación. Una vez creada la aplicación, el usuario final solamente requerirá de la o las bases de conocimiento (archivos con extensiones KNB) y el sistema run-only para poner en marcha la aplicación.

Existen dos aspectos particularmente difíciles en el diseño de SET: la conjunción con el experto y el diseño de la interfaz del usuario.

La conjunción del ingeniero del conocimiento con el experto siempre resulta difícil, pues el experto carece, generalmente, de los conocimientos en el área de los Sistemas Expertos y el ingeniero del conocimiento, a su vez, no posee los conocimientos especializados del experto. Lo anterior provoca que en la primera entrevista, ambos se encuentren hablando en términos diferentes y que ninguna de las partes entienda claramente los requerimientos de la otra; tal parecería que ambos estuvieran hablando en idiomas diferentes pues el lenguaje que usan, para expresar hechos y sus relaciones, es frecuentemente limitado y no es comprensible para otros.

El diseño de la interfaz del usuario es una labor en extremo difícil y laboriosa pues se requieren conocimientos básicos de Diseño Gráfico, principalmente; de Pedagogía, en el caso de tutoriales; y Control de Tránsito Aéreo, para este SET, en particular.

El propósito de un buen diseño de la interfaz del usuario, es el de establecer, de la forma más adecuada, la comunicación de la información, lo cual conlleva a buscar ayuda de otras disciplinas. El Diseño Gráfico, nos indica, por ejemplo, la manera de elaborar las pantallas de acuerdo al grado de importancia de cada uno de los objetos que aparecerán en ellas, de tal manera que las pantallas se encuentren equilibradas; otro ejemplo, es el manejo de los colores que llevarán las pantallas así como el tipo de letras, o fuentes, que deberán usarse en los textos.

La Pedagogía, por su parte, nos indica la manera en cómo debe ser transmitido el conocimiento para que el alumno lo reciba más rápidamente y de manera efectiva. Así, la forma en como se enseñe el conocimiento, dependerá de factores tales como la edad del estudiante, la fase del ciclo del aprendizaje, en la cual se encuentre éste y, al nivel de conocimiento al cual se le quiera llevar.

Por lo anterior, podemos concluir que el desarrollo de un SET por una sola persona es una labor difícil, por tanto, es necesaria la conformación de grupos multidisciplinarios para desarrollar aplicaciones más completas y efectivas.

El uso de SET aporta grandes ventajas. SECATA incrementará el rendimiento y la productividad de expertos en CTA que ocupaban bastante tiempo en la instrucción; en adición, mejorará la calidad de la capacitación de controladores de tránsito aéreo, ya que este sistema incorpora conocimientos especializados, tanto en CTA como en pedagogía.

SECATA, además de ser un almacén de conocimiento que es propio de ciertos controladores, sostiene el conocimiento especializado de un controlador de tránsito aéreo como instructor, el cual no se encontrará en ningún otro lado. Si a lo anterior añadimos que SECATA puede ser transportado a cualquier lugar que cuente con los requerimientos especificados, el sistema se convierte en una herramienta poderosa para asistir a la capacitación de controladores.

El comportamiento inteligente de SECATA se observa en que toma los datos de entrada y empieza a tomar decisiones, pues no se sabe a dónde ira el estudiante (no se conocen las conclusiones).

SECATA es sensible de recibir mejoras. Dadas las tendencias actuales, se le puede incorporar la tecnología de los multimedios por medio de la versiones actualizadas de LEVEL5 OBJECT, o bien, dar más realce a las pantallas mediante efectos de tercera dimensión. Es factible también, la implantación de una simulación (con lo cual no me refiero al radar); así como la integración de las opiniones de varios expertos para incrementar la calidad de las lecciones.

Por otra parte, SECATA puede incluir nuevas materias como Meteorología, siguiendo el enfoque del Control de Aproximación. Es posible también desarrollar una parte completa para el Control de Área, bajo el mismo enfoque que se presenta en esta tesis.

SECATA es, igualmente, una herramienta para despertar el interés tanto de ingenieros en computación, como de los responsables de la instrucción de controladores de tránsito aéreo. Los primeros, para que giren hacia un área donde pocos tienen puestos los ojos: los controladores de tránsito aéreo. - Existen muchos simuladores de vuelo dedicados a los pilotos, así como software diseñado para la navegación aérea, pareciera que se han olvidado de

la educación de los controladores - ; y los responsables de la formación de controladores de tránsito aéreo, para que introduzcan nuevas formas para capacitar a su personal de una manera efectiva, aumentando la calidad de la enseñanza del CTA y mirando hacia la tecnología computacional.

GLOSARIO

Aeródromo. Cualquier área sobre la tierra o agua puesta aparte o comúnmente usada para proporcionar facilidades para el aterrizaje y salida de una aeronave.

Aerovía. Un control de área o parte de un control de área establecido en la forma de un corredor equipado con ayudas de radio navegación.

Algoritmo. Procedimiento para ejecutar según un orden determinado, un conjunto de operaciones elementales, suficientes para la resolución de todos los problemas de una clase. Totalidad de pasos de procesamiento cuya ejecución, en orden predeterminado, termina en la solución de una tarea concreta.

Altitud. La distancia vertical de un nivel, un punto u objeto considerado como un punto, medida desde el nivel medio del mar.

Ambiente. Un ambiente actúa como mediador gráfico entre el usuario y el programa.

Antecedent. Ver Antecedente.

Antecedente. La condición de soporte o premisa de una regla o demonio. Un antecedente es la parte IF de una regla IF-THEN del tipo de inferencia de encadenamiento hacia atrás o hacia adelante y puede ser una orden o expresión. Cuando todos los antecedentes de una regla están verificados, la conclusión (declaración THEN) es alcanzada y se dice que la regla o demonio está encendida.

Aplicación. Todos los componentes de software que son usados para resolver un problema determinado. Una aplicación de LEVEL5 OBJECT incluye una o

más bases de conocimiento y sus programas de soporte externo, bases de datos y archivos de texto.

Application. Ver Aplicación.

Atributo. Define las cualidades de una clase y su tipo de información. Un atributo consiste de un nombre, un valor, un factor propio y las facetas, reglas y demonios asociados. Todos los atributos pueden ser declarados como simples o como arreglos.

Attribute. Ver Atributo.

Backward-chaining. Ver Encadenamiento hacia atrás.

Base de conocimientos. Una colección estructurada de clases, atributos, reglas de encadenamiento hacia atrás, demonios de encadenamiento hacia adelante, agenda, visualizaciones y bases de datos orientadas a objetos creadas por el programador de la aplicación. Una aplicación puede consistir de bases de conocimientos múltiples y relacionadas.

Caja de diálogo. Un modo de diálogo que pide al usuario más información o permite conocer acciones que están siendo realizadas o elaboradas. Las cajas de diálogo, son modales dado que el usuario debe indicar una respuesta en orden para continuar.

Cálculo de predicados. Lógica de predicados, lógica de primer orden. Notación fundamental para representar y razonar con sentencias lógicas.

Clase. Define la estructura y comportamiento de un objeto. Una declaración de clases es una definición estructural similar a la estructura de un récord o registro de una base de datos. La descripción, pues, de los nombres y tipos de los campos de la base de datos es análoga a la descripción del atributo contenido en la clase. Distinto a las bases de datos; sin embargo, las clases pueden contener información acerca de cómo obtener los valores de sus atributos a través de reglas, demonios o métodos.

Clases de usuario. Las clases que están definidas por el programador durante el desarrollo de la aplicación.

Class. Ver Clase.

Click. Presionar y soltar el botón izquierdo del mouse rápidamente cuando el puntero (flecha en pantalla que indica la posición del mouse) se encuentra en el lugar seleccionado.

Conclusión. Es la declaración THEN que concluye una regla o demonio. Todas las conclusiones son introducidas mediante la palabra reservada THEN. Una regla o demonio puede tener múltiples conclusiones y conclusiones alternas, cláusulas ELSE que se ejecutan cuando la conclusión primaria no es alcanzada. En LEVEL5 OBJECT, una conclusión puede ser una orden o una referencia de atributo simple, compuesta o multicompuesta.

Debugging. Ver Depuramiento.

Demon. Ver Demonio.

Demonio. Representa una regla IF-THEN del encadenamiento hacia adelante. Los demonios son evaluados por el Motor de Inferencia cuando el valor de un atributo en las declaraciones IF de los demonios, antecedentes, cambia dentro de la ejecución de la aplicación. Los demonios continuarán siendo evaluados hasta que nuevos atributos referenciados en la parte IF del demonio cambien de valor. Este proceso de razonamiento, es llamado encadenamiento hacia adelante o razonamiento de datos impulsados.

Depuramiento. Es el proceso de detectar errores en una Base de Conocimientos. En LEVEL5 OBJECT, el proceso de depurado es facilitado por el sistema de razonamiento que consiste de Session Monitor, Values List, History Report y Knowledge Tree. Estas herramientas proveen acceso al proceso de razonamiento inferencial durante una sesión de la Base de Conocimientos y permiten controlar y observar la ejecución de la base de datos.

Dialog box. Ver Caja de diálogo.

Encadenamiento hacia adelante. Comienza con hechos o datos conocidos y deduce nuevos acerca de la situación basada en la información contenida en la Base de Conocimientos. El proceso continúa hasta que no sean deducidas

nuevas conclusiones de los datos iniciales. El proceso es desencadenado por eventos iniciales.

Encadenamiento hacia atrás. Comienza con una meta deseada u objetivo y procede hacia atrás a lo largo de una cadena de razonamiento en un intento de tomar la información necesitada para verificar la meta. En LEVEL5 OBJECT, una cadena de razonamiento puede consistir de una serie de declaraciones IF-THEN llamadas reglas, procedimientos llamados métodos, valores por defecto y dudas del usuario final. El orden de la búsqueda de la meta define la combinación específica de estos ítems y determina cómo y en qué orden el Motor de Inferencia intentará verificar la meta.

Epistemología. Parte de la filosofía que se interesa por la verdad, por el problema del método, y por la posibilidad, origen y esencia del conocimiento.

Espacio aéreo controlado. Un espacio aéreo de dimensiones definidas dentro de las cuales el servicio de control de tránsito aéreo es proporcionado a vuelos IFR.

Facet. Ver Faceta.

Faceta. Controla la manera en la cual el Motor de Inferencia y el sistema de visualización procesan y evalúan un atributo. Dentro de LEVEL5 OBJECT hay tres tipos básicos de facetas: valor, aviso e inferencia.

Forward-chaining. Ver Encadenamiento hacia adelante.

Grupo de demonios. Conjunto que hace referencia al mismo atributo en sus antecedentes. LEVEL5 OBJECT evalúa un grupo de demonios basado en la estrategia de demonios, que está en efecto al tiempo que los demonios son impulsados.

Grupo de reglas. Grupo de encadenamiento hacia atrás que concluye el mismo atributo.

Hecho. Expresión considerada como válida.

Herencia. Proceso por el cual una clase llamada padre transfiere su estructura y comportamiento a otra clase llamada hijo. Una clase puede heredar la estructura y forma lógica de más de un padre (herencia múltiple).

Heurístico. Proceso creador que a partir de un conjunto de operaciones elementales suficientes, resuelve todos los problemas de una clase.

Impulsión. Llamada Fire, es la ejecución lograda de la conclusión de una regla, demonio o método por el Motor de Inferencia.

Ingeniería del conocimiento. Subdisciplina de la inteligencia artificial ocupada en construir sistemas expertos adquiriendo el conocimiento de un área específica y estructurándolo en la base de conocimientos. La mayor parte de ese conocimiento se obtiene de personas expertas.

Instancia. Una específica ocurrencia de un objeto. Un objeto consiste de su estructura de clases, la cual define su atributo y comportamiento, y sus instancias, las cuales tienen los valores actuales del objeto. Todas las instancias tienen la misma estructura de la clase o clases a las cuales pertenecen, pero también contienen valores únicos.

Knowledge Base. Ver Base de Conocimientos.

Lenguaje de Producción de Reglas. PRL. La presentación del lenguaje del conocimiento usado en LEVEL5 OBJECT para crear aplicaciones.

Lógica formal. Estudio del análisis de proposiciones y de pruebas, prestando atención, únicamente, a símbolos y formas abstractas y no al significado de las abstracciones.

Method. Ver Método.

Método. Usado para determinar un valor de un atributo asociado o ejecuta una serie de procedimientos. WHEN NEEDED son ejecutados durante el encadenamiento hacia atrás para determinar el valor del atributo. WHEN CHANGED son para encadenamientos hacia adelante e implementan un procedimiento cuando el valor de un atributo cambia durante la ejecución de la aplicación.

Motor de inferencia. El componente que ejecuta e interpreta una Base de Conocimientos. LEVEL5 OBJECT cuenta con dos tipos: un motor que procesa los datos hacia adelante y otro que los procesa hacia atrás.

Object. Ver Objeto.

Objeto. Una estructura de datos que representa cualquier cosa tangible o abstracta. Una estructura de un objeto es definida por sus clases y declaraciones de atributos. Una declaración de clases es una plantilla que define la estructura de un objeto (como la estructura de un registro o récord de una base de datos). La estructura de la clase adicional define las características del objeto como atributos (similar a los campos de una base de datos). Procedimientos y comportamientos son limitados a atributos en la forma de facetas, métodos, reglas y demonios. El actual dato de una aplicación basada en objetos es representado como los tenidos por instancias de objetos individuales.

Pie. Medida de longitud que equivale a 30.5 centímetros.

Plan de vuelo. Información especificada, proporcionada a unidades de servicios de tránsito aéreo relativas al destino de un vuelo o porción de un vuelo de una aeronave.

Procedimiento de aproximación por instrumentos. Una serie de maniobras predeterminadas para la transferencia ordenada de una aeronave bajo condiciones de vuelo por instrumentos desde el comienzo de una aproximación inicial hasta un aterrizaje, o hasta un punto desde el cual un aterrizaje puede ser hecho visualmente.

Programación orientada a objetos. Programación que se utiliza comúnmente en la elaboración de Sistemas Expertos.

Regla. Usada para codificar estrategias de inferencia de encadenamiento hacia atrás. Las reglas son declaraciones IF-THEN y consisten de un antecedente o premisa (IF) que es evaluada por el Motor de Inferencia con el encadenamiento hacia atrás en un intento de verificar la conclusión de la regla (THEN).

Rule. Ver Regla.

Run Time. El proceso de corrida, como opuesto al desarrollamiento de una aplicación.

Seleccionar. Para seleccionar un objeto o región, dirija el puntero del mouse hacia el objeto o región, posicionándolo sobre él, y haga click.

Separación no-radar. La separación usada cuando la información de la posición de una aeronave es derivada de otras fuentes diferentes al radar.

Sesión. El periodo en el cual la aplicación está corriendo.

Sistema de clases. Las clases predefinidas por LEVEL5 OBJECT que permite al desarrollador de la aplicación interactuar y controlar el ambiente de la aplicación. En los editores del RULE TALK y desde los métodos, reglas y demonios de una Base de Conocimientos, el desarrollador o programador puede hacer referencia y cambiar atributos del sistema de clases, permitiendo control de la interfase de usuario, bases de datos y el Motor de Inferencia por sí.

Sistemas expertos. Programas de computadora contruidos para aplicaciones comerciales utilizando las técnicas de programación de la inteligencia artificial, especialmente aquellas desarrolladas para resolver problemas.

Tiempo estimado de llegada. Tiempo en el cual el piloto estima que la aeronave estará sobre una localidad especificada.

Tipo de atributo. Describe el tipo de información representada por un atributo de una clase. En LEVEL5 OBJECT, los tipos de atributos soportados son COMPOUND, MULTICOMPOUND, SIMPLE, NUMERIC, STRING, PICTURE, RECTANGLE, COLOR, TIME e INTERVAL.

User classes. Ver Clases de usuario.

Usuario final. Una persona que usa la aplicación. No es el programador.

Visibilidad. La habilidad, determinada por condiciones atmosféricas y expresada en unidades de distancia, para ver e identificar objetos sin luz eminentes en el día y objetos iluminados eminentes en la noche.

Vuelo VFR especial. Un vuelo VFR controlado autorizado por el control de tránsito aéreo para operar dentro de una zona de control de acuerdo con condiciones meteorológicas bajo las condiciones meteorológicas visuales.

BIBLIOGRAFÍA

APLIQUE TURBO PROLOG

Phillip R. Robinson
edit. Osborne/McGraw-Hill
España, 1990

APRENDER A PENSAR LEYENDO BIEN

Habilidades de lectura a nivel superior
Yolanda Argudín y María Luna
edit. Plaza y Valdés y Universidad Iberoamericana
México, 1994

ARTIFICIAL INTELLIGENCE AND TUTORING SYSTEMS

Computational and Cognitive Approaches to the Communication of Knowledge
Etienne Wenger
edit. Morgan Kaufmann Publishers Inc.
USA, 1987

CONSTRUCTIVISM IN THE COMPUTER AGE

George Forman & Peter B. Pufall
edit. Lawrence Erlbaum Associates Inc.
USA, 1988

DEVELOPING EXPERT SYSTEMS

A knowledge engineer's handbook for rules & objects
Edmun C. Payne, Robert C. Mc Arthur
edit. John Wiley & Sons Inc.
USA, 1990

DIDÁCTICA PARA LOS PROFESORES DE LA ESCUELA
BANCARIA Y COMERCIAL

Juan M. Robredo
EBC
Mecanograma
México, 1991

EL ORDENADOR EN LA ENSEÑANZA

Análisis y perspectivas del futuro
A. Bork
edit. Gustavo Gili S.A.
España, 1986

EXPERT SYSTEM TECHNOLOGY

Development & Application
Robert Keller
edit. Yourdon Press-Prentice Hall
USA, 1987

INTERNATIONAL AIR TRAFFIC CONTROL

Management of the World's Airspace
Arnold Field Obe
edit. Pergamon Press
Gran Bretaña, 1985

INTRODUCCIÓN A LOS SISTEMAS EXPERTOS

Ramón Galán López
Universidad Politécnica de Madrid
E.T. Superior de Ingenieros Industriales
Sección de publicaciones
Madrid, 1990

LOS PORQUÉS DE LA MENTE HUMANA

Reader's Digest México, S.A. de C.V.
México, 1991

PC WORLD

"Inteligencia Artificial y Sistemas Expertos" por Gabriel Contreras Mejuto
Octubre 1988

PRINCIPIOS DE INTELIGENCIA ARTIFICIAL Y SISTEMAS
EXPERTOS

David W. Rolston
edit. McGraw-Hill
México, 1991

SISTEMAS EXPERTOS

Introducción a la técnica y a la aplicación
klaus Bauer, Irmgard Büttel, Lutz Eberhard, Margret Hälker y otros
editor Dieter Nebendahl
edit. Siemens Aktiengesellschaft & Marcombo
España, 1988

SISTEMAS EXPERTOS

Una Metodología de Programación
J.P. Sánchez y Beltrán
edit. Macrobit
México, 1990

A 1

LISTA DE SIGLAS Y ABREVIATURAS

CIAAC. Centro Internacional de Adiestramiento de Aviación Civil

CTA. Control de Tránsito Aéreo

FIR. Flight Information Region

IA. Inteligencia Artificial

ICAI. Intelligent Computer-Aided Instruction (Instrucción Inteligente Apoyada en una Computadora)

ICAO. International Civil Aviation Organization (Organización de Aviación Civil Internacional)

IFR. Instrument Flight Rules

ITS. Intelligent Tutoring System (Sistema Tutorial Inteligente)

kt. Knots (nudos)

mph. Statute miles per hour (millas estatutas por hora)

NDB. Radio faro no direccional

NM. Nautical miles (millas náuticas)

PAR. Precision Approach Radar

PRL. Production Rule Language (Lenguaje de Producción de Reglas)

SE. Sistema Experto

SENEAM. Servicios a la Navegación en el Espacio Aéreo Mexicano

SET. Sistema Experto Tutorial

SECATA. Sistema Experto Tutorial para la Capacitación en Control de Tránsito Aéreo

SRA. Surveillance Radar Approach

UIR. Upper Flight Information Region

VFR. Visual Flight Rules

A2

LISTADO REDUCIDO DE SECATA

DATOS.KNB

CLASS database actions

WITH copia dbase a display SIMPLE

WHEN CHANGED

BEGIN

nombre := nombre OF dB3 USR 1
paterno := paterno OF dB3 USR 1
materno := materno OF dB3 USR 1
passwd := passwd OF dB3 USR 1
previo := previo OF dB3 USR 1
previo2 := previo2 OF dB3 USR 1
ex1 := ex1 OF dB3 USR 1
ex2 := ex2 OF dB3 USR 1
ex3 := ex3 OF dB3 USR 1
ex4 := ex4 OF dB3 USR 1
ex5 := ex5 OF dB3 USR 1
ex6 := ex6 OF dB3 USR 1
ex7 := ex7 OF dB3 USR 1
ex8 := ex8 OF dB3 USR 1
ex9 := ex9 OF dB3 USR 1
final := final OF dB3 USR 1
bandera := bandera OF dB3 USR 1

END

WITH copia display a dbase SIMPLE

WHEN CHANGED

BEGIN

bandera OF dB3 USR 1 := bandera
passwd OF dB3 USR 1 := passwd
nombre OF dB3 USR 1 := nombre
paterno OF dB3 USR 1 := paterno
materno OF dB3 USR 1 := materno

```
previo OF dB3 USR 1 := previo
previo2 OF dB3 USR 1 := previo2
ex1 OF dB3 USR 1 := ex1
ex2 OF dB3 USR 1 := ex2
ex3 OF dB3 USR 1 := ex3
ex4 OF dB3 USR 1 := ex4
ex5 OF dB3 USR 1 := ex5
ex6 OF dB3 USR 1 := ex6
ex7 OF dB3 USR 1 := ex7
ex8 OF dB3 USR 1 := ex8
ex9 OF dB3 USR 1 := ex9
final OF dB3 USR 1 := final
ultimo OF dB3 USR 1 := ultimo
END
```

```
CLASS procedimientos
WITH acepta datos SIMPLE
WHEN CHANGED
BEGIN
  beep OF beeper := TRUE
  FIND dB3 USR 1
  LIMIT 1
  WHERE passwd OF dB3 USR 1 = passwd
  WHEN FOUND
    ASK incorrecto
  WHEN NONE FOUND
    action OF dB3 USR 1 IS append record := TRUE
    nombre OF dB3 USR 1 := nombre
    paterno OF dB3 USR 1 := paterno
    materno OF dB3 USR 1 := materno
    passwd OF dB3 USR 1 := passwd
    previo OF dB3 USR 1 := -1
    previo2 OF dB3 USR 1 := -1
    ex1 OF dB3 USR 1 := -1
    ex2 OF dB3 USR 1 := -1
    ex3 OF dB3 USR 1 := -1
    ex4 OF dB3 USR 1 := -1
    ex5 OF dB3 USR 1 := -1
    ex6 OF dB3 USR 1 := -1
    ex7 OF dB3 USR 1 := -1
    ex8 OF dB3 USR 1 := -1
    ex9 OF dB3 USR 1 := -1
    final OF dB3 USR 1 := -1
    bandera OF dB3 USR 1 := record OF dB3 USR 1 + 1
    inicio OF dB3 USR 1 := NOW
```

```
ultimo OF dB3 USR 1 := NOW
action OF dB3 USR 1 IS pack := TRUE
action OF dB3 USR 1 IS close := TRUE
CHAIN "previo.knb"
FIND END
END
WITH sale de sistema SIMPLE
WHEN CHANGED
BEGIN
beep OF beeper := TRUE
action OF dB3 USR 1 IS close := TRUE
exit OF application := TRUE
END
WITH ayuda registro SIMPLE
WHEN CHANGED
BEGIN
beep OF beeper := TRUE
items[ 5] OF help display := hyperregion 1
items[ 6] OF help display := su
items[ 7] OF help display := hyperregion 2
items[ 8] OF help display := usr
output OF help window := help display
visible OF help window := TRUE
END
WITH sale de ayuda SIMPLE
WHEN CHANGED
BEGIN
visible OF help window := FALSE
items[ 4] OF help display := bluetext ayuda
END
WITH ok passwd SIMPLE
WHEN CHANGED
BEGIN
beep OF beeper := TRUE
FIND dB3 USR 1
LIMIT 1
WHERE passwd OF dB3 USR 1 = passwd
WHEN FOUND
bandera OF dB3 USR 1 := record OF dB3 USR 1
ultimo OF dB3 USR 1 := NOW
mandalo OF procedimientos := TRUE
WHEN NONE FOUND
ASK Incorrecto
visible OF introduce clave := FALSE
FIND END
```



```
END
WITH mandalo SIMPLE
WHEN CHANGED
BEGIN
  IF passwd OF dB3 USR 1 = 70470 THEN
    BEGIN
      bandera OF dB3 USR 1 := 0
      supervisor := TRUE
    END
  ELSE
    IF final OF dB3 USR 1 >= 8 THEN
      CHAIN "menu.knb"
    ELSE
      IF ex9 OF dB3 USR 1 = 10 THEN
        CHAIN "final.knb"
      ELSE
        IF ex8 OF dB3 USR 1 = 10 THEN
          CHAIN "pl59.knb"
        ELSE
          IF ex7 OF dB3 USR 1 = 10 THEN
            CHAIN "pl58.knb"
          ELSE
            IF ex6 OF dB3 USR 1 = 10 THEN
              CHAIN "pl57.knb"
            ELSE
              IF ex5 OF dB3 USR 1 = 10 THEN
                CHAIN "pl56.knb"
              ELSE
                IF ex4 OF dB3 USR 1 = 10 THEN
                  CHAIN "pl55.knb"
                ELSE
                  IF ex3 OF dB3 USR 1 = 10 THEN
                    CHAIN "pl54.knb"
                  ELSE
                    IF ex2 OF dB3 USR 1 = 10 THEN
                      CHAIN "pl53.knb"
                    ELSE
                      IF ex1 OF dB3 USR 1 = 10 THEN
                        CHAIN "pl52.knb"
                      ELSE
                        IF previo2 OF dB3 USR 1 >= 8 THEN
                          CHAIN "menu.knb"
                        ELSE
                          IF previo OF dB3 USR 1 >= 8 THEN
                            CHAIN "previo2.knb"
```

```
        ELSE
            CHAIN "previo.knb"
        END
WITH ve a registro SIMPLE
    WHEN CHANGED
        BEGIN
            title OF main window := "BASE DE DATOS"
            beep OF beeper := TRUE
            ASK registro de usuarios
        END
WITH ve por clave SIMPLE
    WHEN CHANGED
        BEGIN
            title OF main window := "BASE DE DATOS"
            beep OF beeper := TRUE
            output OF introduce clave := pide clave
            visible OF introduce clave := TRUE
        END
WITH ayuda inicio SIMPLE
    WHEN CHANGED
        BEGIN
            beep OF beeper := TRUE
            items[ 4] OF help display := texto ayuda
            text OF texto ayuda := ayuda inicio OF textos 1
            output OF help window := help display
            visible OF help window := TRUE
        END
WITH sale de registro SIMPLE
    WHEN CHANGED
        BEGIN
            beep OF beeper := TRUE
            action OF dB3 USR 1 IS close := TRUE
            variable text OF mensaje salida := nombre
            ASK mensaje salida
            exit OF application := TRUE
        END
WITH borra reg SIMPLE
    WHEN CHANGED
        BEGIN
            action OF dB3 USR 1 IS delete record := TRUE
            action OF dB3 USR 1 IS pack := TRUE
            copla dbase a display OF database actions := TRUE
        END
WITH reemplaza reg SIMPLE
    WHEN CHANGED
```

```
BEGIN
  copia display a dbase OF database actions := TRUE
  record OF dB3 USR 1 := record OF dB3 USR 1
END
WITH llmpla reg SIMPLE
  WHEN CHANGED
  BEGIN
    nombre := ""
    paterno := ""
    materno := ""
    passwd := record OF dB3 USR 1
    previo := -1
    previo2 := -1
    ex1 := -1
    ex2 := -1
    ex3 := -1
    ex4 := -1
    ex5 := -1
    ex6 := -1
    ex7 := -1
    ex8 := -1
    ex9 := -1
    final := -1
  END
WITH inserta reg SIMPLE
  WHEN CHANGED
  BEGIN
    copia display a dbase OF database actions := TRUE
    action OF dB3 USR 1 IS insert record := TRUE
    copia dbase a display OF database actions := TRUE
  END
WITH primero SIMPLE
  WHEN CHANGED
  BEGIN
    record OF dB3 USR 1 := 1
    copia dbase a display OF database actions := TRUE
  END
WITH back SIMPLE
  WHEN CHANGED
  BEGIN
    IF record OF dB3 USR 1 = 1 THEN
      record OF dB3 USR 1 := size OF dB3 USR 1
    ELSE
      record OF dB3 USR 1 := record OF dB3 USR 1 - 1
      copia dbase a display OF database actions := TRUE
    END IF
  END
```

```

END
WITH next SIMPLE
  WHEN CHANGED
  BEGIN
    IF record OF dB3 USR 1 = size OF dB3 USR 1 THEN
      record OF dB3 USR 1 := 1
    ELSE
      action OF dB3 USR 1 IS advance := TRUE
      copia dbase a display OF database actions := TRUE
    END
  END
WITH ultimo SIMPLE
  WHEN CHANGED
  BEGIN
    record OF dB3 USR 1 := size OF dB3 USR 1
    copia dbase a display OF database actions := TRUE
  END
WITH entra SIMPLE
  WHEN CHANGED
  BEGIN
    beep OF beeper := TRUE
    title OF main window := "Para continuar oprima un botón"
  END

```

```

DEMON para mostrar info supervisor
IF info supervisor OF procedimientos
THEN text OF texto ayuda := supervisor OF textos 1
AND items[ 4] OF help display := texto ayuda

```

```

DEMON para mostrar info usuario
IF info usuario OF procedimientos
THEN text OF texto ayuda := usuario OF textos 1
AND items[ 4] OF help display := texto ayuda

```

MENU.KNB

```

ATTRIBUTE ve a parte1 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK Parte 1
  END

```

```

ATTRIBUTE ayuda SIMPLE
  WHEN CHANGED
  BEGIN

```

```
CHAIN "ayuda.knb"  
END
```

```
ATTRIBUTE salir del tutorial SIMPLE  
WHEN CHANGED  
BEGIN  
  bandera OF dB3 USR 1 := 0  
  exit OF application := TRUE  
END
```

```
ATTRIBUTE empieza aqui SIMPLE  
ATTRIBUTE habilita botones SIMPLE  
WHEN CHANGED  
BEGIN  
  IF ex1 OF dB3 USR 1 = 10 THEN  
    enabled OF pictbtn 2 := TRUE  
  IF ex2 OF dB3 USR 1 = 10 THEN  
    enabled OF pictbtn 3 := TRUE  
  IF ex3 OF dB3 USR 1 = 10 THEN  
    enabled OF pictbtn 4 := TRUE  
  IF ex4 OF dB3 USR 1 = 10 THEN  
    enabled OF pictbtn 5 := TRUE  
  IF ex5 OF dB3 USR 1 = 10 THEN  
    enabled OF pictbtn 6 := TRUE  
  IF ex6 OF dB3 USR 1 = 10 THEN  
    enabled OF pictbtn 7 := TRUE  
  IF ex7 OF dB3 USR 1 = 10 THEN  
    enabled OF pictbtn 8 := TRUE  
  IF ex8 OF dB3 USR 1 = 10 THEN  
    enabled OF pictbtn 9 := TRUE  
END
```

```
ATTRIBUTE encuentra al usuario SIMPLE  
WHEN CHANGED  
BEGIN  
  FIND dB3 USR 1  
  LIMIT 1  
  WHERE bandera OF dB3 USR 1 <> 0  
  WHEN FOUND  
    habilita botones := TRUE  
  WHEN NONE FOUND  
    ASK problemas  
  FIND END  
END
```

DEMON para evaluacion del menu
IF empieza aqui
THEN encuentra al usuario := TRUE

PREVIO.KNB

ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
IF q1 OF preguntas IS a THEN
n := 1
IF q1 OF preguntas IS b THEN
n := n + 1
IF q1 OF preguntas IS c THEN
n := n + 1
IF q1 OF preguntas IS d THEN
n := n - 1
IF q1 OF preguntas IS e THEN
n := n - 1
IF q1 OF preguntas IS f THEN
n := n - 1
IF n = 2 THEN
total := 1
IF n = 3 THEN
total := 2
n := 0
IF q2 OF preguntas IS a THEN
n := 1
IF q2 OF preguntas IS b THEN
n := n + 1
IF q2 OF preguntas IS c THEN
n := n + 1
IF q2 OF preguntas IS d THEN
n := n - 1
IF q2 OF preguntas IS e THEN
n := n - 1
IF q2 OF preguntas IS f THEN
n := n - 1
IF n = 2 THEN
total := 1
IF n = 3 THEN
total := total + 2
n := 0
IF q3 OF preguntas IS a THEN

```
n := 1
IF q3 OF preguntas IS b THEN
  n := n + 1
IF q3 OF preguntas IS c THEN
  n := n + 1
IF q3 OF preguntas IS d THEN
  n := n - 1
IF q3 OF preguntas IS e THEN
  n := n - 1
IF q3 OF preguntas IS f THEN
  n := n - 1
IF n = 2 THEN
  total := total + 1
IF n = 3 THEN
  total := total + 2
n := 0
IF q4 OF preguntas IS a THEN
  n := 1
IF q4 OF preguntas IS b THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q5 OF preguntas IS b THEN
  n := 1
IF q5 OF preguntas IS a THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q6 OF preguntas IS a THEN
  n := 1
IF q6 OF preguntas IS b THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q7 OF preguntas IS a THEN
  n := 1
IF q7 OF preguntas IS b THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q8 OF preguntas IS b THEN
```

```
n := 1
IF q8 OF preguntas IS a THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q9 OF preguntas IS c THEN
  n := 1
IF q9 OF preguntas IS b THEN
  n := n - 1
IF q9 OF preguntas IS a THEN
  n := n - 1
IF q9 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 4
n := 0
IF q10 OF preguntas IS c THEN
  n := 1
IF q10 OF preguntas IS b THEN
  n := n - 1
IF q10 OF preguntas IS a THEN
  n := n - 1
IF q10 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 4
n := 0
IF q11 OF preguntas IS c THEN
  n := 1
IF q11 OF preguntas IS b THEN
  n := n - 1
IF q11 OF preguntas IS a THEN
  n := n - 1
IF q11 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 4
n := 0
IF q12 OF preguntas IS a THEN
  n := 1
IF q12 OF preguntas IS b THEN
  n := n + 1
IF q12 OF preguntas IS c THEN
  n := n - 1
```



```
IF q12 OF preguntas IS d THEN
  n := n - 1
IF q12 OF preguntas IS e THEN
  n := n - 1
IF q12 OF preguntas IS f THEN
  n := n + 1
IF q12 OF preguntas IS g THEN
  n := n + 1
IF q12 OF preguntas IS h THEN
  n := n + 1
IF q12 OF preguntas IS i THEN
  n := n - 1
IF n = 5 THEN
  total := total + 4
IF n = 4 THEN
  total := total + 2
n := 0
IF q13 OF preguntas IS a THEN
  n := 1
IF q13 OF preguntas IS b THEN
  n := n - 1
IF q13 OF preguntas IS c THEN
  n := n - 1
IF q13 OF preguntas IS d THEN
  n := n - 1
IF q13 OF preguntas IS e THEN
  n := n + 1
IF q13 OF preguntas IS f THEN
  n := n + 1
IF q13 OF preguntas IS g THEN
  n := n + 1
IF q13 OF preguntas IS h THEN
  n := n - 1
IF q13 OF preguntas IS i THEN
  n := n + 1
IF q13 OF preguntas IS j THEN
  n := n - 1
IF n = 5 THEN
  total := total + 4
IF n = 4 THEN
  total := total + 2
n := 0
IF q14 OF preguntas IS a THEN
  n := 1
IF q14 OF preguntas IS b THEN
```

```
n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q15 OF preguntas IS b THEN
  n := 1
IF q15 OF preguntas IS a THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q16 OF preguntas IS b THEN
  n := 1
IF q16 OF preguntas IS a THEN
  n := n - 1
IF q16 OF preguntas IS c THEN
  n := n - 1
IF q16 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q17 OF preguntas IS b THEN
  n := 1
IF q17 OF preguntas IS a THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q18 OF preguntas IS d THEN
  n := 1
IF q18 OF preguntas IS a THEN
  n := n - 1
IF q18 OF preguntas IS b THEN
  n := n - 1
IF q18 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 5
n := 0
IF q19 OF preguntas IS a THEN
  n := 1
IF q19 OF preguntas IS b THEN
  n := n - 1
IF q19 OF preguntas IS c THEN
```

```

n := n - 1
IF q19 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 5
n := 0
IF q20 OF preguntas IS a THEN
  n := 1
IF q20 OF preguntas IS b THEN
  n := n - 1
IF q20 OF preguntas IS c THEN
  n := n + 1
IF q20 OF preguntas IS d THEN
  n := n - 1
IF q20 OF preguntas IS e THEN
  n := n + 1
IF n = 3 THEN
  total := total + 5
IF n = 2 THEN
  total := total + 3
n := 0
IF total >= 40 THEN
  BEGIN
    text OF textbox 111 := si
    items[ 6] OF resultados := continua
    items[ 7] OF resultados := salida
  END
ELSE
  text OF textbox 111 := no
ASK resultados
END

```

```

ATTRIBUTE salir SIMPLE
WHEN CHANGED
  BEGIN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
      previo OF dB3 USR 1 := total * 10 / 50
      bandera OF dB3 USR 1 := 0
    WHEN NONE FOUND
      ASK problemas
  FIND END
  exit OF application := TRUE

```

END

```
ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    previo OF dB3 USR 1 := total * 10 / 50
  WHEN NONE FOUND
    ASK problemas
  FIND END
  CHAIN "previo2.knb"
END
```

PREVIO2.KNB

```
ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1 OF preguntas IS a THEN
    n := 1
  IF q1 OF preguntas IS b THEN
    n := n + 1
  IF q1 OF preguntas IS c THEN
    n := n - 1
  IF q1 OF preguntas IS d THEN
    n := n - 1
  IF n = 2 THEN
    total := 2
  IF n = 1 THEN
    total := 1
  n := 0
  IF q2 OF preguntas IS c THEN
    n := 1
  IF q2 OF preguntas IS a THEN
    n := n - 1
  IF q2 OF preguntas IS b THEN
    n := n - 1
  IF q2 OF preguntas IS d THEN
    n := n - 1
  IF n = 1 THEN
    total := total + 1
```

```
n := 0
IF q3 OF preguntas IS b THEN
  n := 1
IF q3 OF preguntas IS a THEN
  n := n - 1
IF q3 OF preguntas IS c THEN
  n := n - 1
IF q3 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q4 OF preguntas IS a THEN
  n := 1
IF q4 OF preguntas IS b THEN
  n := n - 1
IF q4 OF preguntas IS c THEN
  n := n - 1
IF q4 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q5 OF preguntas IS c THEN
  n := 1
IF q5 OF preguntas IS a THEN
  n := n - 1
IF q5 OF preguntas IS b THEN
  n := n - 1
IF q5 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q6 OF preguntas IS c THEN
  n := 1
IF q6 OF preguntas IS a THEN
  n := n - 1
IF q6 OF preguntas IS b THEN
  n := n - 1
IF q6 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
```

```
IF q7 OF preguntas IS c THEN
  n := 1
IF q7 OF preguntas IS a THEN
  n := n - 1
IF q7 OF preguntas IS b THEN
  n := n - 1
IF q7 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q8 OF preguntas IS a THEN
  n := 1
IF q8 OF preguntas IS b THEN
  n := n - 1
IF q8 OF preguntas IS c THEN
  n := n - 1
IF q8 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q9 OF preguntas IS d THEN
  n := 1
IF q9 OF preguntas IS b THEN
  n := n - 1
IF q9 OF preguntas IS a THEN
  n := n - 1
IF q9 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q10 OF preguntas IS b THEN
  n := 1
IF q10 OF preguntas IS c THEN
  n := n - 1
IF q10 OF preguntas IS a THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q11[ 1] OF preguntas = 5 THEN
  n := 1
IF q11[ 2] OF preguntas = 6 THEN
```

```
n := n + 1
IF q11[ 3] OF preguntas = 3 THEN
  n := n + 1
IF q11[ 4] OF preguntas = 2 THEN
  n := n + 1
IF q11[ 5] OF preguntas = 7 THEN
  n := n + 1
IF q11[ 6] OF preguntas = 4 THEN
  n := n + 1
IF (n = 5) OR (n = 4) THEN
  total := total + 2
IF n = 6 THEN
  total := total + 4
n := 0
IF q12 OF preguntas IS d THEN
  n := 1
IF q12 OF preguntas IS a THEN
  n := n - 1
IF q12 OF preguntas IS b THEN
  n := n - 1
IF q12 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q13 OF preguntas IS c THEN
  n := 1
IF q13 OF preguntas IS b THEN
  n := n - 1
IF q13 OF preguntas IS a THEN
  n := n - 1
IF q13 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q14 OF preguntas IS d THEN
  n := 1
IF q14 OF preguntas IS a THEN
  n := n - 1
IF q14 OF preguntas IS b THEN
  n := n - 1
IF q14 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
```

```
total := total + 1
n := 0
IF q15 OF preguntas IS d THEN
  n := 1
IF q15 OF preguntas IS a THEN
  n := n - 1
IF q15 OF preguntas IS b THEN
  n := n - 1
IF q15 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q16 OF preguntas IS a THEN
  n := 1
IF q16 OF preguntas IS b THEN
  n := n - 1
IF q16 OF preguntas IS c THEN
  n := n - 1
IF q16 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q17[ 1] OF preguntas = 1 THEN
  n := 1
IF q17[ 2] OF preguntas = 2 THEN
  n := n + 1
IF q17[ 3] OF preguntas = 3 THEN
  n := n + 1
IF q17[ 4] OF preguntas = 4 THEN
  n := n + 1
IF q17[ 5] OF preguntas = 7 THEN
  n := n + 1
IF q17[ 6] OF preguntas = 5 THEN
  n := n + 1
IF q17[ 7] OF preguntas = 6 THEN
  n := n + 1
IF n = 7 THEN
  total := total + 7
IF n = 6 THEN
  total := total + 3
n := 0
IF q18 OF preguntas IS a AND q18 OF preguntas IS b AND q18 OF preguntas IS c = FALSE
AND q18 OF preguntas IS d AND q18 OF preguntas IS e AND q18 OF preguntas IS f AND q18
```


OF preguntas IS g AND q18 OF preguntas IS h AND q18 OF preguntas IS i AND q18 OF preguntas IS j AND q18 OF preguntas IS k AND q18 OF preguntas IS l AND q18 OF preguntas IS m THEN
total := total + 3
IF q18b[2] OF preguntas = 2 THEN
total := total + 2
IF q18b[4] OF preguntas = 1 THEN
total := total + 2
IF q18b[5] OF preguntas = 1 THEN
total := total + 2
IF q18b[10] OF preguntas = 4 THEN
total := total + 2
IF q18b[11] OF preguntas = 3 THEN
total := total + 2
IF q19[1] OF preguntas = 9 THEN
n := 1
IF q19[2] OF preguntas = 5 THEN
n := n + 1
IF q19[3] OF preguntas = 610 THEN
n := n + 1
IF q19[4] OF preguntas = 2000 THEN
n := n + 1
IF q19[5] OF preguntas = 305 THEN
n := n + 1
IF q19[6] OF preguntas = 1000 THEN
n := n + 1
IF q19[7] OF preguntas = 1.8 THEN
n := n + 1
IF q19[8] OF preguntas = 1 THEN
n := n + 1
IF q19[9] OF preguntas = 5.5 THEN
n := n + 1
IF q19[10] OF preguntas = 3 THEN
n := n + 1
IF q19[11] OF preguntas = 610 THEN
n := n + 1
IF q19[12] OF preguntas = 2000 THEN
n := n + 1
IF q19[13] OF preguntas = 305 THEN
n := n + 1
IF q19[14] OF preguntas = 1000 THEN
n := n + 1
IF q19[15] OF preguntas = 1.8 THEN
n := n + 1
IF q19[16] OF preguntas = 1 THEN

```
n := n + 1
IF q19[ 17] OF preguntas = 305 THEN
  n := n + 1
IF q19[ 18] OF preguntas = 1000 THEN
  n := n + 1
IF q19[ 19] OF preguntas = 1.8 THEN
  n := n + 1
IF q19[ 20] OF preguntas = 1 THEN
  n := n + 1
IF n = 20 THEN
  total := total + 10
IF n = 19 THEN
  total := total + 5
n := 0
IF q20[ 1] OF preguntas = 1 THEN
  BEGIN
    IF q20[ 2] OF preguntas = 3 THEN
      BEGIN
        IF q20[ 3] OF preguntas = 9 THEN
          total := total + 3
        END
      ELSE
        IF q20[ 2] OF preguntas = 9 THEN
          BEGIN
            IF q20[ 3] OF preguntas = 3 THEN
              total := total + 3
            END
          END
        END
      ELSE
        IF q20[ 1] OF preguntas = 3 THEN
          BEGIN
            IF q20[ 2] OF preguntas = 1 THEN
              BEGIN
                IF q20[ 3] OF preguntas = 9 THEN
                  total := total + 3
                END
              ELSE
                IF q20[ 2] OF preguntas = 9 THEN
                  BEGIN
                    IF q20[ 3] OF preguntas = 1 THEN
                      total := total + 3
                    END
                  END
                END
              ELSE
                IF q20[ 1] OF preguntas = 9 THEN
```

```
BEGIN
  IF q20[ 2] OF preguntas = 3 THEN
    BEGIN
      IF q20[ 3] OF preguntas = 1 THEN
        total := total + 3
      END
    ELSE
      IF q20[ 2] OF preguntas = 1 THEN
        BEGIN
          IF q20[ 3] OF preguntas = 3 THEN
            total := total + 3
          END
        END
      END
    END
  IF q20[ 4] OF preguntas = 2 THEN
    BEGIN
      IF q20[ 5] OF preguntas = 4 THEN
        BEGIN
          IF q20[ 6] OF preguntas = 6 THEN
            total := total + 3
          END
        ELSE
          IF q20[ 5] OF preguntas = 6 THEN
            BEGIN
              IF q20[ 6] OF preguntas = 4 THEN
                total := total + 3
              END
            END
          END
        END
      END
    ELSE
      IF q20[ 4] OF preguntas = 4 THEN
        BEGIN
          IF q20[ 5] OF preguntas = 2 THEN
            BEGIN
              IF q20[ 6] OF preguntas = 6 THEN
                total := total + 3
              END
            ELSE
              IF q20[ 5] OF preguntas = 6 THEN
                BEGIN
                  IF q20[ 6] OF preguntas = 2 THEN
                    total := total + 3
                  END
                END
              END
            END
          END
        END
      END
    ELSE
      IF q20[ 4] OF preguntas = 6 THEN
        BEGIN
```

```

IF q20[ 5] OF preguntas = 4 THEN
  BEGIN
    IF q20[ 6] OF preguntas = 2 THEN
      total := total + 3
    END
  ELSE
    IF q20[ 5] OF preguntas = 2 THEN
      BEGIN
        IF q20[ 6] OF preguntas = 4 THEN
          total := total + 3
        END
      END
    END
  END
  IF q21[ 1] OF preguntas = FALSE THEN
    total := total + 1
  IF q21[ 2] OF preguntas = TRUE THEN
    total := total + 1
  IF q21[ 3] OF preguntas = FALSE THEN
    total := total + 1
  IF q21[ 4] OF preguntas = FALSE THEN
    total := total + 1
  IF total >= 48 THEN
    BEGIN
      text OF textbox 111 := si
      items[ 6] OF resultados := salida
      items[ 7] OF resultados := continua
    END
  ELSE
    text OF textbox 111 := no
  ASK resultados
END

```

```

ATTRIBUTE salir SIMPLE
WHEN CHANGED
  BEGIN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
      previo2 OF dB3 USR 1 := total * 10 / 60
      bandera OF dB3 USR 1 := 0
    WHEN NONE FOUND
      ASK problemas
    FIND END
    exit OF application := TRUE
  END

```

```
ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    previo2 OF dB3 USR 1 := total * 10 / 60
  WHEN NONE FOUND
    ASK problemas
  FIND END
  CHAIN "menu.knb"
END
```

FINAL.KNB

```
ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1 OF preguntas IS d THEN
    n := 1
  IF q1 OF preguntas IS b THEN
    n := n - 1
  IF q1 OF preguntas IS c THEN
    n := n - 1
  IF q1 OF preguntas IS a THEN
    n := n - 1
  IF n = 1 THEN
    total := 1
  n := 0
  IF q2 OF preguntas IS a THEN
    n := 1
  IF q2 OF preguntas IS c THEN
    n := n - 1
  IF q2 OF preguntas IS b THEN
    n := n - 1
  IF q2 OF preguntas IS d THEN
    n := n - 1
  IF n = 1 THEN
    total := total + 1
  n := 0
  IF q3 OF preguntas IS d THEN
    n := 1
```

```
IF q3 OF preguntas IS a THEN
  n := n - 1
IF q3 OF preguntas IS c THEN
  n := n - 1
IF q3 OF preguntas IS b THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q4 OF preguntas IS b THEN
  n := 1
IF q4 OF preguntas IS a THEN
  n := n - 1
IF q4 OF preguntas IS c THEN
  n := n - 1
IF q4 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q5 OF preguntas IS a THEN
  n := 1
IF q5 OF preguntas IS c THEN
  n := n - 1
IF q5 OF preguntas IS b THEN
  n := n - 1
IF q5 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q6 OF preguntas IS b THEN
  n := 1
IF q6 OF preguntas IS a THEN
  n := n - 1
IF q6 OF preguntas IS c THEN
  n := n - 1
IF q6 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q7 OF preguntas IS a THEN
  n := 1
IF q7 OF preguntas IS c THEN
```

```
n := n - 1
IF q7 OF preguntas IS b THEN
  n := n - 1
IF q7 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q8 OF preguntas IS d THEN
  n := 1
IF q8 OF preguntas IS b THEN
  n := n - 1
IF q8 OF preguntas IS c THEN
  n := n - 1
IF q8 OF preguntas IS a THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q9 OF preguntas IS d THEN
  n := 1
IF q9 OF preguntas IS b THEN
  n := n - 1
IF q9 OF preguntas IS a THEN
  n := n - 1
IF q9 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q10 OF preguntas IS b THEN
  n := 1
IF q10 OF preguntas IS c THEN
  n := n - 1
IF q10 OF preguntas IS a THEN
  n := n - 1
IF q10 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q11 OF preguntas IS d THEN
  n := 1
IF q11 OF preguntas IS a THEN
  n := n - 1
```

```
IF q11 OF preguntas IS b THEN
  n := n - 1
IF q11 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q12 OF preguntas IS d THEN
  n := 1
IF q12 OF preguntas IS a THEN
  n := n - 1
IF q12 OF preguntas IS b THEN
  n := n - 1
IF q12 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q13 OF preguntas IS c THEN
  n := 1
IF q13 OF preguntas IS b THEN
  n := n + 1
IF q13 OF preguntas IS a THEN
  n := n + 1
IF q13 OF preguntas IS d THEN
  n := n - 1
IF n = 2 THEN
  total := total + 0.5
IF n = 3 THEN
  total := total + 1
n := 0
IF q14 OF preguntas IS a THEN
  n := 1
IF q14 OF preguntas IS b THEN
  n := n + 1
IF q14 OF preguntas IS d THEN
  n := n - 1
IF q14 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 0.5
IF n = 2 THEN
  total := total + 1
n := 0
IF q15 OF preguntas IS d THEN
```



```
n := 1
IF q15 OF preguntas IS a THEN
  n := n - 1
IF q15 OF preguntas IS b THEN
  n := n - 1
IF q15 OF preguntas IS c THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q16 OF preguntas IS a THEN
  n := 1
IF q16 OF preguntas IS b THEN
  n := n - 1
IF q16 OF preguntas IS c THEN
  n := n - 1
IF q16 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q17 OF preguntas IS b THEN
  n := 1
IF q17 OF preguntas IS a THEN
  n := n - 1
IF q17 OF preguntas IS c THEN
  n := n - 1
IF q17 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q18 OF preguntas IS d THEN
  n := 1
IF q18 OF preguntas IS a THEN
  n := n - 1
IF q18 OF preguntas IS c THEN
  n := n - 1
IF q18 OF preguntas IS b THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q19 OF preguntas IS c THEN
  n := 1
```

```
IF q19 OF preguntas IS a THEN
  n := n - 1
IF q19 OF preguntas IS b THEN
  n := n - 1
IF q19 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q20[ 1] OF preguntas = 3 THEN
  n := 1
IF q20[ 2] OF preguntas = 3 THEN
  n := n + 1
IF n = 1 THEN
  total := total + 0.5
IF n = 2 THEN
  total := total + 1
n := 0
IF q21[ 1] OF preguntas = 1 THEN
  n := 1
IF q21[ 2] OF preguntas = 2 THEN
  n := n + 1
IF n = 1 THEN
  total := total + 0.5
IF n = 2 THEN
  total := total + 1
n := 0
IF q22 OF preguntas = TRUE THEN
  total := total + 1
IF q23 OF preguntas IS b THEN
  n := 1
IF q23 OF preguntas IS a THEN
  n := n - 1
IF q23 OF preguntas IS c THEN
  n := n - 1
IF q23 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q24 OF preguntas IS a THEN
  n := 1
IF q24 OF preguntas IS c THEN
  n := n - 1
IF q24 OF preguntas IS b THEN
```

```
n := n - 1
IF q24 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q25 OF preguntas IS c THEN
  n := 1
IF q25 OF preguntas IS a THEN
  n := n - 1
IF q25 OF preguntas IS b THEN
  n := n - 1
IF q25 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q26 OF preguntas IS c THEN
  n := 1
IF q26 OF preguntas IS b THEN
  n := n - 1
IF q26 OF preguntas IS a THEN
  n := n - 1
IF q26 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q27 OF preguntas IS c THEN
  n := 1
IF q27 OF preguntas IS d THEN
  n := n - 1
IF q27 OF preguntas IS b THEN
  n := n - 1
IF q27 OF preguntas IS a THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q28 OF preguntas IS a THEN
  n := 1
IF q28 OF preguntas IS c THEN
  n := n - 1
IF q28 OF preguntas IS b THEN
  n := n - 1
```

```
IF q28 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q29 OF preguntas IS b THEN
  n := 1
IF q29 OF preguntas IS c THEN
  n := n - 1
IF q29 OF preguntas IS a THEN
  n := n - 1
IF q29 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q30 OF preguntas IS b THEN
  n := 1
IF q30 OF preguntas IS a THEN
  n := n - 1
IF q30 OF preguntas IS c THEN
  n := n - 1
IF q30 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q31 OF preguntas IS a THEN
  n := 1
IF q31 OF preguntas IS b THEN
  n := n - 1
IF q31 OF preguntas IS c THEN
  n := n - 1
IF q31 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF (UPCASE(q32[ 1] OF preguntas) = "HASTA NUEVO AVISO") AND (UPCASE(q32[ 2] OF
preguntas) = "UNTIL FURTHER ADVISE") THEN
  n := n + 1
IF (UPCASE(q32[ 3] OF preguntas) = "AL LIMITE DEL AREA") AND (UPCASE(q32[ 4] OF
preguntas) = "TO THE BOUNDARY") THEN
  n := n + 1
```

```
IF (UPCASE(q32[ 5] OF preguntas) = "ESPERE MAYOR ALTITUD") AND (UPCASE(q32[ 6]
OF preguntas) = "EXPECT HIGH ALTITUDE") THEN
  n := n + 1
IF (UPCASE(q32[ 7] OF preguntas) = "DEL LIMITE DEL AREA") AND (UPCASE(q32[ 8] OF
preguntas) = "FROM THE BOUNDARY") THEN
  n := n + 1
IF (UPCASE(q32[ 9] OF preguntas) = "DESCIENDA Y MANTENGA") AND (UPCASE(q32[ 10]
OF preguntas) = "DESCEND AND MAINTAIN") THEN
  n := n + 1
IF (UPCASE(q32[ 11] OF preguntas) = "NO SE PREVEE DEMORA") AND (UPCASE(q32[ 12]
OF preguntas) = "NO DELAY EXPECTED") THEN
  n := n + 1
IF (UPCASE(q32[ 13] OF preguntas) = "VIRE POR SU IZQUIERDA") AND (UPCASE(q32[ 14]
OF preguntas) = "TURN LEFT") THEN
  n := n + 1
IF (UPCASE(q32[ 15] OF preguntas) = "VIRE POR SU DERECHA") AND (UPCASE(q32[ 16] OF
preguntas) = "TURN RIGHT") THEN
  n := n + 1
IF (UPCASE(q32[ 17] OF preguntas) = "SOBRE CAPA") AND (UPCASE(q32[ 18] OF
preguntas) = "ON TOP") THEN
  n := n + 1
IF (UPCASE(q32[ 19] OF preguntas) = "REGRESE") AND (UPCASE(q32[ 20] OF preguntas)
= "RETURN") THEN
  n := n + 1
IF (UPCASE(q32[ 21] OF preguntas) = "AREA TERMINAL") AND (UPCASE(q32[ 22] OF
preguntas) = "TERMINAL AREA") THEN
  n := n + 1
IF (UPCASE(q32[ 23] OF preguntas) = "MANTENGA") AND (UPCASE(q32[ 24] OF
preguntas) = "MAINTAIN") THEN
  n := n + 1
IF (UPCASE(q32[ 25] OF preguntas) = "ASCIENDA") AND (UPCASE(q32[ 26] OF preguntas)
= "CLIMB") THEN
  n := n + 1
IF n = 13 THEN
  total := total + 10
IF n > 9 AND n < 13 THEN
  total := total + 5
n := 0
IF q33 OF preguntas IS b THEN
  n := 1
IF q33 OF preguntas IS a THEN
  n := n - 1
IF q33 OF preguntas IS c THEN
  n := n - 1
IF q33 OF preguntas IS d THEN
```

```
n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q34 OF preguntas IS b THEN
  n := 1
IF q34 OF preguntas IS a THEN
  n := n - 1
IF q34 OF preguntas IS c THEN
  n := n - 1
IF q34 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q35 OF preguntas IS c THEN
  n := 1
IF q35 OF preguntas IS a THEN
  n := n - 1
IF q35 OF preguntas IS b THEN
  n := n - 1
IF q35 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q36 OF preguntas IS b THEN
  n := 1
IF q36 OF preguntas IS a THEN
  n := n - 1
IF q36 OF preguntas IS c THEN
  n := n - 1
IF q36 OF preguntas IS d THEN
  n := n - 1
IF n = 1 THEN
  total := total + 1
n := 0
IF q37 OF preguntas IS d THEN
  n := 1
IF q37 OF preguntas IS a THEN
  n := n - 1
IF q37 OF preguntas IS c THEN
  n := n - 1
IF q37 OF preguntas IS b THEN
  n := n - 1
```

```
IF n = 1 THEN
  total := total + 1
  n := 0
  IF q38 OF preguntas IS c THEN
    n := 1
  IF q38 OF preguntas IS a THEN
    n := n - 1
  IF q38 OF preguntas IS b THEN
    n := n - 1
  IF q38 OF preguntas IS d THEN
    n := n - 1
  IF n = 1 THEN
    total := total + 1
    n := 0
    IF q39 OF preguntas IS d THEN
      n := 1
    IF q39 OF preguntas IS a THEN
      n := n - 1
    IF q39 OF preguntas IS c THEN
      n := n - 1
    IF q39 OF preguntas IS b THEN
      n := n - 1
    IF n = 1 THEN
      total := total + 1
      n := 0
      IF q40 OF preguntas IS a THEN
        n := 1
      IF q40 OF preguntas IS b THEN
        n := n - 1
      IF q40 OF preguntas IS c THEN
        n := n - 1
      IF q40 OF preguntas IS d THEN
        n := n - 1
    IF n = 1 THEN
      total := total + 1
      n := 0
      IF q41 OF preguntas IS a THEN
        n := 1
      IF q41 OF preguntas IS b THEN
        n := n - 1
      IF q41 OF preguntas IS c THEN
        n := n - 1
      IF q41 OF preguntas IS d THEN
        n := n - 1
    IF n = 1 THEN
```

```

    total := total + 1
    n := 0
    IF q42 OF preguntas IS b THEN
        n := 1
    IF q42 OF preguntas IS a THEN
        n := n - 1
    IF q42 OF preguntas IS c THEN
        n := n - 1
    IF q42 OF preguntas IS d THEN
        n := n - 1
    IF n = 1 THEN
        total := total + 1
    n := 0
    IF total >= 40.8 THEN
        text OF textbox 111 := si
    ELSE
        text OF textbox 111 := no
    ASK resultados
END

```

```

ATTRIBUTE salir SIMPLE
WHEN CHANGED
BEGIN
    beep OF beeper := TRUE
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
        final OF dB3 USR 1 := total * 10 / 51
        bandera OF dB3 USR 1 := 0
    WHEN NONE FOUND
        ASK problemas
    FIND END
    exit OF application := TRUE
END

```

```

ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
    beep OF beeper := TRUE
    CHAIN "menu.knb"
END

```

EX1.KNB


```

ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1 OF preguntas = FALSE AND q2 OF preguntas = FALSE AND q3 OF preguntas = FALSE
  THEN
    total := 1
  ELSE
    BEGIN
      FIND dB3 USR 1
      LIMIT 1
      WHERE bandera OF dB3 USR 1 <> 0
      WHEN FOUND
        ex1 OF dB3 USR 1 := 0
      WHEN NONE FOUND
        ASK problemas
      FIND END
      CHAIN "plst.knb"
    END
  END
END

```

```

ATTRIBUTE salir SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex1 OF dB3 USR 1 := 10
    bandera OF dB3 USR 1 := 0
  WHEN NONE FOUND
    ASK problemas
  FIND END
  exit OF application := TRUE
END

```

```

ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex1 OF dB3 USR 1 := 10
  WHEN NONE FOUND

```

```

    ASK problemas
  FIND END
  CHAIN "pls2.knb"
END

```

```

DEMON si pasa
IF total = 1
THEN text OF texto := si
AND items[ 3] OF fin := sale
AND items[ 4] OF fin := continuara

```

EX2.KNB

```

ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1[1] OF preguntas = 1 AND q1[2] OF preguntas = 3 AND q1[3] OF preguntas = 5 AND
    q1[4] OF preguntas = 6 AND q1[5] OF preguntas = 8 AND q1[6] OF preguntas = 7 AND
    q1[7] OF preguntas = 4 AND q1[8] OF preguntas = 2 AND q2 OF preguntas = FALSE AND
    q3 OF preguntas IS d AND q4 OF preguntas = FALSE AND q5 OF preguntas IS a AND q5
    OF preguntas IS b AND q5 OF preguntas IS c AND q5 OF preguntas IS d AND q5 OF
    preguntas IS e AND q5 OF preguntas IS f AND q5 OF preguntas IS g AND q5 OF preguntas
    IS h AND q5 OF preguntas IS i AND q5 OF preguntas IS j AND q5 OF preguntas IS k THEN
    total := 1
  ELSE
    BEGIN
      FIND dB3 USR 1
      LIMIT 1
      WHERE bandera OF dB3 USR 1 <> 0
      WHEN FOUND
        ex2 OF dB3 USR 1 := 0
      WHEN NONE FOUND
        ASK problemas
      FIND END
      CHAIN "pls2.knb"
    END
  END
END

```

```

ATTRIBUTE salir SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0

```

```

WHEN FOUND
  ex2 OF dB3 USR 1 := 10
  bandera OF dB3 USR 1 := 0
WHEN NONE FOUND
  ASK problemas
FIND END
exit OF application := TRUE
END

```

```

ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex2 OF dB3 USR 1 := 10
  WHEN NONE FOUND
    ASK problemas
  FIND END
  CHAIN "p1e3.knb"
END

```

```

DEMON si pasa
IF total = 1
THEN text OF texto := si
AND items[ 3] OF fin := sale
AND items[ 4] OF fin := continuara

```

EX3.PRL

```

ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1 OF preguntas = FALSE AND q2 OF preguntas IS d AND q3[ 1] OF preguntas =
  FALSE AND q3[ 2] OF preguntas AND q3[ 3] OF preguntas AND q3[ 4] OF preguntas
  THEN
    total := 1
  ELSE
    BEGIN
      FIND dB3 USR 1
      LIMIT 1
      WHERE bandera OF dB3 USR 1 <> 0
      WHEN FOUND

```

```
    ex3 OF dB3 USR 1 := 0
    bandera OF dB3 USR 1 := 0
    WHEN NONE FOUND
        ASK problemas
    FIND END
    CHAIN "p1s3.knb"
END
END
```

```
ATTRIBUTE sallr SIMPLE
WHEN CHANGED
BEGIN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
        ex3 OF dB3 USR 1 := 10
        bandera OF dB3 USR 1 := 0
    WHEN NONE FOUND
        ASK problemas
    FIND END
    exit OF application := TRUE
END
```

```
ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
        ex3 OF dB3 USR 1 := 10
    WHEN NONE FOUND
        ASK problemas
    FIND END
    CHAIN "p1s4.knb"
END
```

```
DEMON si pasa
IF total = 1
THEN text OF texto := si
AND items[ 3] OF fin := sale
AND items[ 4] OF fin := continuara
```

EX4.KNBATTRIBUTE *evalua* SIMPLE

WHEN CHANGED

BEGIN

```

    IF q1a OF preguntas IS pares AND q1b OF preguntas IS pares AND q1c OF preguntas IS
    nones AND q1d OF preguntas IS nones AND q2 OF preguntas AND q3[ 1] OF preguntas AND
    q3[ 2] OF preguntas = FALSE AND q4 OF preguntas IS a = FALSE AND q4 OF preguntas IS
    b = FALSE AND q4 OF preguntas IS c = FALSE AND q4 OF preguntas IS d THEN

```

```

    total := 1

```

ELSE

BEGIN

```

    FIND dB3 USR 1

```

```

    LIMIT 1

```

```

    WHERE bandera OF dB3 USR 1 <> 0

```

```

    WHEN FOUND

```

```

        ex4 OF dB3 USR 1 := 0

```

```

    WHEN NONE FOUND

```

```

        ASK problemas

```

```

    FIND END

```

```

    CHAIN "pl4.knb"

```

END

END

ATTRIBUTE *salir* SIMPLE

WHEN CHANGED

BEGIN

```

    FIND dB3 USR 1

```

```

    LIMIT 1

```

```

    WHERE bandera OF dB3 USR 1 <> 0

```

```

    WHEN FOUND

```

```

        ex4 OF dB3 USR 1 := 10

```

```

        bandera OF dB3 USR 1 := 0

```

```

    WHEN NONE FOUND

```

```

        ASK problemas

```

```

    FIND END

```

```

    exit OF application := TRUE

```

END

ATTRIBUTE *continuar* SIMPLE

WHEN CHANGED

BEGIN

```

    FIND dB3 USR 1

```

```

    LIMIT 1

```

```

    WHERE bandera OF dB3 USR 1 <> 0

```

```

WHEN FOUND
  ex4 OF dB3 USR 1 := 10
WHEN NONE FOUND
  ASK problemas
FIND END
CHAIN "p1e5.knb"
END

```

```

DEMON si pasa
IF total = 1
THEN text OF texto := si
AND items[ 3] OF fin := sale
AND items[ 4] OF fin := continuara

```

EX5.KNB

```

ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1 OF preguntas AND q2 OF preguntas = FALSE AND q3 OF preguntas = FALSE THEN
    total := 1
  ELSE
    BEGIN
      FIND dB3 USR 1
      LIMIT 1
      WHERE bandera OF dB3 USR 1 <> 0
      WHEN FOUND
        ex5 OF dB3 USR 1 := 0
      WHEN NONE FOUND
        ASK problemas
      FIND END
      CHAIN "p1e5.knb"
    END
  END
END

```

```

ATTRIBUTE salir SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex5 OF dB3 USR 1 := 10
    bandera OF dB3 USR 1 := 0
  END
END

```

```
WHEN NONE FOUND
  ASK problemas
FIND END
exit OF application := TRUE
END
```

```
ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex5 OF dB3 USR 1 := 10
  WHEN NONE FOUND
    ASK problemas
  FIND END
  CHAIN "pl6.knb"
END
```

```
DEMON si pasa
IF total = 1
THEN text OF texto := si
AND items[ 3 ] OF fin := sale
AND items[ 4 ] OF fin := continuara
```

EX6.KNB

```
ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1 OF preguntas IS a AND q1 OF preguntas IS b AND q1 OF preguntas IS c THEN
    total := 1
  ELSE
    BEGIN
      FIND dB3 USR 1
      LIMIT 1
      WHERE bandera OF dB3 USR 1 <> 0
      WHEN FOUND
        ex6 OF dB3 USR 1 := 0
      WHEN NONE FOUND
        ASK problemas
      FIND END
      CHAIN "pl6.knb"
```

```
END
END
```

```
ATTRIBUTE salir SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex6 OF dB3 USR 1 := 10
    bandera OF dB3 USR 1 := 0
  WHEN NONE FOUND
    ASK problemas
  FIND END
  exit OF application := TRUE
END
```

```
ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex6 OF dB3 USR 1 := 10
  WHEN NONE FOUND
    ASK problemas
  FIND END
  CHAIN "p1s7.knb"
END
```

```
DEMON si pasa
IF total = 1
THEN text OF texto := si
AND items[ 3] OF fin := sale
AND items[ 4] OF fin := continuara
```

EX7.KNB

```
ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1 OF preguntas IS a AND q1 OF preguntas IS b AND q1 OF preguntas IS c THEN
```



```
total := total + 1
IF q2 OF preguntas IS c THEN
total := total + 1
IF q3 OF preguntas IS b THEN
total := total + 1
IF q4 OF preguntas IS a THEN
total := total + 1
IF total < 4 THEN
BEGIN
FIND dB3 USR 1
LIMIT 1
WHERE bandera OF dB3 USR 1 <> 0
WHEN FOUND
ex7 OF dB3 USR 1 := 0
WHEN NONE FOUND
ASK problemas
FIND END
CHAIN "pl67.knb"
END
END
```

```
ATTRIBUTE salir SIMPLE
WHEN CHANGED
BEGIN
FIND dB3 USR 1
LIMIT 1
WHERE bandera OF dB3 USR 1 <> 0
WHEN FOUND
ex7 OF dB3 USR 1 := 10
bandera OF dB3 USR 1 := 0
WHEN NONE FOUND
ASK problemas
FIND END
exit OF application := TRUE
END
```

```
ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
FIND dB3 USR 1
LIMIT 1
WHERE bandera OF dB3 USR 1 <> 0
WHEN FOUND
ex7 OF dB3 USR 1 := 10
WHEN NONE FOUND
```

```
    ASK problemas
  FIND END
  CHAIN "plsb.knb"
END
```

```
DEMON si pasa
IF total = 4
THEN text OF texto := si
AND items[ 3] OF fin := sale
AND items[ 4] OF fin := continuara
```

EXB.KNB

```
ATTRIBUTE evalua SIMPLE
WHEN CHANGED
BEGIN
  IF q1 OF preguntas IS a AND q1 OF preguntas IS b AND q1 OF preguntas IS c AND q1 OF
  preguntas IS d = FALSE THEN
    total := 1
  ELSE
    BEGIN
      FIND dB3 USR 1
      LIMIT 1
      WHERE bandera OF dB3 USR 1 <> 0
      WHEN FOUND
        exB OF dB3 USR 1 := 0
      WHEN NONE FOUND
        ASK problemas
      FIND END
      CHAIN "plsb.knb"
    END
  END
END
```

```
ATTRIBUTE salir SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    exB OF dB3 USR 1 := 10
    bandera OF dB3 USR 1 := 0
  WHEN NONE FOUND
    ASK problemas
```

```
FIND END
  exit OF application := TRUE
END
```

```
ATTRIBUTE continuar SIMPLE
WHEN CHANGED
  BEGIN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
      ex8 OF dB3 USR 1 := 10
    WHEN NONE FOUND
      ASK problemas
    FIND END
    CHAIN "pl59.knb"
  END
```

```
DEMON si pasa
IF total = 1
THEN text OF texto := si
AND items[ 3] OF fin := sale
AND items[ 4] OF fin := continuara
```

EX9.KNB

```
ATTRIBUTE evalua SIMPLE
WHEN CHANGED
  BEGIN
    IF q1 OF preguntas IS c THEN
      total := total + 1
    IF q2 OF preguntas IS b THEN
      total := total + 1
    IF q3 OF preguntas IS b THEN
      total := total + 1
    IF q4 OF preguntas IS a THEN
      total := total + 1
    IF total < 4 THEN
      BEGIN
        FIND dB3 USR 1
        LIMIT 1
        WHERE bandera OF dB3 USR 1 <> 0
        WHEN FOUND
          ex9 OF dB3 USR 1 := 0
        END
```

```
    WHEN NONE FOUND
      ASK problemas
    FIND END
    CHAIN "plsg.knb"
  END
END
```

```
ATTRIBUTE salir SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex9 OF dB3 USR 1 := 10
    bandera OF dB3 USR 1 := 0
  WHEN NONE FOUND
    ASK problemas
  FIND END
  exit OF application := TRUE
END
```

```
ATTRIBUTE continuar SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    ex9 OF dB3 USR 1 := 10
  WHEN NONE FOUND
    ASK problemas
  FIND END
  CHAIN "final.knb"
END
```

```
DEMON si pasa
IF total = 4
THEN text OF texto := si
AND items[ 3] OF fin := sale
AND items[ 4] OF fin := continuara
```

PIS1.KNB

```
CLASS muestras
WITH muestra dibujo1 SIMPLE
  WHEN CHANGED
  BEGIN
    paso dibujo1 := TRUE
    visible OF dibujo window := TRUE
    output OF dibujo window := dib1
  END

CLASS presenta
WITH adepis1 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D3
  END
WITH sigueD3 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D4
  END
WITH sigueD4 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso pizarra4a AND paso pizarra4b AND paso dibujo1 THEN
      ASK D5
    ELSE
      ASK mensaje trampa
    END
  END
WITH sigueD5 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D6
  END
WITH sigueD6 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D7
  END
WITH sigueD7 SIMPLE
  WHEN CHANGED
  BEGIN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
```

```
    decide := TRUE
    WHEN NONE FOUND
      ASK problemas
      CHAIN "datos.knb"
    FIND END
  END
  WITH piz4a SIMPLE
    WHEN CHANGED
      BEGIN
        paso pizarra4a := TRUE
        ASK P4a
      END
  WITH piz4b SIMPLE
    WHEN CHANGED
      BEGIN
        paso pizarra4b := TRUE
        ASK P4b
      END
  WITH adP4b SIMPLE
    WHEN CHANGED
      BEGIN
        ASK P4bi
      END
  WITH atP4a SIMPLE
    WHEN CHANGED
      BEGIN
        ASK D4
      END
  WITH atrasD3 SIMPLE
    WHEN CHANGED
      BEGIN
        ASK PresentaP1S1
      END
  WITH adP4bi SIMPLE
    WHEN CHANGED
      BEGIN
        IF paso dibujo1 THEN
          ASK P4bii
        ELSE
          ASK mensaje trampa
        END
      END
  ATTRIBUTE cambia SIMPLE
    WHEN CHANGED
      BEGIN
```

```
IF bandera = FALSE THEN
  BEGIN
    output OF dibujo window := dib1
    bandera := TRUE
  END
ELSE
  BEGIN
    output OF dibujo window := dib2
    bandera := FALSE
  END
END
```

```
ATTRIBUTE cierra dibujo window SIMPLE
WHEN CHANGED
  BEGIN
    visible OF dibujo window := FALSE
  END
```

```
ATTRIBUTE salida SIMPLE
WHEN CHANGED
  BEGIN
    CHAIN "menu.knb"
  END
```

```
ATTRIBUTE decide SIMPLE
WHEN CHANGED
  BEGIN
    IF ex1 OF dB3 USR 1 = 10 THEN
      CHAIN "menu.knb"
    ELSE
      CHAIN "ex1.knb"
    END
```

P192.KNB

```
CLASS presenta
WITH sigueD9 SIMPLE
WHEN CHANGED
  BEGIN
    IF paso pizarra9 THEN
      ASK D10
    ELSE
      ASK mensaje trampa
    END
```

```
WITH sigueD10 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso pizarra10 THEN
      ASK D11
    ELSE
      ASK mensaje trampa
    END
  END
WITH sigueD11 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D12
  END
WITH sigueD12 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D13
  END
WITH sigueD13 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D14
  END
WITH adep162 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D9
  END
WITH piz9 SIMPLE
  WHEN CHANGED
  BEGIN
    paso pizarra9 := TRUE
    ASK P9
  END
WITH adP9 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D9
  END
WITH piz10 SIMPLE
  WHEN CHANGED
  BEGIN
    paso pizarra10 := TRUE
    ASK P10
  END
```



```
WITH adP10 SIMPLE
  WHEN CHANGED
    BEGIN
      ASK D10
    END
WITH sigueD14 SIMPLE
  WHEN CHANGED
    BEGIN
      FIND dB3 USR 1
      LIMIT 1
      WHERE bandera OF dB3 USR 1 <> 0
      WHEN FOUND
        decide := TRUE
      WHEN NONE FOUND
        ASK problemas
        CHAIN "datos.knb"
      FIND END
    END
WITH haclaP1S2 SIMPLE
  WHEN CHANGED
    BEGIN
      ASK PresentaP1S2
    END
```

```
ATTRIBUTE salida SIMPLE
  WHEN CHANGED
    BEGIN
      CHAIN "menu.knb"
    END
```

```
ATTRIBUTE decide SIMPLE
  WHEN CHANGED
    BEGIN
      IF ex2 OF dB3 USR 1 = 10 THEN
        CHAIN "menu.knb"
      ELSE
        CHAIN "ex2.knb"
      END
```

P1S3.KNB

```
CLASS presenta
  WITH adeple3 SIMPLE
  WHEN CHANGED
```

```
BEGIN
  ASK D16
END
WITH sigueD16 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso pizarra16 THEN
      ASK D17
    ELSE
      ASK mensaje trampa
    END
  END
WITH sigueD17 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D18
  END
WITH sigueD18 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D19
  END
WITH adP16 SIMPLE
  WHEN CHANGED
  BEGIN
    paso pizarra16 := TRUE
    ASK P16b
  END
WITH adP16b SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D16
  END
WITH piz16 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK P16
  END
WITH sigueD19 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso pizarra19 THEN
      FIND dB3 USR 1
      LIMIT 1
      WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
```

```
    decide := TRUE
    WHEN NONE FOUND
      ASK problemas
      CHAIN "datos.knb"
    FIND END
  ELSE
    ASK mensaje trampa
  END
WITH piz19 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK P19
  END
WITH adP19 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK P19a
  END
WITH adP19b SIMPLE
  WHEN CHANGED
  BEGIN
    ASK P19c
  END
WITH adP19c SIMPLE
  WHEN CHANGED
  BEGIN
    paso pizarra19 := TRUE
    ASK P19d
  END
WITH adP19a SIMPLE
  WHEN CHANGED
  BEGIN
    ASK P19b
  END
WITH adP19d SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D19
  END
WITH haciaP153 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK PresentaP153
  END
```

```
ATTRIBUTE salida SIMPLE
WHEN CHANGED
BEGIN
  CHAIN "menu.knb"
END
```

```
ATTRIBUTE decide SIMPLE
WHEN CHANGED
BEGIN
  IF ex3 OF dB3 USR 1 = 10 THEN
    CHAIN "menu.knb"
  ELSE
    CHAIN "ex3.knb"
  END
```

P1S4.KNB

```
CLASS muestras
WITH muestra tabla21 SIMPLE
WHEN CHANGED
BEGIN
  paso tabla21 := TRUE
  visible OF tabla window := TRUE
  output OF tabla window := tab1
END
WITH muestra tabla23 SIMPLE
WHEN CHANGED
BEGIN
  paso tabla23 := TRUE
  visible OF tabla window := TRUE
  output OF tabla window := tab2
END
WITH muestra dibujo1 SIMPLE
WHEN CHANGED
BEGIN
  paso dibujo1 := TRUE
  visible OF dibujo window := TRUE
  output OF dibujo window := dib1
END
```

```
CLASS presenta
WITH adeple4 SIMPLE
WHEN CHANGED
BEGIN
```

```
    ASK D21
  END
WITH sigueD21 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso tabla21 THEN
      ASK D22
    ELSE
      ASK mensaje trampa
    END
  END
WITH sigueD22 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D23
  END
WITH sigueD23 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso tabla23 AND paso pizarra23 THEN
      ASK D24
    ELSE
      ASK mensaje trampa
    END
  END
WITH sigueD24 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso pizarra24 THEN
      ASK D25
    ELSE
      ASK mensaje trampa
    END
  END
WITH sigueD25 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D26
  END
WITH sigueD26 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso pizarra26 AND paso dibujot1 THEN
      ASK D27
    ELSE
      ASK mensaje trampa
    END
  END
WITH sigueD27 SIMPLE
```

```
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    declde := TRUE
  WHEN NONE FOUND
    ASK problemas
    CHAIN "datos.knb"
  FIND END
END
WITH hacIaP154 SIMPLE
WHEN CHANGED
BEGIN
  ASK PresentaP154
END
WITH plz23 SIMPLE
WHEN CHANGED
BEGIN
  paso pizarra23 := TRUE
  ASK P23
END
WITH plz24 SIMPLE
WHEN CHANGED
BEGIN
  ASK P24
END
WITH plz26 SIMPLE
WHEN CHANGED
BEGIN
  paso pizarra26 := TRUE
  ASK P26
END
WITH adP24 SIMPLE
WHEN CHANGED
BEGIN
  paso pizarra24 := TRUE
  ASK P24i
END

ATTRIBUTE clerra tabla window SIMPLE
WHEN CHANGED
BEGIN
  visible OF tabla window := FALSE
```

```
END

ATTRIBUTE cierra dibujo window SIMPLE
WHEN CHANGED
BEGIN
  visible OF dibujo window := FALSE
END

ATTRIBUTE cambia SIMPLE
WHEN CHANGED
BEGIN
  IF bandera = FALSE THEN
    BEGIN
      output OF dibujo window := dib1
      bandera := TRUE
    END
  ELSE
    BEGIN
      output OF dibujo window := dib2
      bandera := FALSE
    END
  END
END

ATTRIBUTE salida SIMPLE
WHEN CHANGED
BEGIN
  CHAIN "menu.knb"
END

ATTRIBUTE decide SIMPLE
WHEN CHANGED
BEGIN
  IF ex4 OF dB3 USR 1 = 10 THEN
    CHAIN "menu.knb"
  ELSE
    CHAIN "ex4.knb"
  END
END
```

P155.KNB

```
CLASS presenta
WITH adepl55 SIMPLE
WHEN CHANGED
BEGIN
```

```
    ASK D29
  END
WITH sigueD29 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D30
  END
WITH sigueD30 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D31
  END
WITH sigueD31 SIMPLE
  WHEN CHANGED
  BEGIN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
      decide := TRUE
    WHEN NONE FOUND
      ASK problemas
      CHAIN "datos.knb"
    FIND END
  END
WITH haciaP195 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK PresentaP195
  END

ATTRIBUTE salida SIMPLE
  WHEN CHANGED
  BEGIN
    CHAIN "menu.knb"
  END

ATTRIBUTE decide SIMPLE
  WHEN CHANGED
  BEGIN
    IF ex5 OF dB3 USR 1 = 10 THEN
      CHAIN "menu.knb"
    ELSE
      CHAIN "ex5.knb"
    END
  END
```


P156.KNB

CLASS animacion

WITH anima3 SIMPLE

WITH para3 SIMPLE

WITH numero3 NUMERIC

WITH total3 NUMERIC

WITH paso3 SIMPLE

WITH dib4 frames PICTURE

ARRAY SIZE 9

WITH anima4 SIMPLE

WITH para4 SIMPLE

WITH paso4 SIMPLE

WITH total4 NUMERIC

WITH numero4 NUMERIC

WITH dib3 frames PICTURE

ARRAY SIZE 8

CLASS muestras

WITH muestra tabla34 SIMPLE

WHEN CHANGED

BEGIN

paso tabla34 := TRUE

visible OF tabla window := TRUE

output OF tabla window := tabla3

END

WITH muestra dibujo34 SIMPLE

WHEN CHANGED

BEGIN

paso dibujo34 := TRUE

visible OF dibujo window := TRUE

output OF dibujo window := dib1

END

WITH muestra dibujo35 SIMPLE

WHEN CHANGED

BEGIN

paso dibujo35 := TRUE

visible OF dibujo window := TRUE

output OF dibujo window := dib2

END

WITH muestra dibujo33 SIMPLE

WHEN CHANGED

BEGIN

```
paso dibujo33 := TRUE
visible OF dibujo window := TRUE
output OF dibujo window := dib4
END
WITH muestra dibujo33b SIMPLE
WHEN CHANGED
BEGIN
  paso dibujo33b := TRUE
  visible OF dibujo window := TRUE
  output OF dibujo window := dib5
END

CLASS presenta
WITH adept6 SIMPLE
WHEN CHANGED
BEGIN
  ASK D33
END
WITH sigueD33 SIMPLE
WHEN CHANGED
BEGIN
  IF paso dibujo33 AND paso dibujo33b THEN
    ASK D34
  ELSE
    ASK mensaje trampa
  END
END
WITH sigueD34 SIMPLE
WHEN CHANGED
BEGIN
  IF paso tabla34 AND paso dibujo34 THEN
    ASK D35
  ELSE
    ASK mensaje trampa
  END
END
WITH sigueD35 SIMPLE
WHEN CHANGED
BEGIN
  IF paso dibujo35 THEN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
      decide := TRUE
    WHEN NONE FOUND
      ASK problemas
```

```
    CHAIN "datos.knb"
  FIND END
  ELSE
    ASK mensaje trampa
  END
  WITH haciaP156 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK PresentaP156
  END

ATTRIBUTE cierra tabla window SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF tabla window := FALSE
  END

ATTRIBUTE cierra dibujo window SIMPLE
  WHEN CHANGED
  BEGIN
    anima3 OF animacion := FALSE
    anima4 OF animacion := FALSE
    visible OF dibujo window := FALSE
  END

ATTRIBUTE cerrar dib5 window SIMPLE
  WHEN CHANGED
  BEGIN
    visible OF dibujo window := FALSE
  END

ATTRIBUTE salida SIMPLE
  WHEN CHANGED
  BEGIN
    CHAIN "menu.knb"
  END

ATTRIBUTE decide SIMPLE
  WHEN CHANGED
  BEGIN
    IF ex6 OF dB3 USR 1 = 10 THEN
      CHAIN "menu.knb"
    ELSE
      CHAIN "ex6.knb"
    END
  END
```

DEMON para animar3
IF anima3 OF animacion
THEN picture OF planeb := dib3 frames[numero3 OF animacion] OF animacion
AND numero3 OF animacion := (numero3 OF animacion MOD total3 OF animacion) + 1
AND anima3 OF animacion := anima3 OF animacion

DEMON para parar3
IF para3 OF animacion
THEN anima3 OF animacion := FALSE

DEMON para paso3
IF paso3 OF animacion
THEN anima3 OF animacion := FALSE
AND picture OF planeb := dib3 frames[numero3 OF animacion] OF animacion
AND numero3 OF animacion := (numero3 OF animacion MOD total3 OF animacion) + 1

DEMON para animar4
IF anima4 OF animacion
THEN picture OF picturebox plane := dib4 frames[numero4 OF animacion] OF animacion
AND numero4 OF animacion := (numero4 OF animacion MOD total4 OF animacion) + 1
AND anima4 OF animacion := anima4 OF animacion

DEMON para parar4
IF para4 OF animacion
THEN anima4 OF animacion := FALSE

DEMON para paso4
IF paso4 OF animacion
THEN anima4 OF animacion := FALSE
AND picture OF picturebox plane := dib4 frames[numero4 OF animacion] OF animacion
AND numero4 OF animacion := (numero4 OF animacion MOD total4 OF animacion) + 1

P157.KNB

CLASS animacion
WITH total6 NUMERIC
WITH total6b NUMERIC
WITH numero6 NUMERIC
WITH numero6b NUMERIC
WITH anima6 SIMPLE
WITH anima6b SIMPLE
WITH para6 SIMPLE
WITH para6b SIMPLE

```
WITH paso6 SIMPLE
WITH paso6b SIMPLE
WITH dib6 frame PICTURE
  ARRAY SIZE 13
WITH dib6b frame PICTURE
  ARRAY SIZE 14
WITH anima7 SIMPLE
WITH anima7b SIMPLE
WITH dib7 frame PICTURE
  ARRAY SIZE 13
WITH dib7b frame PICTURE
  ARRAY SIZE 14
WITH numero7 NUMERIC
WITH total7 NUMERIC
WITH numero7b NUMERIC
WITH total7b NUMERIC
WITH para7 SIMPLE
WITH para7b SIMPLE
WITH paso7 SIMPLE
WITH paso7b SIMPLE
```

CLASS muestra

```
WITH dibujo37 SIMPLE
  WHEN CHANGED
  BEGIN
    paso dibujo37 := TRUE
    visible OF dibujo window := TRUE
    output OF dibujo window := dib37a
  END
WITH dibujo39 SIMPLE
  WHEN CHANGED
  BEGIN
    paso dibujo39 := TRUE
    visible OF dibujo window := TRUE
    output OF dibujo window := dib4
  END
WITH dibujo38 SIMPLE
  WHEN CHANGED
  BEGIN
    paso dibujo38 := TRUE
    visible OF dibujo window := TRUE
    output OF dibujo window := dib6
  END
WITH dibujo37i SIMPLE
  WHEN CHANGED
```

```
BEGIN
  paso dibujo371 := TRUE
  visible OF dibujo window := TRUE
  output OF dibujo window := dib371
END
WITH dibujo38a SIMPLE
WHEN CHANGED
BEGIN
  paso dibujo38a := TRUE
  visible OF dibujo window := TRUE
  output OF dibujo window := dib7
END
WITH dibujo38b SIMPLE
WHEN CHANGED
BEGIN
  paso dibujo38b := TRUE
  visible OF dibujo window := TRUE
  output OF dibujo window := dib8
END

CLASS presenta
WITH sigueD36 SIMPLE
WHEN CHANGED
BEGIN
  IF paso pizarra36 THEN
    ASK D37
  ELSE
    ASK mensaje trampa
  END
END
WITH sigueD37 SIMPLE
WHEN CHANGED
BEGIN
  IF paso dibujo37 AND paso dibujo371 THEN
    ASK D38
  ELSE
    ASK mensaje trampa
  END
END
WITH sigueD38 SIMPLE
WHEN CHANGED
BEGIN
  IF paso dibujo38 AND paso dibujo38a AND paso dibujo38b THEN
    ASK D39
  ELSE
    ASK mensaje trampa
  END
END
```

```
WITH sigueD39 SIMPLE
WHEN CHANGED
BEGIN
  IF paso dibujo39 THEN
    ASK D40
  ELSE
    ASK mensaje trampa
  END
WITH sigueD40 SIMPLE
WHEN CHANGED
BEGIN
  FIND dB3 USR 1
  LIMIT 1
  WHERE bandera OF dB3 USR 1 <> 0
  WHEN FOUND
    decide := TRUE
  WHEN NONE FOUND
    ASK problemas
    CHAIN "datos.knb"
  FIND END
  END
WITH adep167 SIMPLE
WHEN CHANGED
BEGIN
  ASK D36
  END
WITH piz36 SIMPLE
WHEN CHANGED
BEGIN
  paso pizarra36 := TRUE
  ASK P36
  END

ATTRIBUTE kierra dibujo window SIMPLE
WHEN CHANGED
BEGIN
  visible OF dibujo window := FALSE
  END

ATTRIBUTE kambia4_5 SIMPLE
WHEN CHANGED
BEGIN
  IF bandera = FALSE THEN
    BEGIN
      output OF dibujo window := dib4
```

```
    bandera := TRUE
  END
ELSE
  BEGIN
    output OF dibujo window := dib5
    bandera := FALSE
  END
END
```

```
ATTRIBUTE cambia6 SIMPLE
WHEN CHANGED
BEGIN
  IF bandera2 = FALSE THEN
    BEGIN
      picture OF plane6 := dib6 frame[ 1] OF animacion 1
      output OF dibujo window := dib6
      bandera2 := TRUE
    END
  ELSE
    BEGIN
      picture OF plane6 := dib6b frame[ 1] OF animacion 1
      output OF dibujo window := dib6b
      bandera2 := FALSE
    END
  END
END
```

```
ATTRIBUTE kierra6 SIMPLE
WHEN CHANGED
BEGIN
  anima6 OF animacion := FALSE
  anima6b OF animacion := FALSE
  anima7 OF animacion := FALSE
  anima7b OF animacion := FALSE
  visible OF dibujo window := FALSE
END
```

```
ATTRIBUTE kambiar7 SIMPLE
WHEN CHANGED
BEGIN
  IF bandera3 = FALSE THEN
    BEGIN
      picture OF avion7 := dib7 frame[ 1] OF animacion 1
      output OF dibujo window := dib7
      bandera3 := TRUE
    END
  END
```



```
ELSE
  BEGIN
    picture OF avion7 := dib7b frame[ 1] OF animacion 1
    output OF dibujo window := dib7b
    bandera3 := FALSE
  END
END
```

```
ATTRIBUTE cambia8 SIMPLE
WHEN CHANGED
  BEGIN
    IF bandera4 = FALSE THEN
      BEGIN
        output OF dibujo window := dib8
        bandera4 := TRUE
      END
    ELSE
      BEGIN
        output OF dibujo window := dib8b
        bandera4 := FALSE
      END
    END
  END
```

```
ATTRIBUTE salida SIMPLE
WHEN CHANGED
  BEGIN
    CHAIN "menu.knb"
  END
```

```
ATTRIBUTE declde SIMPLE
WHEN CHANGED
  BEGIN
    IF ex7 OF dB3 USR 1 = 10 THEN
      CHAIN "menu.knb"
    ELSE
      CHAIN "ex7.knb"
    END
  END
```

```
DEMON para animar6
IF anima6 OF animacion
THEN picture OF plane6 := dib6 frame[ numero6 OF animacion] OF animacion
AND numero6 OF animacion := (numero6 OF animacion MOD total6 OF animacion) + 1
AND anima6 OF animacion := anima6 OF animacion
```

```
DEMON para animar6b
```

IF *anima6b* OF *animacion*
THEN *picture* OF *plane6* := *dib6b* frame[*numero6b* OF *animacion*] OF *animacion*
AND *numero6b* OF *animacion* := (*numero6b* OF *animacion* MOD *total6b* OF *animacion*) + 1
AND *anima6b* OF *animacion* := *anima6b* OF *animacion*

DEMON *para* *parar6*
IF *para6* OF *animacion*
THEN *anima6* OF *animacion* := FALSE

DEMON *para* *parar6b*
IF *para6b* OF *animacion*
THEN *anima6b* OF *animacion* := FALSE

DEMON *para* *paso6*
IF *paso6* OF *animacion*
THEN *anima6* OF *animacion* := FALSE
AND *picture* OF *plane6* := *dib6* frame[*numero6* OF *animacion*] OF *animacion*
AND *numero6* OF *animacion* := (*numero6* OF *animacion* MOD *total6* OF *animacion*) + 1

DEMON *para* *paso6b*
IF *paso6b* OF *animacion*
THEN *anima6b* OF *animacion* := FALSE
AND *picture* OF *plane6* := *dib6b* frame[*numero6b* OF *animacion*] OF *animacion*
AND *numero6b* OF *animacion* := (*numero6b* OF *animacion* MOD *total6b* OF *animacion*) + 1

DEMON *para* *animar7*
IF *anima7* OF *animacion*
THEN *picture* OF *avlon7* := *dib7* frame[*numero7* OF *animacion*] OF *animacion*
AND *numero7* OF *animacion* := (*numero7* OF *animacion* MOD *total7* OF *animacion*) + 1
AND *anima7* OF *animacion* := *anima7* OF *animacion*

DEMON *para* *animar7b*
IF *anima7b* OF *animacion*
THEN *picture* OF *avlon7* := *dib7b* frame[*numero7b* OF *animacion*] OF *animacion*
AND *numero7b* OF *animacion* := (*numero7b* OF *animacion* MOD *total7b* OF *animacion*) + 1
AND *anima7b* OF *animacion* := *anima7b* OF *animacion*

DEMON *para* *parar7*
IF *para7* OF *animacion*
THEN *anima7* OF *animacion* := FALSE

DEMON *para* *parar7b*
IF *para7b* OF *animacion*
THEN *anima7b* OF *animacion* := FALSE

```
DEMON para paso7
IF paso7 OF animacion
THEN anima7 OF animacion := FALSE
AND picture OF avion7 := dib7 frame[ numero7 OF animacion] OF animacion
AND numero7 OF animacion := (numero7 OF animacion MOD total7 OF animacion) + 1
```

```
DEMON para paso7b
IF paso7b OF animacion
THEN anima7b OF animacion := FALSE
AND picture OF avion7 := dib7b frame[ numero7b OF animacion] OF animacion
AND numero7b OF animacion := (numero7b OF animacion MOD total7b OF animacion) + 1
```

P158.KNB

```
CLASS presenta
WITH sigueD42 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D43
  END
WITH sigueD43 SIMPLE
  WHEN CHANGED
  BEGIN
    IF paso pizarra43 THEN
      ASK D44
    ELSE
      ASK mensaje trampa
    END
  END
WITH sigueD44 SIMPLE
  WHEN CHANGED
  BEGIN
    ASK D45
  END
WITH sigueD45 SIMPLE
  WHEN CHANGED
  BEGIN
    FIND dB3 USR 1
    LIMIT 1
    WHERE bandera OF dB3 USR 1 <> 0
    WHEN FOUND
      decide := TRUE
    WHEN NONE FOUND
      ASK problemas
      CHAIN "datos.knb"
```

```
FIND END
END
WITH adepl168 SIMPLE
WHEN CHANGED
BEGIN
  ASK D42
END
WITH hac1aP158 SIMPLE
WHEN CHANGED
BEGIN
  ASK PresentaP158
END
WITH sigueP43 SIMPLE
WHEN CHANGED
BEGIN
  paso plzarra43 := TRUE
  ASK P43
END
```

```
ATTRIBUTE salida SIMPLE
WHEN CHANGED
BEGIN
  CHAIN "menu.knb"
END
```

```
ATTRIBUTE decide SIMPLE
WHEN CHANGED
BEGIN
  IF ex8 OF dB3 USR 1 = 10 THEN
    CHAIN "menu.knb"
  ELSE
    CHAIN "ex8.knb"
  END
END
```

P159.KNB

```
CLASS presenta
WITH adepl169 SIMPLE
WHEN CHANGED
BEGIN
  ASK D47
END
WITH sigueD47 SIMPLE
WHEN CHANGED
```

```
BEGIN
  ASK D48
END
WITH sigueD48 SIMPLE
  WHEN CHANGED
    BEGIN
      ASK D49
    END
  WITH sigueD49 SIMPLE
    WHEN CHANGED
      BEGIN
        ASK D50
      END
    WITH sigueD50 SIMPLE
      WHEN CHANGED
        BEGIN
          IF paso pizarra50 THEN
            FIND dB3 USR 1
            LIMIT 1
            WHERE bandera OF dB3 USR 1 <> 0
            WHEN FOUND
              decide := TRUE
            WHEN NONE FOUND
              ASK problemas
              CHAIN "datos.knb"
            FIND END
          ELSE
            ASK mensaje trampa
          END
        WITH haclaP159 SIMPLE
          WHEN CHANGED
            BEGIN
              ASK PresentaP159
            END
          WITH sigueP50 SIMPLE
            WHEN CHANGED
              BEGIN
                paso pizarra50 := TRUE
                ASK P50
              END
            ATTRIBUTE salida SIMPLE
              WHEN CHANGED
                BEGIN
                  CHAIN "menu.knb"
```

END

ATTRIBUTE *decide* SIMPLE

WHEN CHANGED

BEGIN

IF *ex9* OF *dB3* *USR 1 = 10* THEN

CHAIN "menu.knb"

ELSE

CHAIN "*ex9.knb*"

END