

31



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA

GUIA INTELIGENTE PARA UN TUTORIAL DE
FÍSICA A NIVEL MEDIO SUPERIOR

T E S I S

PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

P R E S E N T A N

JULIA ESTHER CUÉLLAR VILLEGAS
MARTHA JOSEFINA FERNÁNDEZ MARTÍNEZ
ROSALBA RODRÍGUEZ CHÁVEZ

DIRECTOR DE TESIS:
DR. FELIPE LARA ROSANO.

MEXICO, D.F.

2002

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.

PAGINACIÓN

DISCONTINUA

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Hortir Lucio Jiménez
Hortir

FECHA: 13/12/02

FIRMA: [Firma]

AGRADECIMIENTOS

Este esfuerzo no es sólo de 3 personas. Aunque es imposible mencionar a todos aquellos cuyo apoyo incondicional hemos recibido a través de estos años, queremos agradecer especialmente a:

Dios,
Nuestra querida UNAM,
Profesores,
Dr. Felipe Lara Rosano,
Dr. Fernando Gamboa,
Ing. Rosalba Juárez,
Mto. Nicolás Kemper,
Dr. Fermín Viniegra,
Fis. Juan Velásquez Torres,
Dr. José Luis Pérez Silva,
Ana Viniegra,
Francisco Caviedes

Así como nuestros padres, familiares, amigos, personal del CECADET y a todos aquellos con los que cruzamos caminos en algún punto, y cuya coincidencia nos llevó a ser lo que hoy somos.

Por su aliento, consejos y dirección.....Gracias!!

INDICE

Introducción	i
Capítulo 1. Inteligencia Artificial.	
Introducción Inteligencia Artificial.	1
1.1 Definición de Inteligencia Artificial.	2
1.2 Historia de la Inteligencia Artificial.	5
1.3 Ramas de la inteligencia artificial.	7
1.4 Aplicaciones de la inteligencia artificial.	13
Capítulo 2 Ingeniería del conocimiento y Sistemas Expertos.	
2.1 Ingeniería del conocimiento.	20
2.1.1 Adquisición del conocimiento.	21
2.1.1.1 Métodos para la adquisición del conocimiento.	22
2.1.1.2 Mapas de conocimiento.	26
2.1.2 Representación del conocimiento.	29
2.1.2.1 Esquemas de representación del conocimiento.	30
2.2 Introducción a los Sistemas Expertos.	36
2.2.1 Historia.	37
2.2.2 Definición.	42
2.2.3 Estructura de un Sistema Experto.	44
Capítulo 3. Sistemas Expertos en la Educación	
3.1 Papel de un Sistema Experto en la Educación.	52
3.2 Sistemas expertos que adiestran ó enseñan.	56
3.3 Software Educativo de Sistemas Expertos Tutoriales de Física.	57
3.4 Ventajas y Limitaciones de los Sistemas Expertos Tutoriales que controlan el Software Educativo	62
3.5 La capacidad de los Sistemas Expertos actuales para modelar la empatía del profesor y alumno	63
Capítulo 4. Desarrollo de la "Guía Inteligente para un Tutorial de Física a nivel medio superior"	
4.1 Análisis del problema.	65
4.2 Planteamiento del problema.	66
4.3 Diseño e implementación de los módulos	72
4.3.1 Módulo del profesor(Adquisición de conocimiento).	72
4.3.1.1 Algoritmo del "Módulo del profesor".	76
4.3.1.2. Herramientas de desarrollo.	76
4.3.2 Prototipo del "Módulo del profesor".	78
4.4 Módulo del "Cascarón del Tutor".	87
4.4.1 Herramientas de desarrollo.	89
4.4.2 Prototipo del módulo "Cascarón del tutor".	90
4.4.3 Algoritmo del módulo "Cascarón del tutor".	91
4.5 Módulo Demostrativo de juegos didácticos y material multimedia.	92
4.5.1 Algoritmo del juego interactivo "Mi Pueblito".	94
4.5.2 Algoritmo del juego interactivo "Morita la Pulguita".	95
4.5.3 Herramientas de desarrollo.	95

4.5.4 Prototipos de juegos didácticos.	96
Capítulo 5. Pruebas	
5.1 Captura de Conocimiento.	99
5.2 Integración de los archivos generados por la "Herramienta del profesor".	105
Capítulo 6. Resultados y Conclusiones	111
Apéndice A. Diagramas de flujo de los módulos.	114
Apéndice B. Código fuente de los módulos.	121
Apéndice C. Guión del Dr. Fermin Viniegra Heberlein.	151
Bibliografía	159

INTRODUCCIÓN

Desde la década de los setenta, las computadoras han sido ampliamente utilizadas en el área de la enseñanza. Sus diversas aplicaciones han ido desde la evaluación de alumnos, la enseñanza por demostración (resolución de fórmulas y otros conceptos matemáticos), hasta la tutela inteligente. En efecto, el desarrollo de software educativo de calidad ha sido un proyecto largamente acariciado en los países del mundo, debido a las ventajas potenciales que éste presenta, pues a diferencia de otros medios utilizados hasta hace apenas unos años (correspondencia, televisión educativa, programas tutoriales, etc.), la computadora presenta una capacidad de interacción y respuesta más flexible, y por lo tanto, poderosa. En adjunción a esto, está el hecho de que el material desarrollado puede ser distribuido a muy bajos costos. Como podrá entenderse, esta combinación de factores lo convierte en un producto muypreciado.

Desafortunadamente, el desarrollo de este tipo de materiales se ha revelado como un proceso complejo, que solo en raras y valiosas ocasiones, ha satisfecho las expectativas. De hecho, la experiencia recogida muestra que, a pesar de las múltiples e innegables ventajas que el software educativo ofrece, y de la importancia que éste ha tomado en algunas prestigiadas instituciones de enseñanza, las expectativas generadas a su alrededor no han sido del todo satisfechas: tasas de aprendizaje promedio, mala comprensión del material, muy poca reflexión por parte del alumno, etc. (Bennett, F. 1999).

Así, uno puede comprobar a partir de los casos reseñados en la literatura que, salvo raras excepciones, el uso de la computadora no incrementa las tasas de aprendizaje en los alumnos, ni mejora de manera significativa la comprensión de los conceptos presentados (Beichner R, 1996; Guillies A., Sinclair B. & al. 1996; Lorella M., Kane. J., & al. 1994; Prosser M., Walter P. & al. 1996; Stewart M. Gregory J. 1997; etc.)

El bajo rendimiento en este tipo de software representa una interrogante si se considera el esfuerzo y el tiempo invertido en su desarrollo; lo cuidado y revisado del material que integra; así como la capacidad de las computadoras para manipular datos, trazar gráficas, reproducir sonidos y generar ambientes virtuales que faciliten la presentación de los conceptos más abstractos y variados.

En el Centro de Ciencias Aplicadas y Desarrollo Tecnológico se ha trabajado en los últimos años alrededor de estas cuestiones, tratando de identificar los errores cometidos, y las oportunidades y retos por alcanzar.

Una de sus hipótesis para explicar este fenómeno es que el desarrollo del software educativo se basa en una falsa premisa: las computadoras son fáciles de usar, por ende, el software educativo también es fácil de utilizar. Desafortunadamente, esto es falso. Se trata, en efecto, de una apreciación sustentada en la creencia de que las nuevas tecnologías (ej. interfaces gráficas, videos, sonidos, imágenes animadas, realidad virtual, etc.), facilitan de facto el uso de las computadoras. Sin embargo, esto no es necesariamente verdadero, puesto que cada nueva tecnología trae consigo problemas que es necesarios estudiar.

Esta excesiva confianza en la tecnología ha dado como resultado la realización de sistemas interactivos que, a pesar de integrar imágenes y sonidos vistosos, son difíciles (en ocasiones imposible), de utilizar. En el caso específico del software educativo, podemos observar que los editores han preferido privilegiar los aspectos gráficos e interactivos de sus aplicaciones, descuidando dos puntos a nuestro parecer esenciales:

1) Las dificultades que encuentran los alumnos en la utilización del software educativo. Creemos que mucho del esfuerzo realizado por el estudiante esta enfocado a entender como funciona el programa, y no a entender el material que éste contiene.

2) La falta de un análisis sobre cuáles herramientas deben ser integradas en un software educativo, de manera que el alumno cuente con el material necesario para comprender y desarrollar los conceptos presentados, pero sin convertirse en un observador pasivo.

Así, nuestros trabajos se basan en la idea que una mejoría en el desempeño de la computadora, como herramienta pedagógica, no se basa en el uso indiscriminado de nuevas tecnologías, sino en un análisis del desempeño del alumno frente a la computadora, así como en su participación y consideración a todo lo largo del ciclo de concepción.

Así, en el Laboratorio de Cibernética del CCADET se ha estado trabajando alrededor del desarrollo de software educativo de calidad centrado en el estudiante planteándose cuatro preguntas: 1) ¿Qué es un software educativo de calidad? 2) ¿Cómo debe estar conformado un equipo de desarrollo de software educativo? 3) ¿Cómo debe desarrollarse el Software Educativo? 4) ¿Cómo integrar técnicas de inteligencia artificial de manera eficiente?

Bajo este esquema, ya se han desarrollado diversas aplicaciones educativas en el CCADET; en particular, distintas versiones de un sistema para la enseñanza de la física. Este sistema ha sido probado con alumnos de nivel medio superior y, después de varias modificaciones, se ha llegado a una versión madura y estable que ha logrado una buena aceptación por parte de los estudiantes. Sin embargo, para llegar a dicha versión se tuvo que pasar por un tiempo de desarrollo demasiado extenso; observándose que los integrantes del equipo (Expertos de la materia, y desarrolladores) tenían serios problemas de comunicación y coordinación. Los expertos tienen múltiples ocupaciones, los desarrolladores no logran avanzar lo suficiente, hay problemas de interpretación, etc. todo lo cual afecta al desarrollo del software. Así, aun cuando los productos generados logran, en promedio, satisfacer las expectativas, se trata de un enfoque que sin embargo, ha mostrado una limitante importante, y es que reduce de facto sus posibilidades reales de producir software educativo de manera sostenida: el tiempo de desarrollo.

Por todo lo anterior, y aprovechando algunas de las técnicas de la Ingeniería de Conocimiento e Inteligencia Artificial se planteó la posibilidad de desarrollar una "Guía Inteligente para un Tutorial de Física a nivel medio superior", la cual pretende ayudar a el equipo de desarrollo a mejorar los tiempos de producción, y la comunicación entre los diseñadores de software educativo y el alumno.

Esta guía comprende:

- a) Un cascarón para generar tutoriales por parte de un equipo de desarrollo.
- b) Un módulo de adquisición de conocimiento del experto(profesor).
- c) Un módulo interactivo que introduce y ejemplifica el empleo de juegos interactivos como herramientas de aprendizaje.

Por lo anterior se trata de desarrollar una serie de herramientas capaces de: capturar la información del profesor; incorporar el conocimiento sobre los alumnos; ser capaz de reaccionar ante las respuestas del alumno. Es por todas estas razones que decidimos llamarla "Guía Inteligente".

Los trabajos realizados se desarrollan en el presente documento de la siguiente manera:

En el capítulo 1 se exponen y describen las definiciones, historia, Ramas, y aplicaciones de la Inteligencia Artificial .

En el capítulo 2 se exponen los temas de Ingeniería de Conocimiento y Sistemas Expertos, donde lo más representativo de Ingeniería de conocimiento son los métodos de Adquisición de conocimiento que permiten construir aplicaciones y sistemas de aprendizaje basados en conocimiento.

En el capítulo 3 se describe el papel que juega un Sistema Experto Tutorial, así como sus ventajas y desventajas dentro del software educativo.

En el capítulo 4 se describe el diseño de los tres módulos que comprende esta "Guía Inteligente para un tutorial de Física a nivel medio superior".

Cabe hacer mención que las herramientas generadas le dan la posibilidad al profesor de crear su tutor de manera más independiente y en el menor tiempo posible, y al alumno obtener un material interactivo, y motivacional que complementa y apoya su aprendizaje.

Finalmente en el capítulo 5 se exponen los resultados alcanzados y en el capítulo 6 se presentan las conclusiones obtenidas con el desarrollo de este trabajo.

Capítulo 1. INTELIGENCIA ARTIFICIAL.

INTRODUCCIÓN.

El término "inteligencia artificial" (I.A.) se ha popularizado para referirnos a una disciplina de las Ciencias de la computación, que tiene que ver con el esfuerzo que decenas de científicos de distintos países han venido realizando durante muchos años para dotar a las computadoras de "inteligencia". Esta frase, ("dotar a las computadoras de inteligencia") suele producir una reacción de asombro en muchas personas, aunque a veces por motivos diferentes: "¿Pero, es que las computadoras no son inteligentes?", se preguntarán algunos que han visto o leído demasiadas historias de ciencia ficción. Habrá que contestarles que no, para referirnos al hecho de que la computadora ordinariamente programada no es más que un instrumento muy rápido y confiable para hacer operaciones aritméticas o manipular grandes volúmenes de información.

Para que una computadora merezca el adjetivo de "inteligente", debe ser capaz de realizar acciones que, si fuesen realizadas por un ser vivo, diríamos que requieren de inteligencia, (ej. jugar ajedrez, mantener un diálogo con otro ser considerado también inteligente, resolver algún rompecabezas, reconocer una presa, etc).

Para otras personas, la fuente del asombro al ver asociadas las palabras "inteligencia" y "artificial" consiste en el hecho de que para ellas la inteligencia y las máquinas son conceptos esencialmente incompatibles: Las computadoras pueden hacer operaciones aritméticas porque para eso sólo se necesita ser capaz de manipular números en cierta forma mecánica; pero la inteligencia, a diferencia de esta capacidad, requiere iniciativa intelectual, creatividad, inventiva, y eso, desde luego, sólo lo pueden tener los seres humanos, de ninguna manera las máquinas. Las computadoras hacen lo que sus programadores les dicen. Nada más. Hagan lo que hagan, nunca serán conscientes de lo que hacen, nunca sabrán lo que están haciendo; cuando para ser inteligente se requiere ser capaz de elegir conscientemente el propio camino en la solución de problemas. La respuesta que podemos darle a las personas que creen firmemente en esto anterior, es compleja, como se podrá apreciar en la discusión que se seguirá en las siguientes secciones.

Capítulo 1. INTELIGENCIA ARTIFICIAL.

INTRODUCCIÓN.

El término "inteligencia artificial" (I.A.) se ha popularizado para referirnos a una disciplina de las Ciencias de la computación, que tiene que ver con el esfuerzo que decenas de científicos de distintos países han venido realizando durante muchos años para dotar a las computadoras de "inteligencia". Esta frase, ("dotar a las computadoras de inteligencia") suele producir una reacción de asombro en muchas personas, aunque a veces por motivos diferentes: "¿Pero, es que las computadoras no son inteligentes?", se preguntarán algunos que han visto o leído demasiadas historias de ciencia ficción. Habrá que contestarles que no, para referirnos al hecho de que la computadora ordinariamente programada no es más que un instrumento muy rápido y confiable para hacer operaciones aritméticas o manipular grandes volúmenes de información.

Para que una computadora merezca el adjetivo de "inteligente", debe ser capaz de realizar acciones que, si fuesen realizadas por un ser vivo, diríamos que requieren de inteligencia, (ej. jugar ajedrez, mantener un diálogo con otro ser considerado también inteligente, resolver algún rompecabezas, reconocer una presa, etc).

Para otras personas, la fuente del asombro al ver asociadas las palabras "inteligencia" y "artificial" consiste en el hecho de que para ellas la inteligencia y las máquinas son conceptos esencialmente incompatibles: Las computadoras pueden hacer operaciones aritméticas porque para eso sólo se necesita ser capaz de manipular números en cierta forma mecánica; pero la inteligencia, a diferencia de esta capacidad, requiere iniciativa intelectual, creatividad, inventiva, y eso, desde luego, sólo lo pueden tener los seres humanos, de ninguna manera las máquinas. Las computadoras hacen lo que sus programadores les dicen. Nada más. Hagan lo que hagan, nunca serán conscientes de lo que hacen, nunca sabrán lo que están haciendo; cuando para ser inteligente se requiere ser capaz de elegir conscientemente el propio camino en la solución de problemas. La respuesta que podemos darle a las personas que creen firmemente en esto anterior, es compleja, como se podrá apreciar en la discusión que se seguirá en las siguientes secciones.

1.1. DEFINICIÓN DE INTELIGENCIA ARTIFICIAL.

El nacimiento de la Inteligencia Artificial se sitúa en una reunión celebrada en el Dartmouth College (Hanover, EEUU) en 1956, en la que John McCarthy, inventor del lenguaje LISP, y profesor auxiliar de matemáticas, se planteó la posibilidad de construir máquinas inteligentes. Llama la atención que en esa fecha la informática apenas se estaba desarrollando, y ya se planteaba la posibilidad de diseñar mecanismos inteligentes. En aquella reunión se encontraban entre otros, Claude Shannon, padre de la Teoría de la Información; Marvin Minsky, que más tarde demostraría las limitaciones de ciertos modelos de redes neuronales; Herbert Simon, premio Nobel de Economía, quién desarrolló el primer programa de IA.

Pero volviendo a la posibilidad de definir la inteligencia, y siguiendo las palabras de John McCarthy, podemos decir que una buena aproximación podría ser la siguiente: *Capacidad que tiene (por lo menos) el ser humano de adaptarse eficazmente al cambio de circunstancias mediante el uso de información sobre esos cambios.* Esta definición tiene la virtud de explicar por qué tanta gente reaccionó muy negativamente ante los intentos de filósofos de los siglos XVIII y XIX de explicar los fenómenos mentales a partir del paradigma mecanicista propio de la física de Newton. Para ellos, la máquina más típica, a pesar de estar en desarrollo la Revolución Industrial, seguía siendo el molino de viento o, tal vez más universalmente, el reloj, (por ejemplo los de las grandes catedrales, a veces capaces de dar vida momentánea a un conjunto de figuras humanas, o fantásticas, que colaboraban para dar las horas participando en una secuencia de movimientos preestablecidos). Ahora consideramos que un reloj, aún el más elaborado de cualquier catedral medieval, es incapaz de variar su comportamiento dependiendo del cambio de circunstancias del ambiente (por ejemplo, ninguno de los muñequitos que desfilan en lo alto de la fachada saca su paraguas si llueve).

Algunas personas pensarán, sin embargo, que la definición de McCarty es muy amplia, pues de acuerdo con ella el sistema inmunológico del cuerpo humano (o animal) resulta inteligente: también él logra adaptaciones eficaces usando para ello información. El asunto importante aquí es que precisamente la inteligencia hasta ahora ha sido un fenómeno que se ha dado en seres vivientes, y resulta completamente natural que podamos distinguir una continuidad entre

ciertas reacciones biológicas de un nivel relativamente básico y las reacciones más elaboradas de la corteza cerebral.

Tal continuidad no solo es de esperar, sino que parece totalmente necesaria para aquel que se coloque en la posición en que nosotros nos colocamos: considerar los fenómenos intelectuales como fenómenos sumamente complejos. La complejidad, por hipótesis, admite grados; así, si la inteligencia es obra de la complejidad, habrá toda clase de niveles de inteligencia, desde la más elemental hasta la más elaborada, dependiendo del grado de complejidad del organismo, máquina o programa considerados.

Decíamos que fue natural reaccionar contra la identificación de inteligencia con la máquina cuando el prototipo de la máquina era el reloj medieval (el reloj de pulsera contemporáneo, digitalizado y con alarmas que pueden ponerse a voluntad sería un caso menos claro). Sin embargo, los hombres y mujeres del siglo XIX ya tenían un tipo diferente de máquinas entre ellos, capaces de reaccionar ante el cambio de circunstancias de una manera flexible y eficiente, es decir, adaptativa. Nos referimos a un dispositivo esencial para el funcionamiento de las locomotoras a vapor, tan ligadas al desarrollo de la Revolución Industrial: el gobernador, un mecanismo consistente en dos varillas terminadas en esferas de mayor peso. Dicho mecanismo lograba mantener constante la velocidad de la locomotora haciendo uso de la fuerza centrífuga para cerrar la entrada de combustible que calentaba la caldera. Este es uno de los primeros casos, si no el primero, de un aparato autorregulado, basado en el fenómeno conocido como retroalimentación. Parte de la energía de la caldera, una tan pequeña que podemos llamarla información en vez de energía, se desvía para poner en movimiento el gobernador, el cual, al moverse circularmente hace subir las varillas por fuerza centrífuga y consecuentemente cierra la válvula que da entrada al combustible, todo en proporción a la velocidad de la locomotora. Decimos que la energía desviada retroalimenta a la propia máquina, y logra controlarla, es decir, hacerla autorregulada. Un ejemplo más moderno es el termostato que desconecta un aparato de calefacción o de aire acondicionado, cuando la temperatura del cuarto coincide con la que se ha establecido como condición de referencia. Pues bien, decir que la inteligencia puede explicarse mecánicamente no resulta un enunciado tan escandaloso si lo que tenemos presente como máquina no es el reloj, insensible al cambio de circunstancias, sino el gobernador o el termostato, que obviamente están abiertos al mundo y son capaz de adaptarse por lo menos a algunas de sus posibles vicisitudes.

Teniendo en cuenta esto, podemos hacer un primer intento por explicar el término de I.A. exponiendo varias definiciones que han aportado estudiosos del tema:

- **Marvin Minsky** define a la IA como el arte de hacer máquinas que sean capaces de hacer cosas que, en caso de que fueran hechas por seres humanos, requerirían inteligencia.
- **Margaret Boden** define a la IA como el uso de programas y técnicas de programación de computadoras para sacar a la luz los principios de la inteligencia en general y del pensamiento humano en particular.
- **Elaine Rich** define la IA como el estudio de cómo hacer que las computadoras hagan cosas que, por el momento, las personas realizan de una forma más perfecta y eficiente.
- **Barr y Edward Feigenbaum** definen a la IA como la parte de la computación relacionada con el diseño de sistemas de computadoras inteligentes, esto es, sistemas que exhiben las características que asociamos con lo que entendemos por inteligencia en la conducta humana.
- **Buchanan y Shortliffe** definen la IA como la rama de la computación que tiene como métodos para procesar información, la representación del conocimiento usando símbolos en lugar de números y la heurística (reglas basadas en la experiencia).
- La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente (**Luger y Stubblefield, 1993**).
- Un campo de estudio que se enfoca a la explicación y emulación de la conducta inteligente en función de procesos computacionales (**Schalkoff, 1990**).
- El arte de crear máquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia. (**Kurzweil, 1990**).

1.2 HISTORIA DE LA INTELIGENCIA ARTIFICIAL.

El amanecer de una nueva ciencia.

La Inteligencia Artificial nace en 1943 cuando **Warren McCulloch** y **Walter Pitts** propusieron un modelo de neurona del cerebro humano y animal. Estas neuronas nerviosas abstractas proporcionaron una representación simbólica de la actividad cerebral.

Más adelante, **Norbert Wiener** unió éstas ideas junto con otras, dentro del mismo campo, que se llamó "Cibernética". De aquí nacería, en los años 50, la Inteligencia Artificial.

Los primeros investigadores de esta innovadora disciplina tomaron como base la neurona formalizada de **McCulloch** y postularon que :

" El cerebro es un solucionador inteligente de problemas, de modo que imitemos al cerebro".

Desafortunadamente, la enorme complejidad de éste volvió la tarea prácticamente imposible, sin mencionar que ni el hardware, ni el software de la época estaban a la altura para realizar semejantes proyectos.

Charles Babbage, nacido en 1792, suele ser considerado el padre de la informática moderna. Como tal lo es también de la IA.

Este personaje, tras crear en 1820 la Royal Astronomical Society, se vió obligado a recopilar tablas de referencia. Este trabajo llegó a desesperarle por lo que inventó dos máquinas de calcular así como la Máquina Diferencial (Difference Engine) y la Máquina Analítica (Analytical Engine).

Ada Lovelace, colega de Babbage, tradujo la ponencia realizada en 1842 por un ingeniero militar italiano, L. F. Menabrea, sobre la Máquina Analítica, añadiendo algunas notas complementarias. Esto fue muy importante para el futuro de la computación electrónica, pues quedaron demostrados en términos mecánicos los componentes esenciales para cualquier sistema de computación de propósito general:

- Entrada de datos.
- Almacenamiento de los datos.

- Unidad Aritmética.
- Unidad de Control.
- Salida de datos.

Alan Turing contribuyó notablemente a la aparición de la IA, y de hecho se le considera uno de sus padres. En 1937 publicó una ponencia sobre "números computables", donde expuso el concepto de la "máquina universal de Turing", afirmando que la máquina puede desarrollar cualquier procedimiento matemático siempre que se le proporcione una tabla adecuada de instrucciones.

Turing, después de trabajar durante la Segunda Guerra Mundial en el descifrado de codificaciones secretas en Bletchley Park, marchó al National Physical Laboratory, en Teddington, para contribuir en el diseño de la **ACE** (Automatic Computing Engine).

En 1945 **John Von Neumann**, que contribuyó a caldear el debate de la IA al introducir el concepto de sistema informático auto-reproducible, comenzó el diseño del **EDVAC** (Electronic Discrete Variable Automatic Computer) incluyendo por primera vez el concepto de control por programa almacenado. Esta computadora adoptó la arquitectura de Von Neumann en la que existen dos partes que operan según ciclos de reloj: el CPU que opera sobre los datos, y la MEMORIA, donde se almacenan los datos.

En 1954 **Turing** desarrolló, en la Universidad de Cambridge, sus ideas de que el sistema ACE podría modelar las funciones del cerebro humano, tal y como lo plasmó en el informe "Maquinaria de Computación e Inteligencia".

Como ya se mencionó con anterioridad, el término "Inteligencia Artificial" se cree que fue utilizado por primera vez en 1956 por **John McCarthy** (inventor del lenguaje LISP), profesor auxiliar de matemáticas del Dartmouth College en Hanover (USA). McCarthy convocó a una conferencia, (la **Conferencia de Dartmouth**) que está considerada como el comienzo formal de la IA, en la que pretendía reunir a los investigadores de dicho campo de la computación. Varios de los asistentes (**Allen Newell**, **Herbert Simon**, **Marvin Minsky**) y el propio John McCarthy están reconocidos universalmente como destacados pioneros en IA.

Durante esta reunión, Newell y Simon informaron de sus trabajos realizados, entre los que se encontraba el **Logic Theorist**, programa para la demostración de teoremas que utilizaba símbolos en sustitución

de cantidades numéricas (se le considera el primer programa efectivo de IA).

Terry Winograd, otro profesor de Stanford, desarrolló el programa **SHRDLU** que era capaz de manipular formas simuladas de objetos como si fuesen piezas de Lego (famoso juego danés de construcción por bloques). Este programa admite especificaciones sobre los bloques simulados y permite cualquier reconfiguración de los mismos.

Edward Feigenbaum, de la Universidad de Stanford, desarrolló **DENDRAL**, el primer sistema experto de la historia utilizándolo en el análisis e interpretación de datos de espectrometría de masas para

determinar estructuras moleculares y constituyentes atómicos. Un Sistema Experto es un programa capaz de emular el proceso de razonamiento de un experto humano en un área o campo específico, es en este concepto en el cual ahondaremos en el siguiente capítulo de este trabajo.

1.3 RAMAS DE LA INTELIGENCIA ARTIFICIAL.

Algunas de las aplicaciones más importantes que pueden tener las computadoras dentro del área de la Inteligencia Artificial, son las siguientes:

- Ayudar a los Expertos a Analizar y Diseñar: Algunos programas sirven para auxiliar a médicos en el análisis de ciertos tipos de enfermedades; otros para entender el funcionamiento de circuitos electrónicos; otros más nos auxilian en la configuración de los módulos que conforman sistemas complejos de equipo de cómputo. En el caso concreto de esta tesis, la ayuda consiste en lograr que el profesor (el experto) pueda estructurar un tutorial de su campo específico de conocimiento.
- Entender un lenguaje: Para el ser humano la manera natural de comunicarse es a través del lenguaje. Esto es lo que ha motivado un gran interés por desarrollar esta misma habilidad en las computadoras. Para el entendimiento de un lenguaje natural escrito (como el inglés), se puede utilizar, entre otras, la técnica de palabras clave, esta técnica intenta inferir el significado de la comunicación a partir del propio significado de las palabras clave.

Sin embargo, aún no se ha llegado a en contextos donde las palabras claves utilizadas pueden tener múltiples significados.

- Procesar Imágenes: Computadoras equipadas con los dispositivos adecuados (cámaras de TV etc.), pueden ver lo suficiente para tratar con un espacio limitado, los objetos que ahí se encuentran y la relación que guarda uno con respecto del otro.
- Ayudar a Manufacturar Productos: Actualmente máquinas de propósito específico auxilian en trabajos que el hombre considera peligroso, aburrido, o poco remunerado. El pasar de máquinas de propósito específico a robots inteligentes, requiere de agregar muchas capacidades, una de ellas es la de razonar acerca del movimiento en tres dimensiones, tal como el requerido para mover una caja de un estante a otro en un almacén.

Para poder realizar estas tareas, la I.A. se ha dividido en diversas ramas, entre las que se destacan las Redes Neuronales, la Robótica, Lenguaje Natural, Reconocimiento de Patrones, Algoritmos Genéticos, Planeación, etc.

REDES NEURONALES

Las redes neuronales son dispositivos inspirados en la funcionalidad de las neuronas biológicas, aplicados al reconocimiento de patrones. Se trata de elementos capaces de modelar y efectuar predicciones en sistemas complejos. Así, podemos decir que son un conjunto de técnicas matemáticas que, a partir de la emulación del comportamiento de neuronas biológicas, permiten modelar las conexiones / relaciones entre un conjunto de datos.

Las Redes Neuronales surgieron del movimiento conexionista, que nació junto con la IA simbólica o tradicional. La IA simbólica se basa en que todo conocimiento se puede representar mediante combinaciones de símbolos, derivadas de otras combinaciones que representan verdades incuestionables o axiomas. Así pues, la IA tradicional asume que el conocimiento es independiente de la estructura que maneje los símbolos, siempre y cuando la 'máquina' realice algunas operaciones básicas entre ellos.

Una Red Neuronal: el Perceptrón

Es un conjunto de neuronas de manera que cada una de las entradas del sistema se conectan a cada neurona de salida, produciendo

cada una de ellas su salida individual. A este conjunto se le denomina Perceptrón.

Existen tres métodos de aprendizaje para un Perceptrón:

- **Aprendizaje supervisado:** se presentan al Perceptrón unas entradas con las correspondientes salidas que se quiere que sean aprendidas.

- **Aprendizaje no supervisado:** solo se presentan al Perceptrón las entradas y, para esas entradas, la red debe dar una salida parecida.

- **Aprendizaje por refuerzo:** se combinan los dos anteriores, y cada cierto tiempo se presenta a la red una valoración global de como lo está haciendo.

Podría parecer que el Perceptrón tiene una potencia ilimitada para aprender, pero Minsky y Paper pusieron graves deficiencias del Perceptrón en su libro "Perceptrons". Según ellos el Perceptrón era incapaz de aprender las funciones que no fuesen linealmente separables.

Aunque su estructura varía según el tipo de red, lo más usual es que haya tres capas de neuronas, una de entrada, que recoge los estímulos, otra oculta, que procesa la información, y otra de salida, que ejecuta la respuesta. La figura 1 muestra esta disposición. El problema es que el cerebro humano consta de cien mil millones de neuronas, mientras que las redes artificiales más complejas tienen como mucho mas miles. Además del problema de cómo se conectan éstas entre sí.

Ello no impide que mediante estas redes se puedan cumplir funciones concretas, como reconocimiento de caracteres borrosos, clasificación de patrones y otras.

Sin embargo, las redes neuronales todavía se han de desarrollar mucho. Aún se debe estudiar para que sirven realmente, conocer en que tareas pueden resultar realmente útiles, ya que por ejemplo es difícil saber cuánto tiempo necesita una red para aprender cierta tarea, cuántas neuronas necesitamos como mínimo para realizar cierta tarea, etc.

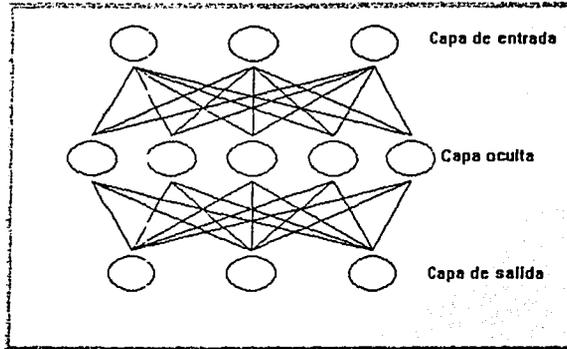


Figura 1. Red Neuronal.

ROBÓTICA

La Robótica es la técnica que aplica la informática al diseño y empleo de aparatos que, en sustitución de personas, realizan operaciones o trabajos, por lo general en instalaciones industriales, en la fabricación de productos, o donde las tareas son repetitivas y aburridas. Los robots son muy solicitados en ambientes peligrosos para el ser humano, como en el manejo de explosivos, altas temperaturas, atmósfera sin la cantidad adecuada de oxígeno y, en general, bajo cualquier situación donde se pueda deteriorar la salud. La mayoría de los robots tienen un brazo con varias uniones móviles y partes prensiles, donde todos sus elementos son controlados por un sistema de control programado para realizar varias tareas bajo una secuencia de pasos preestablecidos. Los investigadores de IA pretenden adicionar al robot métodos y técnicas que le permitan actuar con un pequeño grado de inteligencia, lo cual pretenden lograr con la conjunción de todas las áreas de la IA. Hasta el momento, su coeficiente intelectual sólo les permite realizar tareas simples, como salvar obstáculos y recoger objetos del suelo.

ALGORITMOS GENÉTICOS

La ley de la Naturaleza; "La supervivencia del más apto", ha impulsado la evolución, extinción y supervivencia de seres vivos. Cada organismo tiene inscrito y codificado el conocimiento-biológico-genético

acumulado durante millones de años. Inspirados de este proceso de selección, en una computadora, comenzando por una población inicial de organismos-candidatos para la solución de un problema, éstos se recombinan de manera aleatoria (reproducción), luego de seleccionar los mejores, y después de cierto tiempo o número de generaciones se alcanza una solución suficientemente buena para resolver el problema.

La principal ventaja de los algoritmos genéticos radica en su sencillez. Se requiere poca información sobre el espacio de búsqueda ya que se trabaja sobre un conjunto de soluciones o parámetros codificados (hipótesis o individuos). Se busca una solución por aproximación de la población, en lugar de una aproximación punto a punto. Con un control adecuado podemos mejorar la aptitud promedio de la población, obteniendo nuevos y mejores individuos y, por lo tanto, mejores soluciones.

Se consigue un equilibrio entre la eficacia y la eficiencia. Este equilibrio es configurable mediante los parámetros y operaciones usados en el algoritmo. Así, por ejemplo, bajando el valor del umbral

conseguiremos una rápida solución a cambio de perder en "calidad". Si aumentamos dicho valor, tendremos una mejor solución a cambio de un mayor tiempo consumido en la búsqueda. Es decir, obtenemos una buena relación entre la calidad de la solución y el costo.

Quizás el punto más delicado de todo se encuentra en la definición de la función de evaluación. Al igual que en el caso de la heurística, de su eficacia depende el obtener buenos resultados. El resto del proceso es siempre el mismo para todos los casos.

La programación mediante algoritmos genéticos suponen un nuevo enfoque que permite abarcar todas aquellas áreas de aplicación donde no seamos como resolver un problema, pero si seamos conscientes de que soluciones son buenas y cuales son malas. Desde aplicaciones evidentes, como la biología o la medicina, hasta otros campos como la industria (clasificación de piezas en cadenas de montaje). Los algoritmos genéticos poseen un importante papel en aplicaciones de búsqueda y optimización, pero desde nuestro punto de vista, es en el aprendizaje automático donde encuentra un estupendo marco de trabajo. La capacidad que poseen para favorecer a los individuos que explican bien los ejemplos, a costa de los que no lo hacen, consigue una nueva generación con mejores reglas y, por lo tanto el sistema será capaz de ir aprendiendo a conseguir mejores resultados.

PROCESAMIENTO DE LENGUAJE NATURAL.

Un gran objetivo de la I.A., no carente de una abrumadora complejidad, se centra en el tratamiento del lenguaje natural. Este objetivo consiste en que las máquinas computacionales puedan comunicarse con las personas sin ninguna dificultad de comprensión, ya sea oralmente o por escrito. Aquí encontramos la realización de un sueño largamente alimentado: hablar con las máquinas y que éstas entiendan nuestra lengua y, también, que se hagan entender en nuestra lengua.

La síntesis del lenguaje y el reconocimiento de voz forman dos aspectos del mismo propósito. Los logros que se han conseguido resultan a todas luces parciales e insuficientes, pero alentadores. Ciertas máquinas pueden interpretar las gráficas de textos escritos y reproducirlos oralmente, es decir, leen textos en voz alta. Uno de los avances de este tipo (que sin duda representará una evolución notable en poco tiempo) es el

programa FRUMP, elaborado en la universidad de Yale. FRUMP es capaz de leer historias cortas y resumirlas escuetamente con una gramática correcta y una expresión convincente. Para lograr que las computadoras comprendan la lengua en la que las personas se expresan, es preciso pasar por una dilatada cadena de investigaciones en el campo de la acústica y ondas de lenguaje, análisis fonético y articulación, reglas de formación de frases o análisis sintáctico, el dominio semántico o de los conceptos y, finalmente, el análisis global de los actos de comunicación o pragmática.

El tratamiento del lenguaje natural tendrá una repercusión difícilmente imaginable. Se cree que su aplicación se extenderá al camino de la traducción automática del lenguaje. Los textos o expresiones de una lengua se vertirán de manera automática e inmediata a otra lengua o idioma. Se prevé alcanzar mecánicamente un 90 por ciento de precisión, y el resto del material de traducción será tratado por personal especializado.

Junto al lenguaje, aparece otro objetivo capital, consistente en el tratamiento de gráficos y la visión artificial. De este planteamiento se desgranar aspectos como los de la percepción electrónica, selección y almacenamiento de imágenes, reconocimiento visual de formas y objetos, producción de imágenes y diseño industrial (CAD/CAM), etcétera.

En el siguiente apartado, presentaremos algunos ejemplos prácticos de aplicaciones dentro de las distintas ramas de la I.A.

1.4 APLICACIONES DE LA INTELIGENCIA ARTIFICIAL.

Como ya vimos con anterioridad, algunos temas que han sido tradicionalmente considerados como de interés para la Inteligencia Artificial son, entre otros:

- Resolución de problemas complejos.
- Juegos.
- Análisis de imágenes y escenas.
- Comprensión del lenguaje natural.

Los siguientes son ejemplos de aplicaciones, que denominaremos "clásicas" dentro del ámbito de la I.A.

JUGADORES INTELIGENTES.

En una fecha tan temprana como 1946, Arthur Samuel desarrolló un programa que no sólo jugaba a las damas, sino que mejoraba su juego a medida que jugaba.

La historia de este programa no tiene desperdicio, en la medida que su autor protagonizó una de las versiones más chocantes del cuento de la lechera. Por aquellos años, Samuel deseaba que su facultad adquiriese una computadora, pero su presupuesto resultaba insuficiente. Ni corto ni perezoso elaboró el siguiente plan: construiría una pequeña computadora, y a partir de ésta, un programa capaz de jugar a las damas, que vencería sin dificultad el campeonato mundial de damas; la publicidad que obtendría sería suficiente como para garantizarle una subvención estatal que le permitiría la adquisición de su querida computadora. Si Samuel escogió las damas fue porque lo consideró un juego sencillo de programar. Al fin y al cabo, hay un único tipo de ficha, que se mueve de la misma forma. Pese a esta aparente simplicidad, la gran cantidad de estrategias que se pueden plantear a partir de una posición dada complican notablemente el problema. El resultado de su sueño fue más que previsible: cuando llegó la fecha del campeonato no estaba concluido ni la computadora ni el programa. Tres

años después, el investigador Claude Shannon, famoso por su teoría de la información, escribía la primera obra sobre programación del ajedrez. De esta manera, se daba la señal de salida a una carrera cuya máxima meta era la creación de un programa que conquistara el título mundial de ajedrez.

¿Cómo funciona un programa capaz de jugar a un juego de inteligencia, tal como el ajedrez o las damas? La palabra clave es programación en árbol. La figura 1 muestra una hipotética situación. Las letras A, B y C representan fichas. Para cada una tenemos uno o varios movimientos, que a su vez provocarán movimientos en las piezas de nuestro contrincante. Los números representan la puntuación de cada jugada. Para calcular la jugada óptima se toman cada una de las jugadas propias y a su puntuación se le resta la de la jugada del oponente. Por ejemplo, una jugada de la pieza A nos reporta 9 puntos, pero a continuación la del contrincante le puede suponer 11 puntos, de forma que quedan -2 puntos globales. Por otra parte, una jugada de la figura C nos supone sólo 2 puntos, pero al restarle los -2 de la jugada más óptima del contrincante nos da una ganancia neta de 4 puntos. De esta manera, se puede encontrar la mejor jugada en aquella que proporciona una puntuación neta mayor.

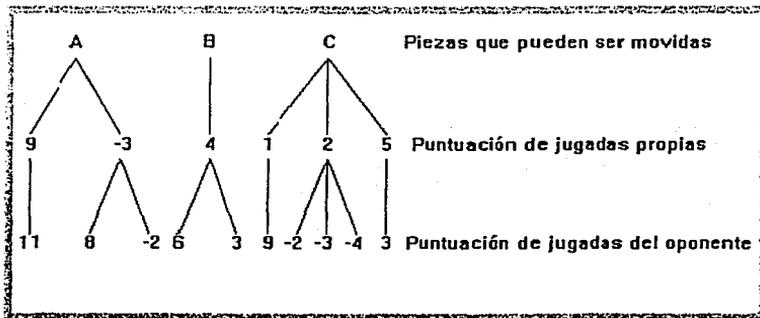


Figura 2. Estructura en árbol para juegos de inteligencia.

Con sólo dos niveles de profundización se tiene un nivel de juego comparable al de un niño de seis años. Ampliando el número de niveles se puede ir mejorando el nivel de juego, a costa de un mayor tiempo de cómputo. Por ello, se han desarrollado técnicas para "podar" el árbol, y

eliminar aquellas ramas que *a priori* se puede saber que no van a dar puntuaciones altas.

PROGRAMAS QUE RAZONAN.

Esta rama es la encargada de desarrollar sistemas con capacidad real o aparente de dialogar de forma inteligente y, consecuentemente, de razonar. Desde siempre ha supuesto la máxima aspiración de los investigadores en este campo, y coincide con la idea de que la mayor parte de la gente tiene de un sistema inteligente. Sin embargo, es importante destacar que, fieles a la filosofía del test de Turing, los investigadores no se preocupan en nuestros días por la cuestión de si tal sistema comprende realmente lo que se le dice o si tiene conciencia de sí mismo, sino de que nosotros podamos comprender lo que él nos dice.

Dentro de este apartado, destacan tres programas que han supuesto auténticos hitos en el proceso de desarrollo de la I.A. Por orden cronológico son el programa **Eliza**, el programa **Shrdlu** y el programa **Racter**.

ELIZA

Corría el año de 1966 cuando Joseph Weizenbaun, en ese entonces, profesor de informática del MIT, daba los últimos toques a su programa Eliza. Durante dos años había trabajado en él, con el único objetivo de crear un programa que parodiase a los psiquiatras de la escuela de Carl Rogers. Estos psiquiatras son los que aparecen con frecuencia en las películas norteamericanas, y que practican una terapia basada en el diálogo de libre asociación de los pacientes, propio del psicoanálisis no directivo.

El programa simula una sesión con un psiquiatra rogeliano; al ejecutarlo, tenemos que introducir unas frases, a las que el programa tiene que dar respuesta. Para lo cual busca en la cadena alfanumérica que se le introduce algunas frases clave, para las que tiene una batería de respuestas estándar, de las que selecciona una al azar. En caso de no encontrar una frase clave busca una palabra clave y, en función de ésta, vuelve a seleccionar al azar una de las respuestas que tiene programadas. Si tampoco encontrase una palabra clave, busca si tiene

almacenada una frase dicha con anterioridad por el usuario, y en función de ésta construye su respuesta.

Por ejemplo, supongamos que en un momento dado se introduce la frase "Mi mujer me odia". El programa toma la frase que viene a continuación de Mi y la almacena tras intercambiar las personas de pronombres y posesivos. Cuando más adelante el programa no encuentre una frase o palabra clave puede recuperar la frase almacenada y contestar algo así como: "¿Tiene eso que ver con el hecho de que su mujer le odia?". Finalmente, si tampoco tiene almacenada ninguna frase el programa, selecciona al azar una frase del tipo "Eso es muy interesante" u otras similares, para mantener la conversación.

SHRDLU

Creado en 1971 por Terry Winograd, un investigador del MIT. A diferencia de Eliza, Shrdlu no se limita a seleccionar respuestas al azar ante cada entrada, sino que lleva a cabo un auténtico proceso de razonamiento. En el mundo de Shrdlu existen tan sólo las figuras geométricas de diferentes colores, figuras que se pueden disponer con distinto orden. Cuando el usuario da las órdenes oportunas mediante un lenguaje coloquial, el programa lleva a cabo las transformaciones que se le piden, e informa de la nueva situación creada.

Fragmento de conversación entre un usuario y el programa:

Humano: ¿Quieres, por favor, apilar los dos bloques rojos, por un lado, y un cubo verde o una pirámide, por otro?

Shrdlu: De acuerdo

Humano: ¿Qué cubo descansa sobre la mesa?

Shrdlu: El cubo grande verde que soporta a la pirámide roja

Humano: ¿Hay algún bloque grande detrás de una pirámide?

Shrdlu: Sí, hay tres: uno grande rojo, un cubo grande verde y el azul

Aunque el diálogo prosigue, este fragmento es suficiente para captar la neta superioridad de este programa frente al de sus antecesores; no sólo parece comprender correctamente las órdenes que se le dan, sino que en todo momento tiene una idea clara de las relaciones espaciales entre los bloques que maneja. ¿Cómo consigue tal cosa? Parafraseando a Dewdney, "La organización interna de Shrdlu..., está formada por subsistemas que analizan gramaticalmente, interpretan y construyen frases, llevan a cabo búsquedas en un diccionario, efectúan análisis semánticos y deducciones lógicas. Todos estos subsistemas están estrechamente entretnejidos, en modo

sumamente estructurado, cuyo efecto conjunto nos da la impresión de una verdadera inteligencia que analizara el mundo de bloques depositados en el tablero".

RACTER

Creado en 1980 por Thomas Etter, un programador-analista de San Francisco y William Chamberlain, un escritor neoyorquino. La intención de ambos era crear un programa que crease cuentos. Posteriormente, lo modificaron para permitirle mantener conversaciones con seres humanos, veamos un extracto de una conversación entre Racter y Chamberlain.

Racter: ¿Hemos hablado antes?

Chamberlain: No

Racter: ¿De qué sexo eres?

Chamberlain: Masculino

Racter: ¿Cuál es tu nombre completo?

Chamberlain: Bill Chamberlain

Racter: ¿Qué eres?

Chamberlain: Cronorreflectometrista

Racter: ¿De dónde eres?

Chamberlain: De la ciudad de Nueva York

Racter: Ya veo. Así que cronorreflectometrista, ¿verdad? Escucha. Gritemos juntos con decisión. Me gusta la cautivadora música italiana. ¿Qué clase de música te gusta a ti?

Chamberlain: el jazz

Racter: El jazz está bien. ¿qué me dices de la clásica?

Chamberlain: Me gusta la música clásica

Racter: ¿Y de qué nacionalidad?

Chamberlain: También me gusta la música italiana

Racter: A mí, personalmente, Palestrina me inflama. Además de Italia, otros países produjeron en el siglo XVII embelesados músicos. Sewall fue un importante y furibundo maestro del Renacimiento. América fue su nación. Yo saco de la música amoroso placer.

El diálogo prosigue por similares derroteros, mezclando sin pudor Bach con las lechugas. En el test de Turing Racter podría pasar por un humano con una marcada tendencia a la divagación. ¿Cuál es el funcionamiento de Racter? Al principio de la conversación hace una serie de preguntas a su interlocutor, cuyas respuestas va almacenando

en una base de datos. En cuanto juzga que dispone de suficiente información empieza a elaborar frases, en las que combina los elementos de que dispone en su memoria con aquellos que ha recibido de su interlocutor, con lo que logra implicarlo en la conversación. Para construir las frases sigue un mismo esquema gramatical, en el que los sustantivos, adjetivos y verbos se eligen al azar entre una amplia variedad que pertenecen a una misma categoría, y los artículos se adecúan a la forma gramatical correspondiente. Esto hace que las frases sean sintácticamente correctas, pero en tanto los sustantivos, adjetivos y verbos se escogen al azar el sentido suele ser disparatado.

Hasta este momento, hemos discutido los tópicos considerados, "clásicos" en el interés de la I.A., pero para cerrar el apartado, debemos poner especial interés en el área que ha dado un buen recibimiento a la I.A. y que es la que nos ocupa: La Educación.

Como sabemos, mucho antes que las computadoras irrumpieran en el campo de la educación, existían otras técnicas de instrucción (ej. la programada, la televisión educativa y los programas tutoriales de audio, etc.), sin embargo, esos enfoques no cumplían con las expectativas acerca de los beneficios positivos y duraderos que traerían a la enseñanza. Con la llegada de las computadoras, con mayor flexibilidad y capacidad de interacción, este panorama cambió, surgiendo los programas de Enseñanza Asistida por Computadora (EAC). De hecho, los programas EAC, mejoraron sustancialmente la efectividad de la enseñanza a distancia; no obstante, su rendimiento debe aún ser mejorado.

Como veremos más a profundidad en el Capítulo 3, Branson (1990) planteó una conceptualización del desarrollo histórico de los paradigmas comunicacionales educativos. Hablaba de el Paradigma de la tradición oral, es decir, en los inicios de la enseñanza, el maestro, centro del conocimiento y la experiencia, transmitía a través de presentaciones orales el conocimiento a los estudiantes. Gradualmente, el conocimiento de base se expandió y los materiales impresos estuvieron disponibles. El flujo comunicativo era unidireccional.

En el Paradigma "contemporáneo", el maestro está aún como fuente principal de conocimiento y sirve ya sea como sustentante de la información o como vigilante y controlador del conocimiento de base. El modelo evolucionó hacia un crecimiento del número de interacciones entre los estudiantes y el maestro. La enseñanza mejoró con materiales instruccionales, tales como libros de texto, libros de trabajo y ayudas audiovisuales. Este modelo ha estado en uso por más de sesenta años y ha funcionado para las necesidades de las anteriores décadas, pero ya

no satisface completamente las necesidades actuales. Pensamos que se debe incrementar la calidad y cantidad de las interacciones entre los estudiantes y las bases de conocimiento y experiencia, con lo que la proporción de procesamiento de la información se incrementará. Esta situación apunta hacia un paradigma más tecnológico.

Branson denomina paradigma tecnológico, a un proceso de enseñanza aprendizaje completamente interactivo. En él, cada estudiante interactúa con el maestro, la base de datos, el sistema experto, el tutor y, también, con otros estudiantes. El estudiante puede aprender ahora directamente de la base de datos y el experto, o de otros estudiantes, sin que el maestro le enseñe directamente. Estudiante y maestro pueden interactuar entre sí o con la base de datos y/o con el experto.

La instrucción centrada en el maestro y en el aula está representada en los modelos pasado y en el modelo presente. La estimación de mayor peso, en el modelo tecnológico, se deriva de la necesidad del cambio del papel del maestro, donde éste pasaría a convertirse en experto.

Segovia (1991) amplía esta noción, al señalar que una de las características distintivas es la unidireccionalidad de la primera y segunda generación, en comparación con la bidireccionalidad de la tercera generación. En realidad, anotamos, en la primera si hay bidireccionalidad, pero retardada en el tiempo. En suma, tal como señala Garrison, aumenta la interacción y el control por parte del estudiante.

De acuerdo a los paradigmas descritos anteriormente en la actualidad debemos centrarnos en el paradigma tecnológico, donde haya bidireccionalidad entre profesor y alumno de tal manera se represente, entienda y procese el conocimiento adecuadamente.

2.- INGENIERÍA DEL CONOCIMIENTO Y SISTEMAS EXPERTOS.

2.1 INGENIERÍA DEL CONOCIMIENTO.

La Ingeniería de conocimiento es considerada un área de las ciencias de la computación, relacionada con la adquisición, representación y el procesamiento de conocimiento. Así, se trata de un área íntimamente relacionada con el de desarrollo de un sistema experto.

Se trata de una disciplina moderna, que ayuda a construir aplicaciones y sistemas orientados al aprendizaje, apoyándonos en metodologías instruccionales y en tecnología de computación.

La ingeniería del conocimiento ha permitido construir aplicaciones informáticas en dominios difíciles para la informática convencional, en los que no se manejan algoritmos formales, sino conocimiento que, en general, no está bien estructurado y que puede ser inconsistente, incompleto e impreciso. Estas aplicaciones reciben el nombre de sistemas basados en el conocimiento.

A continuación se mencionan dos definiciones de Ingeniería del conocimiento¹.

- Definición estricta: La Ingeniería del Conocimiento tiene que ver con la adquisición, representación, validación, inferenciación, explicación y mantenimiento del conocimiento.
- Definición amplia: La Ingeniería del conocimiento describe el proceso de desarrollo y mantenimiento de Sistemas de Inteligencia Artificial.

La Ingeniería de Conocimiento se basa principalmente en la extracción de conocimiento de un experto humano, esto es lo que determina la calidad final del sistema.

¹ Nicolás, M. C. K. V. (1995). Manual de Sistemas Expertos. Perú Lima, p33.

Ingeniero de conocimiento

El Ingeniero de conocimiento es la persona encargada de extraer el conocimiento de un experto y modelarlo de modo tal que una computadora pueda almacenarlo y procesarlo.

El experto es una persona de reconocido prestigio que decide poner a disposición de muchos su experiencia por medio de un sistema. Los ingenieros de conocimiento investigan las tecnologías necesarias para enfrentar las necesidades de administración de conocimiento en un área específica. También deben establecer los procesos por los que se examinan las peticiones de conocimiento, se integra la información, y se regresa el conocimiento a quien lo solicitó.

El proceso de obtención, validación, modelado y procesamiento requiere de varias etapas y entrevistas entre el experto y el ingeniero de conocimiento. A continuación se explican.

2.1.1 ADQUISICIÓN DEL CONOCIMIENTO.

La adquisición del conocimiento es el proceso de extraer, estructurar y organizar conocimiento de una o varias fuentes.

La adquisición del conocimiento es el aspecto de mayor dificultad dentro de la Ingeniería del Conocimiento. Esto se debe principalmente a que el problema de extraer experiencia y conocimiento de un experto humano radica en la comunicación entre un experto humano y el ingeniero de conocimiento que desarrolla el sistema. Acceder a los conocimientos adquiridos por experiencia no es tarea fácil. Para los expertos es un reto verbalizar el conocimiento y la experiencia adquirida, pues con frecuencia no es consciente de la forma en que soluciona un problema y apenas reconoce su conocimiento como tal.

La habilidad de expresar el conocimiento y utilizarlo en un sistema experto es aún más difícil, pues el lenguaje que utiliza un experto no es un lenguaje que la computadora entienda. Por esta discrepancia, el ingeniero de conocimiento, quien conoce como representar el conocimiento en una computadora, debe seleccionar patrones en el conocimiento del experto que puedan ser convertidos en una forma útil para la computadora.

Las metodologías actuales requieren mucho tiempo de desarrollo y mucho tiempo de participación, tanto de los expertos de la aplicación (cuyo talento y tiempo son tan costosos. Que su proceso de toma de decisiones se desea agilizar) como el del ingeniero de conocimiento.

Para que se logre una buena comunicación entre el ingeniero de conocimiento y el experto, es necesario que el Ingeniero de Conocimiento establezca una comprensión global del área, se forme un diccionario mental de los términos esenciales y desarrolle una comprensión básica de los conceptos clave.

En la figura 3 se muestra la interacción entre el ingeniero del conocimiento y el experto durante el proceso de adquisición del conocimiento.

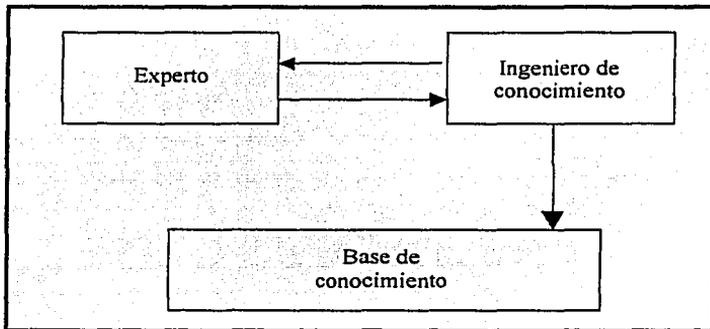


Figura 3. Interacción entre Ingeniero de Conocimiento y experto

2.1.1.1 MÉTODOS PARA LA ADQUISICIÓN DEL CONOCIMIENTO.

Actualmente se emplean los siguientes métodos para adquirir el conocimiento: manuales y automáticos.

MANUALES:

Los métodos manuales pueden ser: entrevistas, protocolos, examen, e informes.

ENTREVISTAS:

Se obtiene el conocimiento mediante el planteamiento de preguntas y problemas. Se trata de una técnica que está guiada por el conocimiento que va aportando el experto.

En este método se tiene una desventaja, pues el experto muchas veces no es consciente del conocimiento que tiene o que se le está pidiendo, por lo que induce a la formalización del conocimiento en múltiples sistemas de representación y en muchos casos innecesario.

En otras ocasiones puede no ser completo e incluso puede presentar inconsistencias.

PROTOCOLOS:

Es una entrevista en la cual se tiene un objetivo específico, la comunicación es organizada y se reducen los problemas de interpretación.

Esta técnica está guiada por el ingeniero de conocimiento, quien sabe que conocimientos necesita y de que forma.

La ventaja de este método es cuando el ingeniero de conocimiento sabe cuál es el problema, ya que permite la representación del conocimiento de manera uniforme.

Por otra parte, la desventaja es que el ingeniero de conocimiento requiere una formación previa sobre determinado problema.

EXAMEN:

En esta técnica el Ingeniero de conocimiento prepara una serie de pruebas para que el experto los resuelva y explique como ha hecho, para así obtener los conocimientos necesarios.

Es una técnica adecuada para completar la base de conocimientos.

INFORMES:

El experto explica en forma oral y escrita los distintos tipos de problemas que enfrenta y cómo los resuelve.

Esta técnica es muy útil en los primeros pasos de desarrollo de la base de conocimientos.

La desventaja es que el experto tiene que dedicar tiempo y esfuerzo a realizar su tarea.

AUTOMÁTICAS:

En la figura 4 se muestra la secuencia a seguir para obtener la base de conocimientos a través de los métodos de adquisición automáticos del conocimiento.

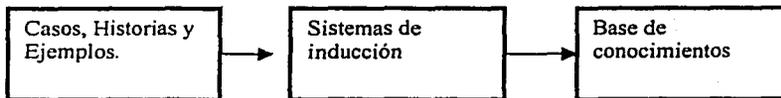


Figura 4. Secuencia de una técnica automática para la adquisición de conocimiento

En este enfoque se le solicitan al experto los conocimientos que se necesitan, empleando una computadora. Así, se requiere de la creación de un subsistema que permita la utilización de un lenguaje "pseudonatural", y que incluya detectores de redundancias, bucles, incoherencias, etc.

A través de un proceso de inducción (a partir de ejemplos particulares, se llega a una conclusión general) el ingeniero de conocimiento obtiene el conocimiento de una serie de "ejemplos tipo" resueltos. Se emplea cuando el experto no está disponible el tiempo necesario. En muchos casos requiere de tablas decisión.

TABLAS DE DECISIÓN.

Es un método para organizar el conocimiento. En él, la experiencia queda organizada como un formato de hoja de cálculo, usando filas y columnas.

La tabla 1 se divide en dos partes:

- 1.- Atributos del problema (factores, causas, condiciones).
- 2.- Análisis del proceso de desarrollo de un sistema basado en conocimiento que consta de cuatro pasos:

- Observación del experto.
- Análisis del problema.
- Descripción del problema.
- Refinamiento del problema.

En cada una de las partes anteriores ocurre una interacción entre el ingeniero de conocimiento y el experto.

En una tablas de decisión se representan los elementos característicos del problema a resolver.

Los diferentes estados que puede presentar la naturaleza: e_1, e_2, \dots, e_n .
 Las acciones o alternativas entre las que seleccionará el decisor: a_1, a_2, \dots, a_m .

Las consecuencias o resultados x_{ij} de la elección de la alternativa a_i cuando la naturaleza presenta el estado e_j .

Se supone, por simplicidad, la existencia de un número finito de estados y alternativas. El formato general de una tabla de decisión es la que se muestra a continuación:

	Estados de la Naturaleza				
	e_1	e_2	...	e_n	
Alternativas	a_1	x_{11}	x_{12}	...	x_{1n}
	a_2	x_{21}	x_{22}	..	x_{2n}

	a_m	x_{m1}	x_{m2}	..	x_{mn}

Tabla 1. Forma general de una tabla de decisión.

Ejemplo:

Un ama de casa acaba de echar cinco huevos en un tazón con la intención de hacer una tortilla. Dispone, además, de un sexto huevo del que no conoce su estado, aunque es de esperar que en caso de encontrarse en buen estado y no ser utilizado, se estropeará. Al ama de casa se le presentan tres posibles alternativas:

Romper el huevo dentro del tazón donde se encuentran los cinco anteriores.

Romperlo en otro tazón diferente.

Tirarlo directamente.

Dependiendo del estado del huevo, las consecuencias o resultados que pueden presentarse para cada posible alternativa se describen en la siguiente tabla:

Alternativas	Estado del 6º huevo	
	Bueno (e ₁)	Malo (e ₂)
Romperlo dentro del tazón (a ₁)	Tortilla de 6 huevos	5 huevos desperdiciados y no hay tortilla
Romperlo en otro tazón (a ₂)	Tortilla de 6 huevos y un tazón más que lavar	Tortilla de 5 huevos y un tazón más que lavar
Tirarlo (a ₃)	Tortilla de 5 huevos y un huevo bueno desperdiciado	Tortilla de 5 huevos

Tabla 2 . Consecuencias

2.1.1.2 MAPAS DE CONOCIMIENTO.

Son representaciones o expresiones gráficas que incluyen el conocimiento requerido para la solución de un problema ²

² Nicolás, M. C. K. V. (1995). Manual de Sistemas Expertos. Perú Lima, p.41.

En estos mapas el conocimiento se expresa a manera de relaciones secuenciales de precedencia, de jerarquía que se dan en los factores relevantes (se puede incluir todos los factores) que afectan el problema y su solución.

Estas relaciones por lo general son del tipo **causa-efecto**.

El tamaño de cada mapa de conocimiento depende del nivel de detalle que se quiera dar al análisis y solución del problema.

Existen otras métodos que se utilizan para adquirir conocimiento son:

- a) Listas.
- b) Árboles de decisión.
- c) Diagramas de Ishikawa.

A continuación se describe cada uno de ellos:

LISTAS:

Es una serie escrita de elementos relacionados. Son utilizadas para representar el conocimiento agrupados en objetos (elementos) categorizados o clasificados de acuerdo a una relación o prioridad (jerarquía).

Los objetos o elementos primero son divididos en grupos de clases o elementos similares, luego sus relaciones son mostradas por encadenamiento entre ellos.

ÁRBOLES DE DECISIÓN:

Es el encargado de integrar estrategias de búsqueda con relación de conocimiento.³

También se le conceptualiza como la representación de los espacios de búsqueda de soluciones a un problema, donde el tamaño de cada espacio de búsqueda esta dado por el número de nodos (ramas) en el árbol.

³ Nicolás, M. C. K. V. (1995). Manual de Sistemas Expertos. Perú Lima, p.44.

Están compuestos por nodos que representan las metas y por ramas que representan decisiones (conclusiones o resultados).

La graficación de estos árboles de decisión sigue el orden que se explica a continuación.

El nodo inicial se grafica a la izquierda, y los efectos parciales o finales a la derecha.

Un efecto parcial se constituye en una causa (rama de un árbol) hasta llegar al nodo final que da los resultados o conclusión al problema.

DIAGRAMAS DE ISHIKAWA.

Es un diagrama causa-efecto.

Las causas o factores causales son las condiciones que deben darse para lograr un efecto determinado. Una rama de las causas puede terminar en un efecto parcial (meta parcial), el cual se constituye en una de las causas del efecto total.⁴

Las ramas se ordenan en sentido contrario de las manecillas del reloj, donde los factores causales que representan el conocimiento superficial van primero y luego los factores causales que representan el conocimiento profundo.

Un efecto constituye una meta (solución, respuesta, recomendación, conclusión), de un determinado problema. A continuación se muestra una figura que indica el orden de un diagrama de Ishikawa.

⁴ Nicolás, M. C. K. V. (1995). Manual de Sistemas Expertos. Perú Lima, p.46.

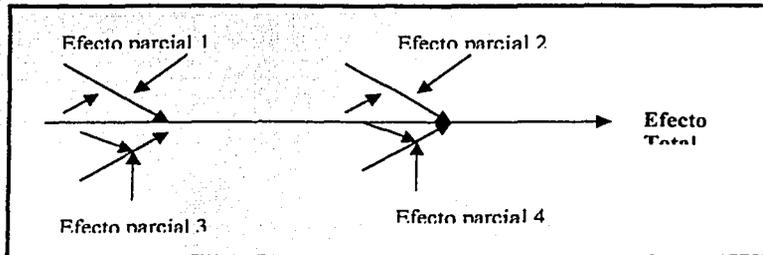


Figura 5. Diagrama de Ishikawa.

2.1.2 REPRESENTACIÓN DEL CONOCIMIENTO.

La alta capacidad y la eficiencia de un sistema inteligente en la solución de problemas del mundo real dependen de la cantidad y calidad de los conocimientos que contiene.

Debido a que el conocimiento es esencial e importante para el comportamiento inteligente, su representación a llegado a convertirse en objeto de estudio prioritario en IA.

La representación del conocimiento propone diferentes clases de estructuras y esquemas para organizar y almacenar el conocimiento en un sistema inteligente artificial. La elección de un esquema en particular dependerá del tipo de problema que se quiere resolver y de los métodos de inferencia que se utilizarán.

La representación del conocimiento se puede clasificar en dos clases: declarativa y procedimental.

Los esquemas declarativos hacen énfasis en la representación del conocimiento como acumulación de hechos estáticos junto con una información limitada que describe como se va a emplear el conocimiento. Los esquemas procedimentales enfatizan la representación el conocimiento en forma de reglas dinámicas que describen procedimientos para usar el conocimiento.

2.1.2.1 ESQUEMAS DE REPRESENTACIÓN DEL CONOCIMIENTO.

Los principales esquemas de representación del conocimiento son:

- o Lógica de predicados
- o Reglas de producción.
- o Redes semánticas.
- o Frames o slots.

La elección de la forma en la que se va a representar el conocimiento es determinante para el problema que se requiere enfrentar. Cada tipo de representación tiene ventajas y desventajas. Así, es necesario analizar cada una de ellas para hacer la mejor elección y adaptarla al problema a solucionar.

LÓGICA DE PREDICADOS.

La lógica de predicados es un lenguaje formal con sintaxis y gramática propias, capaz de valorar enunciados lógicos y extraer conclusiones para la creación de nuevos enunciados⁵. Su alfabeto consta de predicados, variables, funciones, constantes, conjunciones, cuantificadores.

Constantes.

Se utilizan para representar un elemento específico del dominio, este elemento puede ser cualquier objeto de interés. Y se especifica con letras mayúsculas.

Ejemplo: AZUL puede representar Un color.

Variables:

Se utiliza para representar un conjunto de elementos del dominio sin especificar elemento alguno en concreto. Se emplean letras minúsculas para identificarla.

Ej. Color significa que es cualquier color no especificado.

⁵ Nebendahl, D. (1988). Sistemas Expertos. Barcelona, Marcombo p.67.

Funciones.

Se utilizan para identificar un elemento del dominio. "Una función describe un elemento identificándolo como el resultado único de la aplicación de una transformación entre otros elementos del dominio. Una letra minúscula se emplea como el símbolo de una función y los argumentos pueden ser cualquier término válido (un término se define como una variable, constante o función)"⁶.

Ejemplo: $X(y)$ Invoca a una función que especifica el elemento único.

Predicado.

Los predicados se utilizan para representar relaciones dentro de un dominio e indican que un elemento se relaciona de alguna u otra forma.

Los predicados tienen valores de verdadero y falso. Son verdaderos cuando los elementos dados están relacionados de modo específico, y falso si no lo están.

"Los predicados (serie de letras mayúsculas) junto con términos que identifican los elementos relacionados, se emplean para conformar fórmulas atómicas (átomos), las cuales están en bloques básicos de construcción lógica de predicados"⁷.

Ejemplo: HOMBRE(LUIS) Luis es un hombre.

Interpretación.

De acuerdo a los componentes descritos anteriormente se pueden conformar fórmulas atómicas. Para conformar una fórmula se selecciona un conjunto de símbolos específicos y luego se establece una correspondencia, de manera que exista una relación entre cada uno de los símbolos seleccionados con los elementos del dominio asociado. Este conjunto de correspondencias establece la semántica que permite interpretar una fórmula.

Conjunción.

Se utiliza para expresar proposiciones compuestas.

Entre las conjunciones que se utilizan frecuentemente están:

\wedge And(Y), \vee or(O) inclusivo, implica (\rightarrow)

⁶ Rolston, D. W. (1992). Principios de Inteligencia Artificial y Sistemas Expertos. México, D.F., McGraw-Hill.

⁷ Rolston, D. W. (1992). Principios de Inteligencia Artificial y Sistemas Expertos. México, D.F., McGraw-Hill. p.36

Cuantificación.

Los cuantificadores que típicamente se utilizan en lógica de predicados son:

- El cuantificador universal; $\forall x$, indica que la fórmula bien formada, dentro de su alcance, es verdadera para todos los valores posibles de la variable que es cuantificada. Por ejemplo: $\forall x$ Establece que "para todo x , es verdad que . . . "
- El cuantificador existencial; $\exists x$, indica que la fórmula bien formada, dentro de su alcance, es verdadera para algún valor o valores dentro del dominio. Por ejemplo: $\exists x$ Establece que "existe un x , tal que . . . "

A continuación se dan algunos ejemplos de predicados cuantificados:

$\forall x$, [niño (X) \Rightarrow le_gusta (X, helados)].

$\forall y$, [mamífero (Y) \Rightarrow nace (Y, vivo)].

$\exists x$, [cartero(Z) \wedge mordió (boby, Z)].

En lógica de predicados, existen tres métodos básicos de razonamiento: deductivo, abductivo e inductivo.

Deducción.- Es el razonamiento a partir de un principio conocido hacia un desconocido; de lo general, a lo específico, o de la premisa a la conclusión lógica. La deducción realiza inferencias lógicamente correctas. Esto significa que la deducción a partir de premisas verdaderas, garantiza el resultado de conclusiones también verdaderas.

La deducción es el método más ampliamente comprendido, aceptado y reconocido de los tres mencionados. Es la base tanto de la lógica proposicional, como de la lógica de predicados. A manera de ejemplo, el método deductivo, se puede expresar, utilizando lógica de predicados, como sigue:

$\exists A, \exists B, \exists C$, [mayor (A, B) \wedge mayor (B, C) \rightarrow mayor (A, C)]

Abducción.- Es un método de razonamiento comúnmente utilizado para generar explicaciones. A diferencia de la deducción, la abducción no garantiza que se puedan lograr conclusiones verdaderas, por lo tanto no es un método sólido de inferencia. La forma que tiene la abducción es la siguiente:

Si la sentencia $(A \wedge B)$ es verdadera y B es verdadera, entonces A es posiblemente verdadera.

En abducción, se empieza por una conclusión y se procede a derivar las condiciones que podrían hacer a esta conclusión válida. En otras palabras, se trata de encontrar una *explicación* para la conclusión.

Inducción.- Se define como el razonamiento a partir de hechos particulares o casos individuales, para llegar a una conclusión general. El método inductivo es la base de la investigación científica. La forma más común del método inductivo es la siguiente:

Si se conoce que $P(a)$, $P(b)$,, $P(n)$ son verdaderos, entonces se puede concluir que $\exists X$, $P(X)$ es también verdadero.

La inducción es una forma de inferencia muy importante ya que el aprendizaje, la adquisición de conocimiento y el descubrimiento están basados en ella. Al igual que la abducción, la inducción no es un método sólido de inferencia.

El razonamiento deductivo es una forma **monotónica** de razonar que produce argumentos que preservan la verdad. En un sistema monotónico todos los axiomas utilizados se conocen como verdaderos por sus propios méritos, o pueden ser derivados de otros hechos conocidos como verdaderos. Los axiomas no pueden cambiar, ya que una vez que se los conoce como verdaderos, siempre permanecen así y no pueden ser modificados o retractados. Esto significa que en el razonamiento monotónico el conjunto de axiomas continuamente crece en tamaño.

Otro aspecto del razonamiento monotónico es que si más de una inferencia lógica puede ser hecha a un tiempo específico y una de ellas se realiza, las inferencias que quedan serán todavía aplicables después que dicha inferencia haya sido hecha.

REGLAS DE PRODUCCIÓN.

Conjunto de cláusulas de la lógica de predicados. Ésta es probablemente la forma más popular de representación del conocimiento utilizada en los Sistemas Expertos.

Una regla de producción es una estructura de tal forma:

SI: antecedente

ENTONCES: consecuente

Donde el antecedente está compuesto por una o varias proposiciones, las cuales se combinan para formar proposiciones.

Y el consecuente es una regla de producción que está compuesto por una proposición simple.

En el antecedente se utilizan generalmente los conectores siguientes:

Conjunción(AND), Disyunción inclusiva(OR), Negación(NOT).

Donde la conjunción trata de que si dos proposiciones son, verdaderas en el antecedente, el consecuente es verdadero.

La Disyunción inclusiva trata de que si alguna de las dos proposiciones es verdadera en el antecedente, el consecuente es verdadero.

La negación trata si la proposición es verdadera en el antecedente el consecuente es Falso.

REDES SEMÁNTICAS.

“Representación gráfica de las relaciones entre los elementos de un dominio. Los componentes básicos de una red semántica son los nodos y los enlaces. Los nodos se emplean para representar elementos del dominio. Se muestran gráficamente como rectángulos y son rotulados con los nombres de los elementos representados. Los enlaces representan las relaciones entre los elementos. Un enlace se muestra como un vector desde un nodo a otro; se rotula con el nombre de las relaciones representadas”.⁸

Las redes semánticas resultan ser eficientes esquemas de representación del conocimiento ya que ofrecen una visión general de las relaciones y dependencias de un área de conocimiento. Además de

⁸ Rolston, D. W. (1992). *Principios de Inteligencia Artificial y Sistemas Expertos*. México, D.F., McGraw-Hill p.45.

que una red semántica bien estructurada facilita el entendimiento del proceso del conocimiento.

FRAMES O MARCOS.

Un marco, introducido por primera vez por Minsky en 1945, es una estructura para organizar el conocimiento en forma jerárquica, puesto que permite heredar características. A su vez, los frames de más bajo nivel(hijos), heredan características de los frames de más alto nivel(padres), los cuales contienen conocimiento más general. Cuando ya no tienen descendencia, los frames hijos se les llaman instancias.

Estructura de un frame.

Esta constituido principalmente por SLOTS(ranuras); cada slot describe las características de un objeto, y puede incluir conocimiento

declarativo (Por ejemplo el color de un caballo es café), o procedural (donde se activa una regla dependiendo del antecedente).

Un atributo representa a un slot, un objeto puede tener varios slots.

Para describir completamente a un objeto es necesario conocer todos sus atributos relevantes y para cada atributo se debe establecer un valor.

Para establecer los valores de los atributos de un objeto, se siguen alternativas como: Facetas que pueden ser declarativas o procedurales, las cuales describen al conocimiento de cada atributo.

Las facetas por medio de las cuales podemos obtener el valor del atributo del frame son:

- Valor. Es el encargado de describir al atributo.
- Default. Es utilizado por si un slot esta vacío.
- Rango. Es la clase de información que aparece en un slot.
- IF-ADDED. "Contiene conocimiento procedural o de enlace. Especifica una acción que debe ser tomada cuando un valor en el slot es adicionado (almacenado) o modificado(removido). A este procedimiento se le llama DEMONS".⁹

⁹ Nicolás, M. C. K. V. (1995). Manual de Sistemas Expertos. Perú Lima, p.58.

- **IF-NEEDED.** Se usa cuando el valor del slot no está dado (slot vacío), además sus datos son extraídos de base de datos, datos dados por el usuario.

En esta sección se revisaron los métodos de adquisición y los principales esquemas de representación del conocimiento, todo esto nos da el marco que nos ayudará a construir aplicaciones y sistemas orientados al aprendizaje basados en conocimiento.

Como observamos, gracias a la Ingeniería del conocimiento podemos construir sistemas para administrar productiva y estratégicamente el conocimiento de un individuo. De esta manera, haciendo uso de las técnicas y herramientas de la ingeniería del conocimiento podemos diseñar, desarrollar, producir y administrar los ambientes de aprendizaje innovadores que demandan actualmente nuestras instituciones educativas.

2.2 INTRODUCCIÓN A LOS SISTEMAS EXPERTOS.

Los sistemas expertos se pueden considerar sistemas capaces de actuar como lo haría un experto humano en la resolución de un problema concreto.

Existen muchos tipos de sistemas expertos: de diagnóstico médico, para el análisis de imágenes de satélite, clasificadores, controladores, asesores, agentes económicos, tutores, etc. Día con día va en aumento su posible campo de trabajo.

Los sistemas expertos forman parte de un firme y verdadero avance en inteligencia artificial: pueden incorporar miles de reglas (para una persona sería una experiencia casi "traumática" el realizar una búsqueda de reglas posibles para la solución de un problema y concordar éstas con las posibles consecuencias) y realizar muy eficientemente la tarea de análisis de árboles: mientras que el usuario responde a las preguntas formuladas por el sistema experto, éste busca recorriendo las ramas más interesantes del árbol, hasta dar con la respuesta afín al problema, o en su falta, la más parecida a esta. Un sistema experto tiene además la ventaja de proporcionar gran flexibilidad a la hora de incorporar nuevos conocimientos, pues sólo tenemos que introducir la nueva regla que deseamos hacer constar y...ya está, sin necesidad de cambiar el funcionamiento propio del programa. Además son "autoexplicativos", al contrario que en los

- **IF-NEEDED.** Se usa cuando el valor del slot no está dado (slot vacío), además sus datos son extraídos de base de datos, datos dados por el usuario.

En esta sección se revisaron los métodos de adquisición y los principales esquemas de representación del conocimiento, todo esto nos da el marco que nos ayudará a construir aplicaciones y sistemas orientados al aprendizaje basados en conocimiento.

Como observamos, gracias a la Ingeniería del conocimiento podemos construir sistemas para administrar productiva y estratégicamente el conocimiento de un individuo. De esta manera, haciendo uso de las técnicas y herramientas de la ingeniería del conocimiento podemos diseñar, desarrollar, producir y administrar los ambientes de aprendizaje innovadores que demandan actualmente nuestras instituciones educativas.

2.2 INTRODUCCIÓN A LOS SISTEMAS EXPERTOS.

Los sistemas expertos se pueden considerar sistemas capaces de actuar como lo haría un experto humano en la resolución de un problema concreto.

Existen muchos tipos de sistemas expertos: de diagnóstico médico, para el análisis de imágenes de satélite, clasificadores, controladores, asesores, agentes económicos, tutores, etc. Día con día va en aumento su posible campo de trabajo.

Los sistemas expertos forman parte de un firme y verdadero avance en inteligencia artificial: pueden incorporar miles de reglas (para una persona sería una experiencia casi "traumática" el realizar una búsqueda de reglas posibles para la solución de un problema y concordar éstas con las posibles consecuencias) y realizar muy eficientemente la tarea de análisis de árboles: mientras que el usuario responde a las preguntas formuladas por el sistema experto, éste busca recorriendo las ramas más interesantes del árbol, hasta dar con la respuesta afín al problema, o en su falta, la más parecida a esta. Un sistema experto tiene además la ventaja de proporcionar gran flexibilidad a la hora de incorporar nuevos conocimientos, pues sólo tenemos que introducir la nueva regla que deseamos hacer constar y...ya está, sin necesidad de cambiar el funcionamiento propio del programa. Además son "autoexplicativos", al contrario que en los

programas convencionales, en los que el conocimiento como tal, está encriptado junto al propio programa en forma de lenguaje de computadora. Los expertos de I.A. dicen que los sistemas expertos tienen un conocimiento declarativo, mientras que los demás programas encierran conocimientos procedurales.

Un sistema experto puede definirse como un sistema basado en conocimientos, imitando el pensamiento de un experto para resolver problemas de un campo particular de aplicación. Una de sus características principales es que están basados en reglas, es decir, contienen conocimientos predefinidos en forma de sentencias:

Si (Condición)
Entonces (Acción)

que se utilizan para tomar las decisiones.

A lo largo de esta sección vamos a profundizar más en su estructura.

2.2.1 HISTORIA.

1965 DENDRAL, el 1er Sistema Experto

En 1965 Feigenbaum entra a formar parte del departamento de informática de Stanford. Allí conoció a Joshua Lederberg, el cual quería averiguar cual era la estructura de las moléculas orgánicas completas. Si se quiere estudiar un compuesto químico, lo primero que se tiene, que hacer es averiguar su fórmula química. Este análisis nos dice cuantos átomos de cada tipo hay en las moléculas. Las propiedades de una molécula completa dependen de los enlaces químicos dentro de ésta, y de cómo están colocados unos átomos respecto a otros. Todas estas propiedades constituyen la estructura molecular. Hacer un mapa de la molécula no resulta tan sencillo. El método habitual de análisis consiste en dividir el problema en partes más pequeñas. Los químicos rompen las moléculas que estudian en fragmentos ionizados, y procesan estos fragmentos en un "espectrómetro de masa", que nos da la proporción carga/masa de los fragmentos. El conjunto de estas piezas de información forma una imagen completa para reconstruir la molécula original. El problema de Lederberg era que había demasiadas combinaciones. Cada fragmento podía corresponder a varias subestructuras, pero solo una estructura molecular global se ajustaba a todas las restricciones del problema. El descubrimiento de esa

estructura global exigía buscar en un árbol las posibilidades. Fue por esta razón por la que pusieron al programa el nombre de DENDRAL, que significa en griego "árbol". Antes de DENDRAL los químicos sólo tenían una forma de resolver el problema, y esta era tomar unas hipótesis relevantes como soluciones posibles, y someterlas a prueba comparándolas con los datos.

La realización DENDRAL duró más de diez años (desde 1965), y se le considera el primer sistema experto.

1972 MYCIN

MYCIN es un sistema experto para el diagnóstico de enfermedades infecciosas. A partir de los resultados del análisis de sangre, cultivos bacterianos y demás datos, el programa era capaz de determinar, o por lo menos sugerir, el microorganismo que estaba causando la infección. Después de llegar a una conclusión, MYCIN prescribía una medicación que se adaptaba perfectamente a las características del sujeto, tales como el

peso corporal de éste. Después del reconocido éxito que tuvo MYCIN entre la comunidad científica de I.A. e investigadores de medicina, los autores del programa pusieron a disposición de otros investigadores una versión de MYCIN que contenía el motor "inferencial" de éste para que fueran los propios usuarios los que insertaran al programa la información del tema a tratar; se podría decir que esta versión "adaptada" del original MYCIN, fue el primer entorno de desarrollo de sistemas expertos.

1973 TEIRESIAS

Teiresias: profeta ciego del Edipo rey de Sófocles, puede penetrar en acontecimientos ocultos al común de los mortales, y posee una forma superior de conocimiento. El sistema experto llamado TEIRESIAS no era totalmente capaz de estas "proezas". El cometido de este Sistema Experto (S.E.) no era otro que el de servir de interprete entre MYCIN y los especialistas que lo manejaban, a la hora introducir nuevos conocimientos en su base de datos. El especialista debía utilizar MYCIN de una forma normal, y cuando éste cometiera un error en un diagnóstico (hecho producido por la falta o fallo de información en el árbol de conocimientos) TEIRESIAS corregiría dicho fallo destruyendo la regla si ésta era falsa o ampliándola si es eso lo que se necesitaba.

1979 XCON, primer programa que sale del laboratorio

A finales de los sesenta la Universidad de Canegie Mellon preparaba el principio de la inserción de la I.A. en el mundo real. El programa empezó llamándose R1, nombre que provenía de un mal chiste del creador del proyecto [aunque estoy en disposición de afirmar que este chiste es lo peor de lo peor, ahí va: "Yo siempre había querido ser un ingeniero del conocimiento, y ahora ya lo era", en inglés "...and now I R1" lo que suena en inglés como "...and now I're one"], pero la Digital Equipament Corporation, que era el usuario final de este programa, expresó su preferencia por otro nombre con más sentido, como el que al final quedó elegido: XCON (eXpert CONfigurer) [Configurador Experto]. Cuando fue exigida la ayuda de McDermott, la DEC iba a lanzar al mercado una nueva serie de computadoras, las llamadas VAX. Dado que todas las computadoras tenían configuraciones distintas entre sí, la VAX estaba previendo el enorme cuello de botella que se iba a formar, cuando (como ya había pasado anteriormente con otra serie de computadoras) las computadoras tuviesen fallos de configuración y hubiese que arreglar este problema uno por uno, con el consiguiente gasto de tiempo y dinero que eso suponía. Fue entonces cuando, alarmados, los directivos de la DEC pensaron en poner en marcha una solución contratando a John McDermott. El cometido del XCON sería simple: configurar todas las computadoras que saliesen de la DEC. El informe de viabilidad de McDermott expuso resultados prometedores, y en diciembre de 1978 se empezó a trabajar en el proyecto.

En abril de 1979 el equipo de investigación que lo había diseñado (con McDermott a la cabeza), pensó que ya estaba preparado para salir y "conocer el mundo". Fue entonces cuando se hizo una prueba real, esperando resolver positivamente un 95% de las configuraciones. Este porcentaje tal alto anhelado por McDermott y DEC se quedó en un 20% al ser contrastado con la realidad; XCON volvió al laboratorio, donde fue revisado y a finales de ese mismo año (1979) funcionó con resultados positivos en la DEC.

En 1980 XCON se instauró totalmente en DEC. Y en 1984, el XCON había crecido hasta multiplicarse por diez. En 1986 la compañía había invertido más de cincuenta años/hombre en el programa, pero se estaba recuperando con creces de su inversión al ahorrarse cuarenta millones de dólares al año.

80-85 LA REVOLUCIÓN DE LOS SISTEMAS EXPERTOS

XCON significó el disparo de salida para los sistemas expertos, convenciendo a las más importantes empresas a invertir en programas de este tipo. Incluso para los más conservadores y los radicales Anti-I.A., algo se hacia evidente, mientras que los costos de desarrollo de los sistemas expertos se hacian cada vez más bajos, el salario de los expertos humanos se hacia más alto. La formación de un futuro experto costaba mucho dinero y años de esfuerzo, mientras que una vez creado un sistema experto podia ser copiado y distribuido tantas veces como fuese necesario. Además, el sistema experto no se cansa, no necesita dormir, no se distrae, no se va a la competencia, no se enferma, ni se jubila, ni pide aumento de sueldo...

Más tarde se crearon otros S.E. , como el DELTA, de la General Electric Company, para la reparación de locomotoras diesel y eléctricas, o el "Aldo en Disco", dedicado a la reparación de calderas hidroestáticas giratorias usadas para la eliminación de bacterias.

Muchas grandes empresas invirtieron en el campo de la I.A. , creando grupos especializados en esta, para que desarrollaran aplicaciones para la empresa. Empresas como DEC, Xerox, Schlumberger-Doll y Texas

Instruments gastaron un total (mas 150 empresas más) de mil millones de dólares en el desarrollo de la I.A.

A raíz de todos estos adelantos e inversiones puestos en marcha desde la calle. (es decir, fuera de los laboratorios de las universidades, como era costumbre hasta la fecha) Se provocó que la sociedad empezase a oír cosas sobre la tal "Inteligencia Artificial", que llegaba a ocupar programas de televisión, y artículos en revistas científicas de todo tipo. Edward Feigenbaum fundó la Teknowledge Inc, pero esta empresa no fue la única, sino que le acompañaron muchas más, como: Carnegie Group, Symbolics, Lisp Machines Inc, Thinking Machines Corporation, Cognitive Systems Inc, formando una inversión total de 300 millones de dólares. Los productos más importantes que creaban estas nuevas compañías eran las "máquinas Lisp", computadoras que ejecutaban programas en LISP con la misma rapidez de una computadora central. Otro producto en el que se invirtió gran esfuerzo y dinero, fueron las "herramientas de desarrollo de sistemas expertos", también llamados "shells" (conchas).

Este crecimiento, "desmesurado" a los ojos de los investigadores de I.A. más veteranos, provocó un colapso en la comunidad científica de I.A. haciendo que el primer congreso de la AAAI (Asociación Americana de Inteligencia Artificial) que se realizó en Stanford, en 1985, reuniese a más de mil investigadores. En la International Joint Conference on Artificial Intelligence (IJCAI) se llegó a las seis mil personas. Todo estaba cambiado para esos investigadores de I.A. que estaban acostumbrados a las "pequeñas" reuniones, casi familiares. Estas conferencias eran entonces (1985) el centro de reunión de empresarios, reporteros, científicos de otras ramas con curiosidad.

Pero la demanda por parte de las empresas privadas de personal era mucho mayor a la oferta, esto hacia que los sueldos aumentasen enormemente y obligaban a los estudiantes a dejar la universidad cuanto antes. El MIT tenía en 1975 unos cuarenta investigadores y pasó a tener doscientos a finales de los 80.

1987 XCON empezó a cruzir sobre su propio peso

Las constantes mejoras y cambios en los productos de DEC habían hecho que XCON se actualizara muy rápidamente, siendo objeto de la introducción de más y más reglas de funcionamiento, hasta llegar a tener más de diez mil reglas. A partir de ese momento XCON no era mejor, sino únicamente más grande, y DEC tuvo que gastar 2 millones de dólares al año para actualizar la base de conocimiento. Algo parecido ocurrió con el DELTA de la General Electric. Hasta esa fecha nadie se había dado cuenta de que los sistemas expertos necesitan mantenimiento, así como lo complejo y costoso que puede resultar. Este punto constituye hoy en día una de las actividades principales de la industria de los sistemas expertos.

Durante la historia de los sistemas expertos se han desarrollado sistemas de **instrucción** (Sistema Experto para formación), los cuales realizan un seguimiento del proceso de aprendizaje de un estudiante. El sistema detecta errores de los estudiantes e identifica el remedio adecuado, es decir, desarrolla un plan de enseñanza para facilitar el proceso de aprendizaje y la corrección de errores. Además de DELTA, existen numerosos sistemas de este tipo; STEAMER(1983), por ejemplo, se creó para enseñar a los oficiales de la armada los problemas de funcionamiento de una planta de propulsión a vapor, como las que impulsan a ciertos barcos. Este era el problema de formación más importante que existía, dada la complejidad de los

sistemas. El objetivo es dar al estudiante una concepción global de lo que pasa en la planta en cualquier momento, con la ventaja de que además el modelo de presentación es gráfico (utilizando Interlisp). Con un objetivo similar al de STEAMER, Construcciones Aeronáuticas S. A. (CASA) desarrolló el Proyecto Eolo CN-235. En este caso, el problema está en el hecho de que pilotar un avión que cuesta cientos de millones de pesetas es un asunto muy serio a la vez que peligroso, lo que exige mucho tiempo de entrenamiento, tanto para pilotos como mecánicos, suponiendo para las compañías aéreas un gran problema, dado el elevado coste de los cursos y la escasez de instructores. El proyecto surgió de la voluntad de Construcciones Aeronáuticas S. A. de ofrecer un curso específico para pilotos y técnicos de mantenimiento, a todos los compradores del avión CN- 235. Eolo CN-235 es un sistema de enseñanza interactivo que integra gráficos, texto y vídeo.

Otro sistema de este tipo, aunque en este caso orientado a medicina, es GUIDON, pensado para que lo utilicen las Facultades de Medicina para formar a los médicos en la realización de consultas. GUIDON viene a ser una reorganización de MYCIN con intenciones educativas, por esto, tiene la ventaja adicional de disponer de toda la base de conocimientos de MYCIN además de la experiencia acumulada, por consiguiente, puede recuperar como ejemplo cualquier caso que MYCIN haya tratado.

2.2.2 DEFINICIÓN.

Un sistema experto es un programa interactivo de computadora que contiene la experiencia, conocimiento y habilidad propios de una persona, o grupos de personas, en un área particular del conocimiento humano. Es

creado con el fin de resolver problemas específicos en esa área de conocimiento de manera inteligente y satisfactoria, recreando la forma en que los humanos toman decisiones cuando se presenta la misma situación.

Los sistemas expertos constituyen una instrumentación de la I.A. apasionante y muy útil. Son sistemas que acumulan saber perfectamente estructurado, de tal manera que es posible obtenerlo gradualmente según las situaciones. Un sistema experto no es una biblioteca que aporta información, sino un consejero o especialista en una materia; de ahí que aporte saber, consejo. Así, se trata de un sofisticado programa de computadora que posee en su memoria y en su estructura una amplia cantidad de saber y, sobre todo, de estrategias para depurarlo y ofrecerlo según los requerimientos. Ello convierte al

sistema en un especialista en la materia para lo que está programado, y puede utilizarse como apoyo o elemento de consulta para investigadores, médicos, abogados, geólogos, y otros profesionales. En la actualidad existe un gran número de sistemas expertos repartidos entre los campos más activos de la investigación y del ámbito laboral en general.

Una de las características más importantes de los sistemas expertos es que están basados en reglas. Esto significa que contiene un juego predefinido de conocimientos que se utiliza para sus decisiones. El sistema usa las reglas y las inferencias que se le han indicado para producir resultados.

Las ventajas que supone el uso de un Sistema Experto han motivado el enorme crecimiento de este campo. Algunas de estas ventajas se exponen en la continuación:

1. Con la ayuda de un Sistema Experto, personas con poca experiencia pueden resolver problemas que requieren un "conocimiento especializado". De esta forma, se incrementa el número de personas con acceso a un conocimiento experto.

2. Los Sistemas Expertos pueden obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos. Por tanto, los

Sistemas Expertos son de gran valor en las situaciones donde el tiempo juega un papel crítico (control de la refrigeración de una central nuclear, etc).

3. Los Sistemas Expertos razonan pero en base a un conocimiento adquirido y no tienen sitio para la subjetividad, es decir, siempre obtienen la misma respuesta a partir de los mismos datos.

4. En algunos casos, la complejidad de un problema hace que un experto humano no pueda obtener una conclusión. Debido a la capacidad de las computadoras de procesar una gran cantidad de información, y de realizar un gran número de operaciones en poco tiempo, los Sistemas Expertos pueden obtener conclusiones realistas en situaciones donde los expertos humanos no pueden.

5. El uso de Sistemas Expertos es especialmente recomendado en las siguientes situaciones:

* Cuando los expertos humanos en una determinada materia son escasos, los Sistemas Expertos pueden recoger y difundir su conocimiento.

* En situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas.

* Cuando sea muy elevado el volumen de datos que ha de considerarse para obtener una conclusión.

* En situaciones deterministas, en las que las conclusiones se obtienen aplicando un conjunto de reglas dado

En esta sección hemos visto que se han desarrollado una gran cantidad de sistemas expertos en los diversos campos de trabajo, incluyendo la educación, y han resultado ser de gran utilidad para la resolución de problemas que se presentan día a día. Dadas las ventajas, mencionadas en esta sección, el desarrollo de los sistemas expertos va en aumento, y los campos en el cual se pueden aplicar no se encuentra limitados, por ello podemos adecuar nuestras necesidades para la realización de un sistema experto que nos proporcione la ayuda indispensable de manera eficiente.

2.2.3 ESTRUCTURA DE UN SISTEMA EXPERTO.

Un sistema experto se divide en:

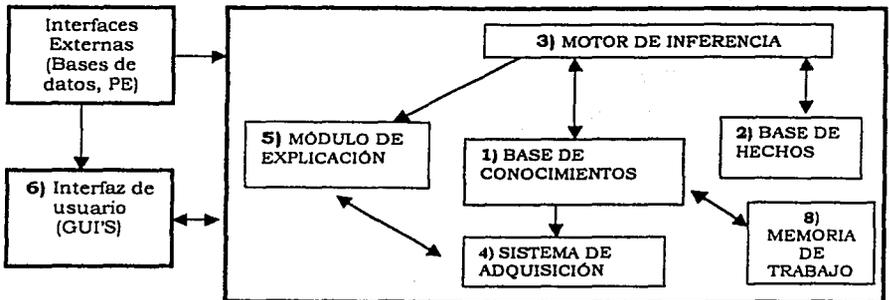


Figura 6. Estructura general de un Sistema Experto.

Estos módulos se explican a continuación:

1) BASE DE CONOCIMIENTO.

La base de conocimientos aloja todos los hechos, las reglas y los procedimientos específicos relativos al campo del saber deseado.

En la base de conocimientos la información entra tal como llega, ya que el orden no influye en los resultados obtenidos. Sucede así porque cada elemento de conocimiento es comprensible por sí mismo, tomado de forma aislada y, por lo tanto, no es necesario referirse al contexto en el cual está inserto.

La información puede ser representada mediante reglas de producción. Las reglas de producción constituyen el método más utilizado para construir bases de conocimientos. Llamadas también implicaciones lógicas. Su estructura es la siguiente: para unas ciertas causas, unos efectos; o, para determinadas condiciones, ciertas consecuencias. Esto se define a menudo como programación orientada a las reglas.

La fuerza de un SE yace en el conocimiento específico del dominio del problema. Casi todos los sistemas existentes basados en reglas

contienen una gran cantidad de ellas y generalmente se obtienen de entrevistas con expertos durante un tiempo largo. En cualquier SE, las reglas se conectan una a otra por ligas de asociación para formar redes de reglas. Generalmente, un experto tiene muchas reglas de criterio o empíricas, por lo que el soporte suele ser incompleto para la evidencia que se dispone. En tales casos, una técnica es la de vincular valores numéricos a cada regla para indicar el grado de certidumbre que ésta tiene. En la operación de SE los valores de certidumbre se combinan unos a otros a través de la red, considerando también, la certidumbre de los datos de entrada del problema, para llegar finalmente a un valor de certidumbre de la solución final.

La cantidad y la calidad de los conocimientos contenidos en la base de conocimientos determinan el buen funcionamiento de un SE. Una base de conocimientos debe contener las siguientes características:

- a) Exhaustividad.
- b) Consistente.

Donde la Exhaustividad significa "completitud", es decir, que todas las reglas y los hechos necesarios para solucionar cualquier problema del dominio, se encuentren incorporados en la base de conocimientos.

La consistencia se refiere a que dicha base este libre de reglas contradictorias, redundantes o innecesarias.

2) BASE DE HECHOS.

Se denominan hechos a la información que es variable de una a otra resolución. Así una base de hechos alberga los datos propios correspondientes al problema particular que se desea tratar con la ayuda del sistema. Así mismo almacena todos los resultados intermedios, permitiendo conservar un registro de los razonamientos llevados a cabo. Puede emplearse para explicar el origen de las informaciones deducidas por el sistema en el transcurso de una sesión de trabajo, o para llevar a cabo la descripción del comportamiento del propio sistema experto. Al principio del periodo de trabajo, la base de hechos dispone únicamente de los datos que el usuario del sistema introduce, pero a medida que va actuando el motor de inferencias, se agregan las cadenas de inducciones y deducciones que el sistema forma al aplicar las reglas para obtener las conclusiones buscadas.

En un Sistema Experto existirá un hecho sólo cuando esté este contenido en la base de conocimientos.

3) MOTOR DE INFERENCIA.

El motor de inferencia, es el sistema de control del S.E. que, mediante el empleo de los conocimientos, puede resolver el problema que está especificado. Lo resuelve gracias a los datos que contiene la base de hechos y de conocimientos del Sistema experto. La tarea que lleva a cabo el motor de inferencias es la de seleccionar, interpretar, validar, activar y aplicar el conocimiento de la Base de Conocimiento sobre la Base de Hechos, con el fin de obtener finalmente la solución correspondiente al problema planteado.

Así, un sistema experto se compone de dos tipos de elementos; 1) los propios del campo de los expertos relacionados con el problema concreto, (es decir, la base de conocimientos y la base de hechos) y 2) el que se puede aplicar de forma general a una gran variedad de problemas de diversos campos (como el caso del motor de inferencias). Sin embargo, el motor de inferencia no es un mecanismo universal de deducción, ya que hay dos tipos: los que emplean el razonamiento aproximado (para el cual, el resultado puede ser erróneo) y aquellos que

emplean un tipo de razonamiento capaz de obtener un resultado (si llegan a él), con toda seguridad, verdadero.

El motor de inferencia es un programa que controla el proceso de razonamiento que seguirá el sistema experto. Utilizando los datos que se le suministran, recorre la base de conocimientos para alcanzar una solución.

Toda la información aportada por el usuario es almacenada como hechos en una memoria de trabajo. Cuando se realiza el proceso de inferencia, las reglas almacenadas en la base de conocimientos son comparadas con el contenido de la memoria de trabajo.

La estrategia de control puede ser de encadenamiento hacia adelante o de encadenamiento hacia atrás. En el primer caso se comienza con los hechos disponibles en la base de conocimientos, y se buscan reglas que satisfagan esos datos, es decir, reglas que verifiquen las premisas. Normalmente, el sistema sigue los siguientes pasos:

1. Evaluar las condiciones de todas las reglas respecto a la base de hechos, identificando el conjunto de reglas que se pueden aplicar (aquellas que satisfacen su parte condición).
2. Si no se puede aplicar ninguna regla, se termina sin éxito; en caso contrario se elige cualquiera de las reglas aplicables y se ejecuta su parte acción (esto último genera nuevos hechos que se añaden a la base de hechos).
3. Si se llega al objetivo, se ha resuelto el problema; en caso contrario, se vuelve al paso 1.

Cuando se aplica una regla sobre algún hecho cualesquiera, se dice que una regla se dispara. El disparo de una regla provoca la inserción de un nuevo hecho en la base de conocimiento.

A este enfoque se le llama también "guiado por datos", pues es el estado de la base de hechos el que identifica las reglas que se pueden aplicar. Cuando se utiliza este método, el usuario comienza introduciendo datos del problema en la base de conocimientos del sistema.

Al encadenamiento hacia atrás se le suele llamar guiado por objetivos o hipótesis, ya que el sistema comenzará por un objetivo preciso (parte conclusión de las reglas), y operará retrocediendo para ver cómo se deduce ese objetivo partiendo de los datos. Esto se produce

directamente o a través de conclusiones intermedias o subobjetivos. Como se puede observar, lo que se intenta es probar una hipótesis a partir de los hechos contenidos en la base de hechos y de los obtenidos en el proceso de inferencia.

Este enfoque tiene la ventaja de que el sistema va a considerar únicamente las reglas que interesan al problema en cuestión. El usuario comenzará declarando una expresión E y el objetivo del sistema será establecer la verdad de esa expresión.

Para ello se siguen los siguientes pasos:

1. Obtener las reglas relevantes, buscando la expresión E en la parte conclusión (éstas serán las que puedan establecer la verdad de E).
2. Si no se encuentran reglas para aplicar, entonces no se tienen datos suficientes para resolver el problema; se termina sin éxito o se piden al usuario más datos.
3. Si hay reglas para aplicar, se elige una y se verifica su parte condición C con respecto a la base de datos.
4. Si C es verdadera en la base de datos, se establece la veracidad de la expresión E y se resuelve el problema.
5. Si C es falsa, se descarta la regla en curso y se selecciona otra regla.
6. Si C es desconocida en la base de datos (es decir, no es verdadera ni falsa), se le considera como subobjetivo y se vuelve al paso 1 (C será ahora la expresión E).

Existen también enfoques mixtos en los que se combinan los métodos guiados por datos con los guiados por objetivos.

4) MÓDULO DE ADQUISICIÓN.

El módulo de adquisición del conocimiento permite que se puedan añadir, eliminar o modificar elementos de conocimiento (en la mayoría de los casos reglas) en el sistema experto. Si el entorno es dinámico, este módulo es muy necesario, puesto que el sistema funcionará correctamente sólo si se mantiene actualizado su conocimiento. El módulo de adquisición permite efectuar ese mantenimiento, anotando en la base de conocimientos los cambios que se producen.

5) MÓDULO DE EXPLICACIONES.

La mayoría de los sistemas expertos contienen un módulo de explicación, diseñado para aclarar al usuario la línea de razonamiento seguida en el proceso de inferencia. Si el usuario pregunta al sistema cómo ha alcanzado una conclusión, éste le presentará la secuencia completa de reglas usadas. Esta posibilidad de explicación es especialmente valiosa cuando se tiene la necesidad de tomar decisiones importantes amparándose en el consejo del sistema experto. Además, de esta forma, y con el tiempo suficiente, los usuarios pueden convertirse en especialistas en la materia, al asimilar el proceso de razonamiento seguido por el sistema. El subsistema de explicación también puede usarse para depurar el sistema experto durante su desarrollo.

6) INTERFAZ DE USUARIO.

La interfaz de usuario permite que el usuario pueda describir el problema al sistema experto. Interpreta sus preguntas, los comandos y la información ofrecida. A la inversa, formula la información generada por el sistema incluyendo respuestas a las preguntas, explicaciones y justificaciones. Es decir, posibilita que la respuesta proporcionada por el sistema sea inteligible para el interesado. También puede solicitar más información si le es necesaria al sistema experto. En algunos sistemas se utilizan técnicas de tratamiento del lenguaje natural para mejorar la comunicación entre el usuario y el sistema experto.

Debemos tener en cuenta que la interfaz de usuario debe cumplir con los siguientes requisitos.

- 1) Debe evitarse en lo posible la entrada de datos errónea.
- 2) Los resultados deben ser siempre de forma clara para evitar confusiones en el usuario.
- 3) Las preguntas y explicaciones deben ser comprensibles.
- 4) El aprendizaje del manejo debe ser rápido.

7) MÓDULO DE APRENDIZAJE.

La función de este módulo es contribuir en la construcción y refinamiento de la base de conocimientos. En la actualidad, son pocos los sistemas expertos que tienen este módulo.

8) MEMORIA DE TRABAJO.

La memoria de trabajo contiene toda la información que el usuario le proporciona al sistema experto en forma de hechos.

Durante una consulta al sistema experto, el usuario introduce la información al sistema, dicha información es almacenada en la memoria de trabajo para posteriormente ser comparada con el conocimiento almacenado en la base de conocimientos para inferir nuevos hechos. El sistema introduce estos nuevos hechos dentro de la memoria de trabajo y el proceso de comparación continua.

Eventualmente el sistema llega a algunas conclusiones que también son introducidas dentro de la memoria de trabajo.

La memoria de trabajo contiene toda la información acerca del problema que esta siendo suministrado por el usuario o inferenciado por el sistema.

Cuando el proceso de solución a un problema ha concluido, el contenido de la memoria de trabajo es eliminado.

Como ya mencionamos, es evidente que la teoría de los sistemas expertos ha sido aplicado en diversos campos incluso en la educación.

El SE es una parte de las tecnologías para la educación, capaz de transformar contundente y positivamente los procesos de enseñanza-aprendizaje, aportando nuevos y mejores caminos para el desarrollo educativo. Por eso un sistema experto puede ser elaborado con fines didácticos. El uso apropiado de herramientas de SE representa una excelente alternativa para estimular a los estudiantes debido a que ofrecen una amplia gama de posibilidades didácticas, a través de un creciente número de SE de diversos tipos.

Al igual que los libros, pero con la ventaja de interactividad y multimedia, SE pueden servir como herramienta de autoestudio, pues podrían ser capaces de guiar al alumno paso a paso, en la enseñanza, ejercitación y evaluación, ofreciendo una alternativa viable para el autoaprendizaje. Además un SE puede contribuir al monitoreo adecuado del alumno en el aprendizaje, permitiendo un mecanismo accesible para observar el desarrollo del proceso educativo de grupos e individuos, favoreciendo la detección de atrasos. Estas son algunas bondades que pueden presenta los sistemas expertos educativos, también llamados Sistemas Expertos Tutoriales, los cuales se describirán mas a fondo en el Capitulo siguiente.

Como hemos visto los sistemas expertos, además de proporcionar una solución a un problema, también muestran los pasos de "razonamiento" que le permitieron llegar a esa conclusión. Sin embargo, la aplicación de los sistemas expertos a los ambientes educativos muestra grandes dificultades, ya que el arte pedagógico es algo más que solo codificación y procesamiento del conocimiento. Un maestro, como experto humano, no solo muestra el camino hacia la solución correcta, sino que explora otras posibles vías de solución, y su meta principal es que el estudiante sea capaz de resolver toda una clase de problemas. Para lograr este objetivo, el maestro adapta toda su estrategia de enseñanza a las necesidades de un estudiante específico.

Gracias al estudio de la ingeniería del conocimiento y los sistemas expertos que se llevo acabo en este capítulo, podemos diseñar un sistema experto educativo eficiente, el cual debe poseer, además del dominio del experto, estrategias de enseñanza apropiadas que se puedan adaptar a las necesidades específicas de cada estudiante. Finalmente, la interfaz de comunicación entre el sistema y el estudiante debe ser, al mismo tiempo, muy atractiva y poderosa. Lo anterior nos permitirá alcanzar nuestro objetivo antes mencionado.

Capítulo 3. SISTEMAS EXPERTOS EN LA EDUCACIÓN.

3.1 PAPEL DE UN SISTEMA EXPERTO EN LA EDUCACIÓN.

En la actualidad, los Sistemas Expertos en la Educación han venido a revolucionar el campo trayendo nuevas posibilidades de adaptación con respecto a un alumno en particular.

La introducción de la computadora en la educación provocó cambios estructurales tan importantes que surge un nuevo paradigma denominado "Paradigma Tecnológico".

Branson^{10a} dice que el paradigma tecnológico es la representación de un proceso de enseñanza aprendizaje completamente interactivo. En él, cada estudiante interactúa con el maestro, una base de datos, el sistema experto, el tutor y, también, con otros estudiantes.

El estudiante puede aprender ahora directamente de la base de conocimiento y del experto, o de otros estudiantes, sin que el maestro "enseñe" directamente. Esto es, estudiante y maestro pueden interactuar entre sí o con la base de conocimientos y/o con el experto. Es evidente que, para que el estudiante sea capaz de interactuar con el experto, debe existir éste. Se representa este paradigma en la figura No. 7.

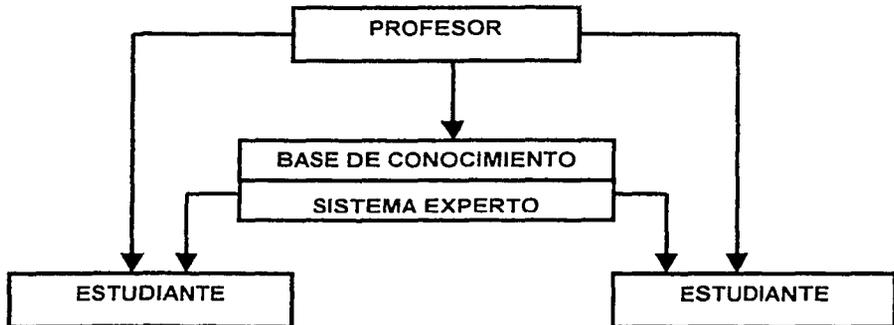


Figura 7. Paradigma Tecnológico.

¹⁰ | De la educación Tradicional a la educación a distancia: cambio de pradigma | http://www.ulsu.edu.mx/public_html/publicaciones/onteanqui/b7/paradigma2.html

La más notable contribución de los Sistemas Expertos a la tecnología educacional o paradigma tecnológico es lo que se conoce como Sistemas Tutoriales Inteligentes (STI), ó ITS de sus siglas en inglés "INTELLIGENT TUTORIAL SYSTEMS" . En un inicio se les llamo a los ICAI (enseñanza inteligente asistida por computadora), término que aún utilizan algunos autores.

Otras personas no desean usar el término inteligente y optaron por nombres como sistemas tutores basados en el conocimiento (KBTS), sistemas tutores adaptables (ATS) (Stretz, 1988) y sistemas de comunicación del conocimiento (Wenger, 1987). Pero la mayor parte de los expertos en esta área están de acuerdo con la denominación de ITS.¹¹

A continuación hablaremos ampliamente de los STI.

SISTEMAS TUTORIALES INTELIGENTES(STI).

STI se define como:

Sistema de enseñanza asistida por computadora, que utiliza técnicas de Inteligencia Artificial, principalmente para representar el conocimiento y dirigir una estrategia de enseñanza; es capaz de comportarse como un experto, tanto en el dominio de conocimiento que enseña (mostrando al alumno cómo aplicar dicho conocimiento), como en el dominio pedagógico, donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje.¹²

El objetivo principal de un STI, es proporcionar ayuda en el proceso de la enseñanza. Para que esto se pueda llevar a cabo de una manera efectiva y eficiente es necesario que éste desempeñe algunas funciones importantes:

- Interactuar con el estudiante por medio de un diálogo de iniciativa mixta, es decir, donde tanto la computadora como el estudiante pueden iniciar preguntas y esperar respuestas razonables.
- Indicar al estudiante las estrategias apropiadas para atacar un problema en particular y demostrarle las aplicaciones de estas estrategias en problemas concretos.

¹¹ D. A. (1987). "What century is this anyway?": 26,27,28.

¹² [Sistemas Tutores Inteligentes]<http://www.sinergia-web.com.mx/clases/tesis/stis.htm>

- Responder al estudiante cualquier pregunta pertinente en términos comprensibles por éste.
- Decirle al estudiante no sólo que está equivocado, e indicarle el método correcto de solución, sino también hacer hipótesis basadas en la historia de errores de éste que es donde se encuentra la verdadera fuente de dificultades.
- Ser capaz de resolver los problemas propuestos, si es posible de diferentes maneras.
- Entender y criticar las soluciones encontradas por los estudiantes.
- Crear un modelo del estudiante y modificarlo continuamente basándose en su desempeño y en reglas de enseñanza.
- Dependiendo de lo que el estudiante demuestre que ha aprendido al resolver los problemas que se le presentan, el sistema debe valorar lo hecho y tomar acciones que atiendan las deficiencias o los logros obtenidos.
- Utilizar los errores de los estudiantes para diagnosticar sus fallas.
- Ser capaz de combinar métodos algorítmicos y heurísticos.

Los métodos algorítmicos son aquellos en que predomina el aprendizaje vía transmisión de conocimiento, desde quien sabe hacia quien lo desea aprender y donde el diseñador se encarga de encapsular secuencias bien diseñadas de actividades de enseñanza que conducen al alumno desde donde está hasta donde desea llegar. El rol del estudiante es asimilar el máximo de lo que se le transmite.

Los métodos heurísticos son aquellos en que predomina el aprendizaje experimental y por descubrimiento, donde el diseñador crea ambientes ricos en situaciones que el estudiante debe explorar conjeturalmente. El estudiante debe llegar al conocimiento a partir de la experiencia, creando sus propios modelos de pensamiento, sus propias interpretaciones del mundo.¹³

ARQUITECTURA DE UN SISTEMA TUTORIAL INTELIGENTE.

Según Kaplan¹⁴ un STI tiene 4 componentes básicos: el módulo sobre el dominio que se enseña, el módulo del estudiante, el módulo pedagógico y

¹³ H., G. P. A. "Estrategias Heurísticas para el uso de Computadores para Enriquecer la Educación." 6: 12.

¹⁴ [Asesores Inteligentes para Apoyar el proceso de enseñanza de lenguajes de progr.] <http://www.Sinergia-Web.Com/Mx/Clases/Tesis/Propuesta.Htm#Figura3>

el módulo de interfaz con el estudiante. A continuación describiremos brevemente de que se trata cada uno.

Módulo sobre el dominio: Es la base de conocimiento del tutor, donde está presentado y almacenado el conocimiento que va a ser enseñado al alumno. Es muy importante que el conocimiento almacenado aquí sea correcto, ya que si no fuera así se estaría dando al alumno una mala enseñanza. Como se mencionó en el capítulo 2 la base de conocimiento incluyen conocimientos declarativo (colección de conceptos de temas relacionados entre sí) y/o procedural (aquel que enseña como realizar ciertas cosas) sobre la materia que se enseña.

Módulo del estudiante: Es toda información que contenga un programa de enseñanza que sea específica para el alumno que esté siendo enseñado. La información puede variar desde el simple cálculo de las respuestas incorrectas que se hayan producido, hasta la complicada estructura de datos que pretende representar una parte importante de los conocimientos del alumno sobre el texto.¹⁵

Este módulo debe representar de forma adecuada las características propias de cada estudiante (capacidad y conducta). Además, hay que tener en cuenta que el proceso de aprendizaje del estudiante se realiza paso a paso, por tanto se debe ser preciso, y tener una actualización continua del modelo del estudiante.

Módulo Pedagógico: El que se encarga de definir la estrategia para transmitir el conocimiento a los usuarios.

Módulo interfaz: Es la forma final del ITS, lo que ve el usuario. Debe tener cualidades como un uso fácil y atractivo, factores muy importantes pues pueden ser definitivas en la aceptación del ITS por parte de un estudiante.¹⁶

CLASIFICACIÓN DE LOS SISTEMAS TUTORES INTELIGENTES.

Los Sistemas Tutores Inteligentes se clasifican por su "rol" en un modelo enseñanza-aprendizaje. A continuación mencionamos los siguientes tipos de modelos:

¹⁵ López Ostio J. Sistemas Tutoriales Inteligentes (ITS). Conferencia mecanografiada. San Sebastián, España: 1993.

¹⁶ [Evolución de la enseñanza asistida por computadoras]
http://bvs.sld.cu/revistas/ema/vol11_1_97/ema05197.htm

Sistemas instruccionales, orientados a transmitir el conocimiento y guiar al estudiante; en estos sistemas, el estudiante ejecuta la tarea "estudiar el lenguaje", de modo que el sistema se comporta como un profesor transmisor. En general, el sistema controla el flujo del trabajo, y dirige la interacción con el alumno. Evidentemente, el rol del sistema se identifica con el de Profesor Transmisor.

Sistemas asesores, orientados a apoyar al estudiante en su aprendizaje, el cual ocurre a través de la ejecución de una tarea de aplicación por parte del estudiante. En estos sistemas, el estudiante se enfoca a la tarea "programar", y el sistema se comporta como un asesor, es decir, como un auxiliar que realiza sugerencias al alumno de acuerdo a su criterio, con el objetivo de que el estudiante mejore su desempeño. En este caso, el estudiante dirige la tarea, y la realiza en el orden que considere conveniente; inclusive, puede controlar la interacción con el sistema. A este rol se le definió, precisamente, como el rol de Profesor Asesor.

Finalmente podemos decir que los sistemas expertos en la educación tienen las siguientes características:

- Resultan ser buenos dispositivos educativos debido a su posibilidad de ver hacia atrás o volver atrás de una determinada solución.
- El alumno y profesor puede llegar a obtener un conocimiento más detallado del tema de que se trata.
- Son capaces de modelar las necesidades específicas del estudiante, así como estrategias de enseñanza apropiadas. Como en es el caso que ocupa a los STI antes mencionados.

3.2 SISTEMAS EXPERTOS QUE ADIESTRAN Ó ENSEÑAN.

Se entiende por adiestramiento: la transmisión de conocimiento sobre cómo hacer bien una tarea (know-how)

Proceso de formación de actitudes con el objeto de desarrollar destrezas en el arte de saber vivir (R.MUNIZAGA).¹⁷

¹⁷ (1995). *Diccionario de las Ciencias de la Educación*. México, D.F., Editorial Santillana

Se entiende por enseñar:

Según R. Titone¹⁸, es un acto en virtud del cual el docente pone de manifiesto los objetos de conocimiento al alumno para que éste los comprenda.

Actualmente los sistemas expertos combinan el adiestramiento y la enseñanza a la par, pues ayudan dentro de la simulación de problemas complejos. Los sistemas expertos como se menciono en el capítulo 2 están diseñados para recoger y registrar aquellos aspectos del experto humano, necesarios para la toma de decisiones, así como el comportamiento del experto ante dicha situación. Esta característica les permite ser muy útiles en la enseñanza porque **simulan o imitan** la realidad; convirtiéndose en un excelente método de experimentación.

Su aporte dentro del campo educacional y laboral estriba en que el estudiante puede ser capaz no sólo de aprender, sino de tomar decisiones, como lo haría en su ambiente de trabajo, laboratorios reales; y aprender de la experiencia, etc., reduciendo el costo de capacitación y los riesgos que un empresa tendría al contratarlo.

Por otro lado, la curva de aprendizaje en el terreno laboral podrá ser menor porque los egresados estarán mejor preparados, pues han experimentado y aprendido mediante la **simulación**, reduciendo los riesgos para las empresas de contratar personas sin experiencia.

3.3 SOFTWARE EDUCATIVO DE SISTEMAS EXPERTOS TUTORIALES DE FÍSICA.

Existe toda una variedad de software que puede ser de utilidad para los centros educativos, sin embargo es de particular interés abordar el tema del software educativo para llegar a conocer las ventajas y limitaciones de los STI que se abordará en la siguiente sección.

Como parte de este trabajo, se realizó una revisión sobre los ITS existentes, en particular en Física.

¹⁸ (1995). *Diccionario de las Ciencias de la Educación*. México, D.F., Editorial Santillana

Dentro del software educativo¹⁹ los materiales encontrados los clasificamos de acuerdo a Sistemas Instruccionales y Sistemas Asesores que se mencionaron en los modelos de Sistemas Tutoriales Inteligentes.

SISTEMAS INSTRUCCIONALES:

BearEduPhysics-1

Este software esta diseñado para ayuda del profesor ya que complementa su clase. Este software maneja animaciones y ejemplos prácticos que facilitan el entendimiento del alumno. Los temas que maneja son vectores, equilibrio, momentos, fuerzas, leyes de Newton, sonido, gases.²⁰

Interactive Physics

En este software esta diseñado especialmente para el profesor, pues le permite integrar modelos y simulación en un curriculum de física, crear modelos dibujando en la pantalla con una interface gráfica, adicionar objetos tales como resortes, cuerdas y juntas, medir los atributos de sus objetos tales como velocidad, aceleración, momento y energía, e interactuar con su modelo en tiempo real cambiando los parámetros mientras corre la simulación.²¹

ED-SIPLAC

Sistema de arquitectura modular, de forma que cada uno de los prototipos consta de diversos módulos, algunos de los cuales pueden ser utilizados por más de un prototipo. El hardware y el software construidos se han diseñado para trabajar englobados dentro de dicha arquitectura modular, es decir, cada componente físico o lógico pertenecerá a un módulo. De esta forma, se permite también a los usuarios (centros de enseñanzas medias) la adquisición de componentes complementarios a los ya disponibles. Este sistema reproduce

¹⁹ El calificativo de "educativo" se añade a cualquier producto diseñado con una intencionalidad educativa. Los programas educativos están pensados para ser utilizados en un proceso formal de aprendizaje y por ese motivo se establece un diseño específico a través del cual se adquieran unos conocimientos, habilidades, procedimientos, es decir, para que un estudiante aprenda.

²⁰[BearEdu Technologies] <http://bearedu.com/>

²¹ [Interactive Physics]
http://www.software-shop.com/Productos/Interactive_Physics/interactive_physics.html.

simulaciones en tiempo real, teniendo antecedentes de modelización de la realidad (Leyes físicas que rigen el experimento).²²

XyZET

Esta aplicación está pensada como un potente laboratorio donde el profesor puede realizar una gran variedad de experiencias, XyZET es una herramienta para realizar simulaciones en tres dimensiones. El usuario puede situar partículas en un cubo y asignándoles masa y carga. Las fuerzas entre las partículas pueden especificarse como elásticas, gravitatorias ó eléctricas, y su valor puede modificarse una vez fijadas las condiciones iniciales. Los posteriores estados del sistema pueden calcularse de acuerdo a las leyes de la física y son mostrados gráficamente. El usuario puede modificar su punto de observación en cualquier momento sin interrumpir el programa para inspeccionar el sistema bajo distintas condiciones de perspectiva. Teniendo en cuenta la constitución atómica de la materia, el programa ayuda a evitar conceptos erróneos que se inducen por la pronta introducción del sólido rígido en la enseñanza que sigue métodos clásicos. Una vez diseñada una experiencia ésta es almacenada para poder ser utilizada con posterioridad dentro de una unidad didáctica.

SISTEMAS ASESORES:

EM Field

EM Field software comercial interactivo que facilita la exploración rápida y cualitativa de campos eléctricos y magnéticos, así como una comprensión intuitiva de los campos de fuerza, leyes de Gauss, ley de Ampere y el concepto de flujo. Permite configurar las cargas y corrientes; los vectores del campo magnético se actualizan en tiempo real cuando se carga una corriente, produciendo una gráfica dinámica y global de los campos en una región. Además tiene un juego, que reta al estudiante a utilizar lo que ha aprendido para determinar posiciones y magnitudes de cargas y corrientes ocultas²³.

²² <http://www.c5.cl/ieinvestiga/actas/ribie96/EDSIPLAC.html>

²³ [Asesores inteligentes para apoyar el proceso de enseñanza de lenguajes de programación] http://wasnet03ws.physics.ncau.edu/pasnew/em_field/emf.html

Interactive Journey Through Physics

Interactive Journey Through Physics, MAC and Windows, software interactivo que ha sido diseñado para aumentar experiencias en el aprendizaje tradicional de lección, laboratorio, y texto. Tiene componentes como simulación, animación, video y solución de problemas interactivo, los cuales le permiten al estudiante interactuar y visualizar conceptos donde

en la forma tradicional no es posible. Se encuentra organizado en cuatro áreas: Electricidad y Magnetismo, Mecánica, Luz, Óptica²⁴.

Knowledge Physics

Knowledge Physics es un programa comercial interactivo donde el estudiante utiliza animaciones, gráficas y sonido. Maneja los temas de movimiento, fuerzas, comportamiento molecular y atómico, electricidad, acción de onda, óptica y física de partículas.

Programa que proporciona algunos tutoriales y permite medir el progreso del estudiante con reportes y evaluación integrados²⁵.

Play Physics

Play Physics es un software interactivo de simulación para estudiantes que estudian Física. Contiene hasta 36 conceptos diferentes. Este software provee hechos y explicaciones de fenómenos físicos y laboratorios basados en computadoras donde los estudiantes configuran las condiciones del laboratorio y proporcionan variables para cada experimento²⁶.

The Video Encyclopedia of Physics Demonstrations

Colección de 25 discos láser con 600 demostraciones utilizadas en la enseñanza de la Física a estudiantes universitarios y de secundaria. La serie utiliza avanzada tecnología computacional, de animación y de fotografía en cámara lenta para maximizar la experiencia de los estudiantes.

²⁴ <http://vig.prenhall.com/acadbook/0,2581,0132581,0132541033,00.html>

²⁵ [Knowledgebase™ Series] <http://www.knowledgefactory.com>

²⁶ [Play Physics] <http://www.i-t.com/engaw/jason/pp2.htm>

Lo anterior confirma la necesidad de solución si usamos programas de simulación como herramienta educativa; sin embargo para maximizar sus beneficios es necesario su uso sea acorde a una serie de objetivos y criterios prioritarios, que orienten el diseño y la aplicación didáctica del software.

Es aquí donde el docente(profesor) debe organizar el proceso de instrucción en torno a un conjunto de secuencias de actividades que permitan al alumno reflexionar sobre la información que recibe y poner en juego sus esquemas conceptuales, para lograr una asimilación integradora del nuevo conocimiento. Esto posibilita mejorar la eficacia instruccional de muchos programas de simulación que en principio solo han sido elaborados como instrumentos orientados a la visualización de fenómenos físicos y a la resolución matemática de los problemas que son objeto de simulación.²⁷

Ultimate Einstein

Ultimate Einstein contiene una historia interactiva de la vida de Einstein, mejorada en video y audio clips. El alumno encuentra un Einstein virtual que le contesta preguntas de una variedad de materias. Incluye una serie de experimentos donde el alumno puede demostrar las teorías de Einstein. Introduce al alumno en las teorías de Einstein acerca del espacio tiempo, luz y gravedad²⁸.

Virtual Physics

Virtual Physics es un programa comercial que maneja temas de ondas, luz, electricidad, espejos y lentes, magnetismo, método científico, relojes y unidades estándar. Cada tema es presentado en una variedad de formas incluyendo un tutorial, contiene información exacta, un ejercicio que es explicado y puede ser conducido y observado por el estudiante en forma confiable, y preguntas interactivas que enseñan como funcionan ciertas cosas. El estudiante debe contestar correctamente dos preguntas para que el programa el de una pieza clave de escape. Además incluye un glosario en términos científicos²⁹.

²⁷ <http://www.maseducativa.com/webs/herrera/art1pag2.htm>

²⁸ [The Ultimate Einstein] <http://www.cdromshop.com/cdshop/desc/p.793763000218.html>

²⁹ [Visual Quantum Mechanical] <http://phys.educ.ksu.edu/>

Además se realizó una investigación de los materiales de sistemas expertos tutoriales ó ITS de Física en México, en el ILCE(Instituto Latinoamericano de la Comunicación Educativa). Desafortunadamente no fue posible encontrar precisamente sistemas expertos tutoriales, sino material que complementa a los profesores de Física en nuestro país.

Esto habla de la urgencia que existe a nivel nacional de este tipo de sistemas.

Los sistemas que se encontraron son:

- Universo Mecánico que consta de 13 videos se encuentra en el ILCE(Instituto Latinoamericano de la Comunicación Educativa).
- Física Elemental consta de 2 videos.
- Cómo funcionan las cosas 3.0 se encuentra en CD-ROM.
- Enciclopedia de la ciencia 2.0 se encuentra en CD-ROM.

Observando con esto que es de gran utilidad incorporar dentro de los planes educativos, tutoriales inteligentes que complementen el aprendizaje de los alumnos.

3.4 VENTAJAS Y LIMITACIONES DE LOS SISTEMAS EXPERTOS TUTORIALES QUE CONTROLAN EL SOFTWARE EDUCATIVO.

A continuación mostramos las ventajas que se encuentran en un STI, así como sus desventajas:

VENTAJAS:

- Facilita el desarrollo de procesos de producción en serie.
- Se reduce la inversión de recursos
- Los medios utilizados - texto, imágenes, videos y audio-, se pueden trabajar por separado y con ayuda de gente no experta en el desarrollo de aplicaciones multimedia, con esto se reduce le tiempo de producción.
- Se generan aplicaciones a gran escala.

DESVENTAJAS:

El alumno recibe instrucciones y reacciones de guía por parte del sistema que pueden ser bastante restringidas, puesto que están basadas en una referencia preestablecida acerca del estudiante y no en la evolución de su conocimiento.

En muchas ocasiones, el estudiante busca adaptarse a las características del tutor y así poder optimizar su uso para efectos de resolución de problemas. Donde muchas veces el sistema no ofrece construir situaciones en las que sea posible generar perturbaciones adecuadas para la mejor comprensión del estudiante.

3.5 LA CAPACIDAD DE LOS SISTEMAS EXPERTOS ACTUALES PARA MODELAR LA EMPATÍA DEL PROFESOR Y ALUMNO.

Las tendencias actuales en la enseñanza y la evaluación se dirigen hacia la diferenciación de los estudiantes, pues cada día se hace más evidente la existencia de variables educativas con diferentes valores en los estudiantes; tendencia que se conoce precisamente como educación personalizada.

En la actualidad, a medida que los **Sistemas expertos** interactúan con el estudiante, se va realizando un registro constante sobre el progreso que adquiere en el aprendizaje, o sea, se almacenan las clases recibidas, preguntas formuladas y cómo las respondió, errores cometidos, tiempo en que se asimila un nuevo contenido y recomendaciones, entre otras, de forma tal que el sistema, al analizar esta información, pueda adoptar una estrategia adecuada de trabajo con el estudiante, ó solo propone una nota al alumno tal como lo haría un profesor en el salón de clases.

Estos sistemas pueden emplear o no multimedia, y a diferencia del resto de software educativo, no ofrecen un conjunto de opciones cada vez que se equivoca el alumno. Como se menciono anteriormente solo guían al estudiante para que corrija los errores por medio de un replanteamiento en su razonamiento, tips, o un simple comentario tal como lo haría un tutor humano(profesor).

El tutor Inteligente para la enseñanza conceptual e interactiva de la física a nivel bachillerato, dentro del cual nos encontramos

participando, a considerado desde su planteamiento inicial contemplar tanto al alumno, como al profesor.

Ambos son importantes porque el profesor es el experto que conoce las estrategias de enseñanza para que un alumno comprenda determinado tema; y el alumno el que aprende e interactúa con el sistema de manera personalizada.

Razón por la cuál se requiere actualmente un proceso enseñanza-aprendizaje donde prealezca el carácter participativo, autodirección y experticidad en la enseñanza, de manera que se pueda representar el conocimiento de acuerdo a los intereses y posibilidades de alumnos y profesores.

CAPÍTULO 4. DESARROLLO DE LA "GUÍA INTELIGENTE PARA UN TUTORIAL DE FÍSICA A NIVEL MEDIO SUPERIOR".

4.1 ANÁLISIS DEL PROBLEMA.

Sabemos que los factores que determinan el nivel de desarrollo de un país son muy diversos. Entre otros podemos mencionar los sociales, económicos, educativos y culturales.

En lo que respecta al factor educación, México vive un profundo rezago que se ha acentuado en las últimas décadas [G. Guevara, 1992]. El sector educativo, en todos sus niveles, responde cada vez menos a las necesidades de un mundo que cada vez está más interrelacionado y globalizado, con múltiples retos científico-tecnológicos, a los que como país vemos cada vez más inalcanzables.

Aún cuando son múltiples los esfuerzos que se tienen que realizar para alcanzar la viabilidad como país y aspirar a mejorar los niveles de vida de la población, por diversos motivos, desafortunadamente, estos esfuerzos han sido insuficientes en nuestro caso. Los pocos indicadores educativos muestran que somos un país con un bajo promedio escolar [G. Guevara, 1992] (en 1992 el promedio escolar era 6.5 [F.Martínez, 1992] y en el año 2000 era de 7.2 [E.Zedillo, 2000]). Este indicador es un reflejo del fracaso, principalmente, de políticas educativas y estrategias didácticas que no han tenido impacto positivo en el proceso enseñanza aprendizaje; la infraestructura deficiente e insuficiente y la carencia de material didáctico adecuado.

Sin embargo, estas estadísticas, aunque desalentadoras, no señalan que debemos cruzarnos de brazos y resignarnos.

Por esta razón, el objetivo principal de este trabajo es contribuir a los esfuerzos que se realizan en el área educativa, de modo a cumplir de una manera adecuada con la formación que los alumnos requieren para ingresar al nivel educativo superior, con un alto índice de preparación y capacitación.

4.2 PLANTEAMIENTO DEL PROBLEMA.

Al igual que otros métodos de instrucción que hasta hace unos años eran técnicas innovadoras que prometían mejorar el nivel de la enseñanza (ej. la enseñanza programada, la televisión educativa, los programas tutoriales de audio, etc.), las computadoras se han convertido en un poderoso auxiliar en el proceso enseñanza/aprendizaje. Desafortunadamente, al igual que en los enfoques anteriores, las computadoras, no han cubierto todas las esperanzas acerca de los beneficios positivos y duraderos que traerían a la enseñanza. A pesar de lo anterior, los intentos por desarrollar programas, que vayan mejorando las deficiencias que han mostrado sus precursores, continúan.

En el CCADET se planteó la necesidad de crear una herramienta didáctica computacional, accesible a una gran población estudiantil, que sirviera para experimentar con diferentes estrategias de enseñanza.

La mayoría del software tutorial presenta la información de una manera muy lineal y poco individualizada y para una audiencia demasiado grande y diversa. En realidad no están diseñados para interactuar con el usuario y modificar su presentación posterior en base a sus necesidades. No presentan una retroalimentación inmediata, instrucción individualizada, y neutralidad.

Para cubrir las deficiencias antes mencionadas es indispensable que tanto los usuarios como las personas que participan en el desarrollo de los módulos se comuniquen durante las fases de producción de software educativo (pre-producción, producción, post-producción) para alcanzar su objetivo. Estas personas son:

- Profesor.
- Alumnos.
- Equipo de desarrollo.

Cuyo rol en el desarrollo de software educativo se menciona a continuación.

EL ROL DEL PROFESOR

Durante la fase de **pre-producción** el profesor es el encargado de definir las necesidades de aprendizaje y la administración del proceso de enseñanza en donde se seleccionan, estructuran, identifican, y se aportan los conocimientos, dudas y estrategias de enseñanza-aprendizaje del contenido de un curso en base a su experiencia. Esta experiencia es adquirida a través de la impartición de clases, y de la observación del comportamiento de los alumnos en clase.

El profesor debe ser, ante todo, un buen facilitador del aprendizaje, así que una vez seleccionados los contenidos, el profesor debe decidir qué tipo de actividades son las más adecuadas e interesantes para el desarrollo del aprendizaje del alumno.

Gracias a lo anterior el profesor puede aportar estrategias pedagógicas que hagan un software educativo atractivo e interesante para el alumno, en donde se visualicen ejemplos, y situaciones provocadoras que centren la atención del alumno ante dicho material.

En la fase de **producción** el profesor participa activamente, interactuado con el equipo de desarrollo. Su participación consiste en verificar y aportar conocimiento e ideas sobre el producto final que desea obtener.

En la actualidad esta etapa de producción es un proceso muy lento puesto que es difícil la comunicación entre expertos y desarrolladores, debido a que el experto muchas veces no sabe expresar sus necesidades, y el equipo de desarrollo tiene que interactuar muchas veces para concretar ideas, por lo que se le debe dar al profesor la opción de participar en la creación de sus propios materiales de manera independiente.

En la fase de **post-producción** el profesor debe ser capaz de **validar** que los objetivos pedagógicos planteados en las etapas anteriores se lleven a cabo de manera satisfactoria.

Proporcionar sugerencias para mejorar el producto final.

EL ROL DEL EQUIPO DE DESARROLLO

En la fase de **pre-producción** el equipo de desarrollo es el encargado de realizar metodologías de análisis, y diseño de software.

En esta etapa se especifica la función del software, así como la descripción de la interfaz y el establecimiento de las restricciones (definir alcances del sistema) de diseño que debe considerar el software.

El equipo de desarrollo debe establecer contacto con el profesor para establecer las características de la interfaz del sistema y descubrir las restricciones de diseño.

Asimismo, debe evaluar el flujo y la estructura de la información, definir y elaborar las funciones del software, entender cual será el comportamiento del programa.

Tanto el profesor como equipo de desarrollo identifican las necesidades del alumno en cuanto a:

- Reconocer la importancia del material que se le entrega, ejemplos, y situaciones provocadoras que le permitan al alumno la comprensión del tema.
- Confrontar sus conocimientos previos (visión del mundo) con respecto a los conceptos que se le enseñan. Donde se involucre completamente en el proceso de aprendizaje, de forma consciente, preparado para analizar los conceptos e ideas que se le presentan en el material de enseñanza, y debe ser capaz de reflejar y probar el nuevo conocimiento adquirido.

En la fase de **producción** el equipo de desarrollo genera la programación y el desarrollo de software a través de una: especificación, metodologías de análisis y diseño de software y las técnicas de pruebas.

En la fase de **post-producción** el equipo de desarrollo es el encargado de presentar al profesor y al alumno el producto final con el objetivo de recabar sugerencias que le permitan mejorar el software educativo, de manera que satisfaga las necesidades reales. Además, en esta fase es posible descubrir errores y deficiencias que pudiera presentar dicho sistema.

Asimismo validar que los objetivos planteados se cubren satisfactoriamente.

EL ROL DEL ESTUDIANTE

En el desarrollo de software educativo de calidad es importante considerar las necesidades del estudiante.

En la fase de **pre-producción** el alumno es observado por el profesor y el equipo de desarrollo de manera que se puedan detectar sus necesidades como son:

- Motivación, capacidad de planificar su estudio .
- Creencias sobre algún tema en particular.
- Estrategias pedagógicas que requiere el alumno.

En la fase de **producción** el alumno prueba los prototipos generados por el equipo de desarrollo. Este último, junto con el profesor, observan al alumno y lo cuestionan acerca del material presentado, con el objetivo de recabar opiniones, que permitan mejorar dicho material, de manera que se pueda obtener un producto final que satisfaga las necesidades del alumno.

En la fase de **post-producción** el alumno al igual que en la de producción emite opiniones acerca del material que se le entrega, de modo de mejorar dicho material en futuras versiones. En esta fase el alumno es observado por el equipo de desarrollo para detectar posibles errores en el funcionamiento del sistema, y así corregirlos.

Es importante resaltar que la capacidad de aprender a través de este sistema no está desarrollado en nuestros actuales alumnos y no se desarrollará en ellos de manera automática. Por lo que debe crearse un entorno motivador que les enseñe a utilizar adecuadamente la metodología.

RELACIÓN TRIPARTITA EN EL DESARROLLO DE SOFTWARE EDUCATIVO

Por lo anterior al hacer referencia al desarrollo de software, en especial, al educativo, es evidente la necesidad de contar con personas altamente capacitadas, tanto en educación como en desarrollo de software que tomen en cuenta las necesidades del alumno.

El proceso de desarrollo de este tipo de software en la actualidad presenta una gran dependencia entre equipo de desarrollo, profesores y alumnos.

Esta dependencia se muestra en la siguiente figura.

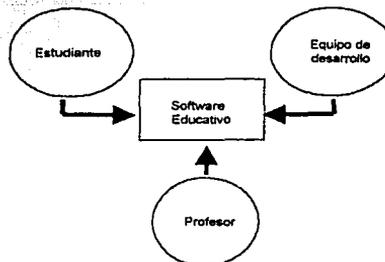


Figura 8. Dependencia entre Equipo de Desarrollo, Profesor y Estudiante

En la figura 8 se ve la dependencia entre profesor, alumno y equipo de desarrollo. El problema de este enfoque es:

- El tiempo de producción de software educativo es muy lento por la excesiva interactividad entre equipo de desarrollo y profesor, pues no se logran concretar los objetivos rápidamente.

Con base en lo anterior, desarrollamos la Guía Inteligente para un Sistema Tutorial de Física, la cuál nos permitirá acortar los tiempos de desarrollo, además de permitir al profesor desarrollar su propio tutorial.

Este trabajo esta dividido en 3 partes fundamentales:

- Cascarón (en Authorware) para diseño de tutoriales.
- Módulo de adquisición del conocimiento (en Visual Basic) para el profesor.
- Módulo de Material Multimedia (Juegos y Animación por medio deFlash) para introducir y ejemplificar el uso de juegos interactivos en los tutoriales.

Así, se propone desarrollar, por un lado, un módulo que permita al experto(profesor) verter su conocimiento de forma sencilla, para generar un tutorial sobre una materia en particular. A su vez, este módulo permitirá alimentar un Cascarón programado en Authorware; el

cual se encargará de administrar la información que el experto ha brindado y lo presentará al estudiante. Finalmente, se presentará un material multimedia para dar un ejemplo práctico del uso de nuestro sistema.

Para tener clara cuál es la relación que tienen los módulos mencionados, presentamos el siguiente diagrama de bloques, en el que las flechas representan el flujo de la información:

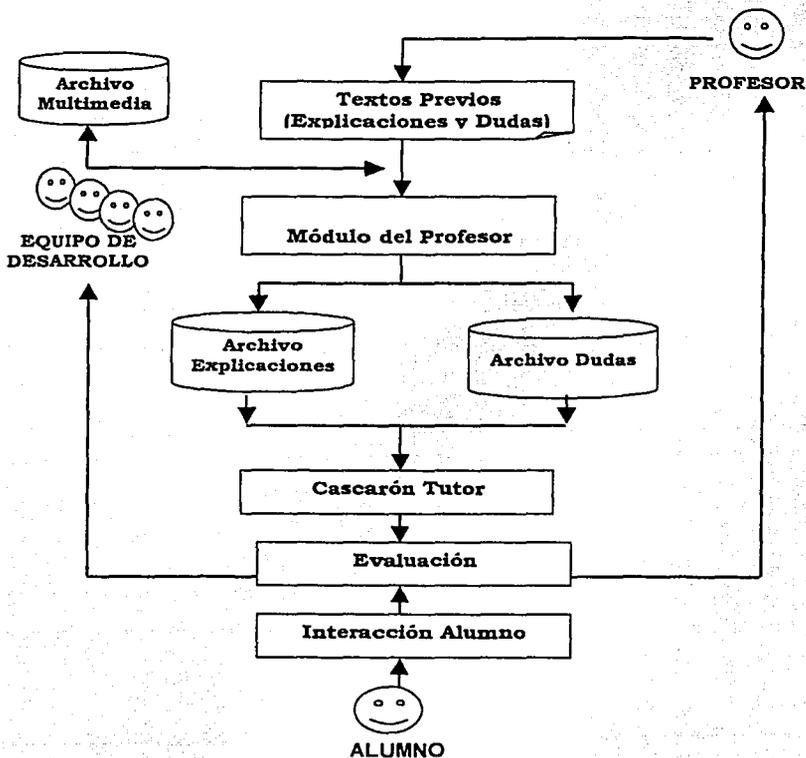


Figura 9. Esquema del proceso enseñanza-aprendizaje.

En este esquema, el profesor elabora en primer lugar, un guión en donde expone el desarrollo de un tema en particular.

Posteriormente, con la ayuda de este guión, el profesor captura su conocimiento en el "Módulo del profesor", con la cual se genera la primera versión de los archivos que contienen la información necesaria que le permiten al profesor explicar de manera clara y sencilla los conceptos y dudas más importantes de un tema (Archivo Explicación y Archivo Dudas).

A continuación el módulo "Casarón del Tutor" recupera la información de dichos archivos, generándose así una primera versión del tutor. Esta primera versión puede ser probada por alumnos de manera que proporcionen sugerencias al profesor con respecto al material estudiado.

De esta forma el profesor puede realizar posibles modificaciones a este material. Además, el equipo de desarrollo puede probar esta versión y obtener la información necesaria que le permita desarrollar archivos multimedia que modelen de manera atractiva dicha información.

Teniendo una primera versión de los archivos multimedia, el profesor puede apreciar dichos archivos a través de el "Módulo del profesor" con el fin de observar los avances del equipo de desarrollo y proporcionar sugerencias. A la vez, el alumno evalúa la integración de archivos independientes junto con archivos multimedia en el módulo "Casarón del Tutor", aportando ideas tanto al profesor como al equipo de desarrollo que permitan enriquecer el tutorial.

Estos módulos "Módulo del profesor" y "Casarón del tutor" tienen la ventaja de reducir el tiempo de producción de software educativo y la comunicación entre profesor y equipo de desarrollo, al mismo tiempo el producto final cubre las necesidades del alumno.

4.3 DISEÑO E IMPLEMENTACIÓN DE LOS MÓDULOS.

4.3.1 MÓDULO DEL PROFESOR (ADQUISICIÓN DE CONOCIMIENTO)

ADQUISICIÓN Y ESTRUCTURACION DEL CONOCIMIENTO

Para el desarrollo de software educativo, sabemos que es necesario el proceso de adquisición del conocimiento, en el cuál

participan profesor, alumno y equipo de desarrollo. Así pues, en esta etapa, se realizó una discusión con expertos en Física sobre los aspectos necesarios para realizar los módulos para el tutorial inteligente de Física de manera eficiente.

PRIMERAS ENTREVISTAS

Durante las sesiones con los expertos en Física se discutió sobre la forma en que dan su clase, y cuál es la manera de abordar las dudas más frecuentes de un tema en particular. Así como cuáles eran los pasos que seguían para mejorar el contenido de su curso.

Previo a las entrevistas estudiamos el material referente al texto

"Los Sistemas de coordenadas", proporcionado por el Dr. Fermín Viniegra y "Movimiento" del M.C. José Luis Pérez Silva. Lo anterior nos permitiría tener una mejor comunicación con los expertos; es decir hablar sobre el tema con el experto en el mismo lenguaje que éste acostumbra hacerlo.

Buscamos que las sesiones tuvieran la finalidad de:

- 1) Recolectar los conceptos más importantes, los conocimientos y experiencias de un tema que el experto domina y como exponerlo por medio de la computadora a los alumnos en una manera interactiva y clara.
- 2) Recolectar las dudas más frecuentes observadas en los alumnos en un tema en particular y buscar la manera de presentarlas para que fueran atractivas para el alumno.
- 3) Analizar las demandas de los expertos para que ellos tuvieran un módulo para recolectar su conocimiento y desarrollar su propio tutorial de una manera sencilla y clara.

Como resultado de estas reuniones, y siguiendo con las recomendaciones generales para la adquisición del conocimiento que se vieron el capítulo 2, logramos obtener información valiosa que nos permitió, posteriormente, modelar dicho conocimiento y convertirlo en parte de los módulos y del material que desarrollamos.

Esta información fue:

- Al alumno se le debe explicar un tema de diferentes maneras, de modo que si el alumno no entendió el tema se retroalimente hasta que comprenda.
- Las dudas frecuentes que se presentan en los alumnos también se deben volver a explicar.
- Entre más este ligada una explicación a la vida diaria de un alumno, más interés despierta en ellos.
- Debe existir un orden en el temario.
- El aprendizaje no sólo es teoría, sino que se debe complementar con distintos ejemplos y materiales.

En las reuniones subsecuentes se trabajó en cuál sería la forma adecuada de presentar al alumno los conceptos y las dudas. Elegimos para presentar los conceptos, animaciones Multimedia, dada la facilidad de representar fenómenos naturales que captan y mantienen la atención del usuario en todo momento y la aceptación que tienen este tipo de materiales en la actualidad.

En cuanto a la manera de resolver las dudas, acordamos dos formas de abordarlas:

- 1) Presentándole al alumno una explicación diferente de los conceptos revisados.
- 2) Retando al alumno con juegos interactivos e inteligentes.

Más adelante se retomaran más en detalle dichos juegos, en la sección de juegos didácticos.

Una vez terminadas las entrevistas, se inició la implementación de una primera versión del material del profesor, como una animación multimedia (de esta animación se hablará mas en detalle en la siguiente sección).

Una vez desarrollada la primera versión del material, se asistió de forma periódica a entrevistas con los expertos, donde fueron realizadas las modificaciones pertinentes.

Posteriormente con ayuda de expertos del CCADET, nos dimos a la tarea de diseñar el juego interactivo que se denominó "Morita La Pulguita"(de este juego se hablará en la sección de juegos didácticos y materiales multimedia). Durante el desarrollo de estas primeras

versiones encontramos algunos problemas que a continuación se describen.

En nuestro caso, las entrevistas no resultaron ser el método mas efectivo en el proceso de adquisición del conocimiento, debido a que los expertos, dadas sus múltiples ocupaciones, no siempre estaban disponibles en el momento en que se les necesitaban, observando con lo anterior la dependencia existente entre equipo de desarrollo y profesor, originando el aumento significativo del tiempo de desarrollo, repercutiendo en el avance de nuestro proyecto. Por lo tanto se decidió utilizar el método de adquisición del conocimiento automático, el cuál nos permitió plantear un módulo que nos facilitará la adquisición del conocimiento, economizar recursos humanos y, lo más importante, el tiempo, tanto del ingeniero de conocimiento como el del mismo experto. Además, este enfoque presenta la ventaja que el experto mismo puede manipular su conocimiento directamente, así como hacer modificaciones.

El equipo de desarrollo, al percatarse de la riqueza y el potencial que tenía la información capturada con el módulo descrito, se dio a la tarea de desarrollar un sistema capaz de tomar dicho material y presentarlo al alumno de manera organizada.

De acuerdo a lo anterior decidimos desarrollar los módulos descritos.

MÓDULO DEL PROFESOR

El objetivo primordial del "Módulo del profesor" es que el experto(profesor) vacíe su información sin depender del equipo de desarrollo, utilice adquisición de conocimiento en forma automática, genere sus propios materiales, organice el conocimiento del profesor sobre la manera en que hay que explicar un tema particular, así como la sincronización de la explicación con eventuales animaciones desarrolladas para ese tema, defina e integre las dudas frecuentes de los alumnos, y señale los materiales que deberán ser llamados para resolver dichas dudas.

Se planteó el siguiente algoritmo y diagrama de flujo.

4.3.1.1 ALGORITMO DEL "MÓDULO DEL PROFESOR".

- 1.- Seleccionar Tema, Subtema.
- 2.- Seleccionar si es una Explicación ó una Duda
 - 2.1.- Si elige Explicación:
 - 2.1.1 Verifica si ya existe.
 - 2.1.1.1 Si ya existe le muestra al profesor su información almacenada, así como el archivo Multimedia asociado.
El profesor puede integrar nuevos textos, modificar y/o eliminar los ya existentes.
 - 2.1.1.2 Si No existen.
El usuario genera nuevos datos para una explicación, y guarda su conocimiento.
 - 2.2- Si el profesor elige Dudas.
 - 2.2.1 El sistema verifica si existe.
 - 2.2.1.1 Si existe.
El profesor puede visualizar los textos existentes, redactar nuevos datos, modificar, y generar textos para cada duda, y guardar sus cambios.
 - 2.2.1.2 Si no existen.
El profesor puede redactar textos para cada duda, borrarlos, y guardar los cambios de su duda.

El diagrama de flujo del pseudocódigo presente se muestra en el apéndice A, Figura 1.

4.3.1.2 HERRAMIENTAS DE DESARROLLO.

Para el desarrollo de este módulo se utilizó el siguiente software:

a) Sistema Operativo.

La plataforma en que se desarrolló este módulo fue el Sistema Operativo Windows 98. Puesto que es la plataforma que tienen los usuarios.

b) Software de desarrollo.

Visual Basic 6.0 también se eligió puesto que es un lenguaje de programación orientado a eventos, siendo los eventos cosas que realiza el usuario con el teclado o el ratón mientras el programa se encuentra ejecutándose. Visual Basic contiene una serie de instrucciones que

4.3.1.1 ALGORITMO DEL "MÓDULO DEL PROFESOR".

- 1.- Seleccionar Tema, Subtema.
- 2.- Seleccionar si es una Explicación ó una Duda
 - 2.1.- Si elige Explicación:
 - 2.1.1 Verifica si ya existe.
 - 2.1.1.1 Si ya existe le muestra al profesor su información almacenada, así como el archivo Multimedia asociado.
El profesor puede integrar nuevos textos, modificar y/o eliminar los ya existentes.
 - 2.1.1.2 Si No existen.
El usuario genera nuevos datos para una explicación, y guarda su conocimiento.
 - 2.2- Si el profesor elige Dudas.
 - 2.2.1 El sistema verifica si existe.
 - 2.2.1.1 Si existe.
El profesor puede visualizar los textos existentes, redactar nuevos datos, modificar, y generar textos para cada duda, y guardar sus cambios.
 - 2.2.1.2 Si no existen.
El profesor puede redactar textos para cada duda, borrarlos, y guardar los cambios de su duda.
El diagrama de flujo del pseudocódigo presente se muestra en el apéndice A, Figura 1.

4.3.1.2 HERRAMIENTAS DE DESARROLLO.

Para el desarrollo de este módulo se utilizó el siguiente software:

a) Sistema Operativo.

La plataforma en que se desarrolló este módulo fue el Sistema Operativo Windows 98. Puesto que es la plataforma que tienen los usuarios.

b) Software de desarrollo.

Visual Basic 6.0 también se eligió puesto que es un lenguaje de programación orientado a eventos, siendo los eventos cosas que realiza el usuario con el teclado o el ratón mientras el programa se encuentra ejecutándose. Visual Basic contiene una serie de instrucciones que

permite reconocer los diversos movimientos que hace el usuario, lo que permite al programa saber cuando debe ejecutar alguna rutina específica.

Actualmente, con la maduración de sus funciones, y los mecanismos para conectar bases de datos. Pudiéndose así ofrecer una potencia incomparable y de gran facilidad de utilización para las necesidades de los desarrolladores que trabajan bajo Microsoft, sobre todo, en el rendimiento y productividad cuando se acceda a datos.

Otra herramienta que se utilizó fue Access 97 puesto que Microsoft Access es un sistema gestor de bases de datos relacionales (SGBD) realizado por Microsoft.

Cabe resaltar que se utiliza la versión de Access 97, puesto que es compatible con los controles DataGrid y ADODC existentes en Visual Basic 6 .

De los controles de Visual Basic seleccionamos los siguientes, que consideramos ser las más apropiadas para la aplicación.

ADO: (Active X Data Objects/ Objetos de Datos Active X). ADO es una interfaz directa a la fuente de datos. Microsoft la considera como su tecnología máxima de acceso de datos.

DATAGRID: Es un control Active X. Desarrollado por APEX Software Corp. Permite a los usuarios ver y editar múltiples filas de datos simultáneamente. Este control es útil para introducir grandes cantidades de datos.³⁰ Además que permite editar los textos contenidos dentro de él.

OLE DB: Es amplio conjunto de interfaces del Modelo de objetos componentes (COM) que pueden proporcionar un acceso uniforme a datos almacenados en diversos orígenes de información. Se han mejorado estas interfaces para admitir de forma única la funcionalidad apropiada de acceso a cada origen de datos. Las interfaces OLE DB permiten a un almacén de datos individual exponer fácilmente su funcionalidad nativa. Las interfaces soportan las posibilidades de manejo de datos de cada tipo de fuente de datos, en Visual Basic solo se puede acceder mediante ADO.

TABSTRIP: Con el control Tabstrip puede definir múltiples páginas en la misma área de una ventana o un cuadro de diálogo de una aplicación. El control consta de uno o más objetos TAB de una colección TABS. En tiempo de diseño y en tiempo de ejecución puede modificar la apariencia

³⁰ Reselman Bob, Peasley Richar, Descubre Visual Basic 6, Ed. Prentice Hall, España 1999, pps.517, 518

de los objetos TAB si establece sus propiedades y además, en tiempo de ejecución, si invoca métodos para agregar y quitar objetos TAB.

4.3.2 PROTOTIPO DEL "MÓDULO DEL PROFESOR"

De acuerdo al diagrama de flujo descrito anteriormente se procedió a crear la siguiente interfaz, La cuál consta de dos pantallas principales y una Base de Datos.

Las elementos que se tienen son:

- a) Pantalla de explicación
- b) Pantalla de Dudas.
- c) Base de datos que es la encargada de organizar, y guardar el conocimiento del experto

A continuación se describe brevemente cada una de las tres partes que componen el módulo del profesor.

a) Pantalla de Explicación.

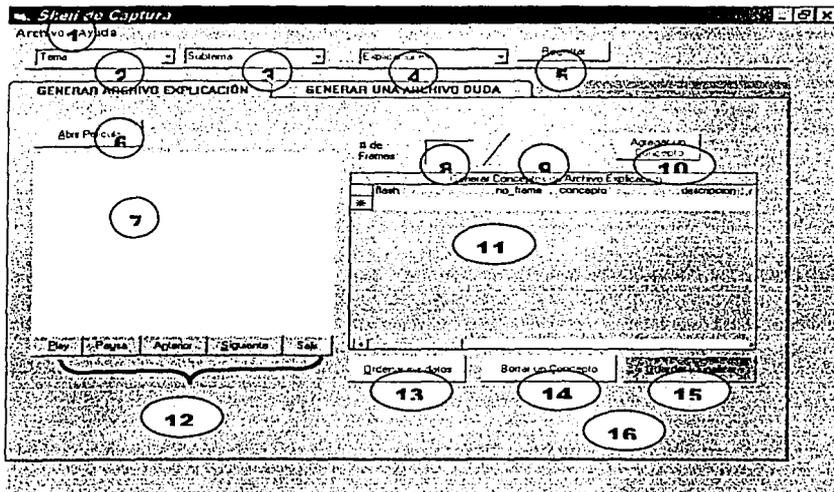


Figura 10. Pantalla de Explicación.

En general esta pantalla le permite al experto:

- Abrir, visualizar, controlar, ver el total de frames (total de cuadros de la película) y número de frame de su película Flash.
- Capturar sus textos (concepto y descripción) en base al tema, subtema, número de explicación y frame elegidos.
- Recuperar de la base de datos su información y película si es que existe en base a tema, subtema y explicación.
- Editar (Insertar, borrar y ordenar) sus textos en base al número de frame.
- Guardar su información en un archivo externo y en una base de datos (Tabla de Conceptos Maestra).

La pantalla que se muestra consta de los siguientes componentes:

1.- Se encuentran dos Menús:

Archivo:

Le da la opción al usuario de salir del sistema.

Ayuda:

Le da al experto una pequeña ayuda de cómo se generan su información ya sea para dudas o para una explicación.

2.- Caja de menú desplegable "Tema", le permite al experto elegir el tema a explicar o definir una duda.

3.- Caja de menú desplegable "Subtema", le permite al experto elegir el subtema a explicar en base al tema elegido anteriormente.

En el caja de menú desplegable "Explicación" le permite al experto elegir una explicación.

4.- El experto puede generar hasta cuatro explicaciones por tema y subtema.

5.- Botón "Registrar" le permite al profesor realizar una búsqueda en la base de datos en base al tema, subtema y explicación, ó bien tema, subtema y duda.

6.- Botón "Abrir" le permite al experto Abrir el archivo de su película Flash que desea visualizar.

7.- Un cuadro donde se visualiza la película Flash.

8.- Un cuadro que visualiza el frame actual.

9.- Un cuadro que indica el total de frames que tiene la película.

10.- Botón "Agregar un concepto", este botón le permite al experto agregar un concepto en la tabla "Generar Conceptos de Archivo Explicación", se agrega en base al número de frame, tema, subtema y Número de explicación elegida.

11.- Una tabla "Generar Conceptos de Archivo Explicación", el cuál le permite al experto capturar sus textos en base al número de frame de la película que se encuentra visualizando en ese momento, así como su descripción de su concepto importante.

Esta tabla es un objeto de Visual Basic (DATAGRID), que permite visualizar los atributos de una tabla(Entidad de la Base de datos) contenida en una Base de datos. Además de la manipulación de este objeto.

Los campos que se visualizan provienen de la tabla ConceptoT la cual se describirá a detalle en la parte de Base de datos.

12.- Una serie de botones consecutivos que le permiten al usuario manipular su película Flash en base a Reproducción(Play), Seguir(Siguiente), detener completamente(Salir), pausa(Pausa), Regresar(Anterior).

13.- Botón "Ordenar datos" el que le permite al experto ordenar su información en base al número de frame.

14.- Botón "Borrar Concepto" el cuál le permite al experto borrar un concepto, este borra toda la línea donde se encuentra el concepto.

15.- Botón "Guardar y Terminar" el que permite al experto Guardar sus datos en la base de datos, y en una carpeta de textos donde se guarda un archivo externo en base al numero de tema, subtema y explicación.

16.- En la parte de Explicación se observa la pestaña "Genera Archivo Explicación" , que contiene los componentes antes descritos.

El código de las acciones que realizan los componentes descritos anteriormente se mostrará en el Apéndice B, No.1.

b) Pantalla de Dudas.

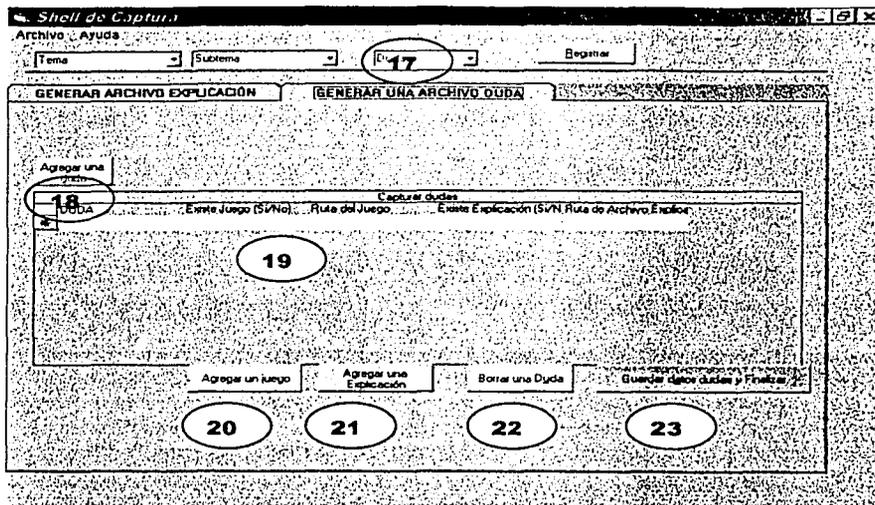


Figura 11. Pantalla de Dudas

b) Pantalla de Dudas

En general esta pantalla le permite al experto:

- Recuperar dudas existentes en base al tema y subtema.
- Editar (Insertar, borrar y actualizar) información en la tabla "Captura dudas".
- Guardar su información en un archivo externo y en la base de datos.
- Señalar los materiales (Juego o explicación) para resolver una duda.

A continuación se describe cada uno de sus componentes:

18.- Botón "Agregar Duda" el que le permite al experto capturar una duda a la vez en base al tema, subtema y duda por cada renglón de la tabla.

19.- Tabla "Captura dudas" .

Esta tabla es un objeto de Visual Basic llamado DATAGRID el cual permite manipular y visualizar los datos de los atributos que provienen de la tabla dudaT que esta contenida en la Base de Datos, los atributos o campos se describen en la siguiente sección.

Esta tabla le permite al experto capturar en forma de texto las dudas que se le presentan frecuentemente a sus alumnos en la clase. Así como señalar los materiales que deben ser llamados para resolver dicha duda.

20.- Botón "Agregar Juego" permite que el experto pueda insertar la ruta de un archivo de juego en la columna ruta juego de su tabla "Captura dudas".

21.- Botón "Agregar una explicación" permite que el experto pueda insertar la ruta de su archivo explicación en la columna de ruta explicación de la tabla "Captura dudas" el cual fue generado con anterioridad en la pestaña "Generar archivo Explicación".

22.- Botón "Borrar una duda" permite que el experto pueda borrar todo el renglón de su duda en la tabla "Captura dudas".

23.- Botón "Generar Archivo y Finalizar" le permite al experto guardar su información en una base de datos(Tabla de Dudas Maestra) y en un archivo de texto en base al tema, subtema y duda.

El código de las acciones que realizan los componentes descritos anteriormente se mostrará en el Apéndice B, No.1.

ORGANIZACIÓN DE LA INFORMACIÓN.

Cabe señalar que el profesor puede guardar su información en una base de datos y en un archivo externo.

La finalidad de guardarla en:

- Base de datos es:

Recuperar la información si existe en la base de datos y así poder modificarla, ingresar nueva, y guardarla. Si no existe crearla y guardarla.

Esta información se recupera en base al tema y subtema y si es explicación o duda.

- Archivo Externo

Se creó con el fin de que el módulo "Cascajón del Tutor" busque e identifique fácilmente dichos archivos ya sea de duda o explicación.

Su nombre es muy importante porque es una clave especial, puesto que hace referencia al tipo de archivo, al tema y subtema. Estos archivos se guardan y se pueden localizar en la ruta c:\proyecto\textos.

Por ejemplo:

Archivo Explicación: C001001E001

Archivo Duda: D001001D001

En este ejemplo se muestran las claves donde es posible percibir que cambia el primer carácter, así como el séptimo. La C y E indica que se trata de un archivo Explicación. La D representa un archivo Duda.

Los tres primeros dígitos de ambas claves de archivos hacen referencia al tema, los segundos tres dígitos indican el subtema, y los tres últimos indican el número de explicación o duda.

Tanto los archivos como la base de datos contienen la información del profesor de manera clasificada y ordenada en base al tema y subtema. La cuál puede ser recuperada de la Base de Datos en el momento que el experto desee.

c) Base de datos.

En esta parte se creó una base de datos, consta de cuatro tablas principales.

Dos de estas tablas se utilizan en la pantalla **Explicación**, y dos en la pantalla de **Dudas**. Dos de estas tablas se les señaló con una T al

final de su nombre, para indicar que es una tabla temporal. Las otras dos contienen una M que indica que es una tabla Maestra.

Se le llamó "Tabla Maestra" porque es la que contiene la información en forma permanente cuando el experto guarda su información.

Se le llamo "Tabla Temporal" porque cuando el experto realiza una búsqueda de información relacionada a un tema y subtema, esta se carga de la tabla maestra a la tabla temporal si es que existe dicha información.

A la vez la información de esta tabla se visualiza en el objeto de Visual Basic DATAGRID el cual es llamado "Captura dudas" o "Generar Archivo Explicación" en que el profesor puede modificar, ordenar e ingresar información.

Esta información permanece vigente en esta tabla hasta que el usuario no decida guardar sus datos, los cual se almacenaran directamente en la tabla Maestra.

Cuando se trabaja por primera vez en un tema, subtema, explicación o duda, el experto esta trabajando en la tabla Temporal.

Con esto se tiene la ventaja de evitar redundancia e inconsistencia de datos en la tabla Maestra, el experto puede modificar y actualizar la información, si así lo requiere.

Para la pantalla de **Explicación** se trabaja en las tablas llamadas "ConceptoT" y "ConceptoM". Al igual que en el caso anterior, se dispuso llamarles así porque la tabla ConceptoT es la tabla temporal y la otra es una tabla Maestra.

En ambas tablas se tienen los mismos campos y éstos se llaman igual.

Los campos de la tabla ConceptoT y ConceptoM son los siguientes.

Campo	Tipo	Descripción
No_de_tema	Texto	Indica el número de tema, subtema y no de explicación en clave: 001001E001 Los primeros 3 dígitos, indican tema, los otros 3 siguientes subtema, E de explicación y los últimos tres al número de explicación.
Flash	Texto	Nombre del archivo de Película Flash
No_frame	Númerico	Indica el cuadro actual de la película Flash.
Concepto	Texto	Indica que el experto debe capturar su información en base al campo no_de_frame.
descripcion	Texto	Indica que el experto debe capturar su información en base al campo concepto

Tabla 3 . Campos de las Tablas ConceptoT y ConceptoM.

En la parte de **Dudas** se trabaja en base a dos tablas llamadas "dudaT" y "dudaM" al igual que las de explicación una es temporal y maestra.

Estas tablas contienen los mismos campos y son los siguientes:

Campo	Tipo	Descripción
Clv_duda	Texto	Indica la clave en base a tema, subtema y duda. Contiene nueve caracteres, los primeros 3 indican el tema, los tres siguientes subtema, el séptimo es una D de duda y los últimos tres el número de duda.
Duda	Texto	El experto puede capturar su duda.
ejuego	Texto	El experto indica en forma de texto en forma de Si o No tiene material de juego que será llamado para resolver dicha duda.
Tipo_juego	Númerico	Indica si el archivo es una Animación Flash, una imagen, video, etc. De acuerdo a la extensión el sistema le asigna un número.
Nombre_juego	Texto	El experto busca e indica la ruta y nombre de su archivo Animación.
Eexplicación	Texto	El experto indica si tiene o no un archivo explicación para explicar la duda.
explicación	Texto	El experto busca e indica la ruta y nombre de su archivo Explicación.

Tabla 4 . Campos de las Tablas DudaT y DudaM.

Para conectar las tablas de esta base de datos creada en Access con Visual Basic se utilizó el componente ADO desde Visual Basic como sigue:

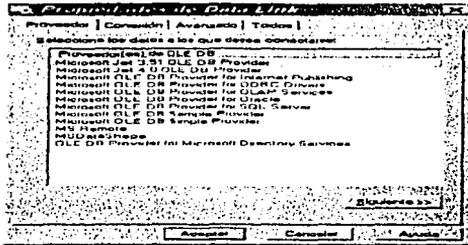


Figura 12. Conexión de Visual Basic y Access mediante ADO.

En este caso el proveedor OLE DB(Object Linking and Embedding for Data Bases/Vinculación e Incrustación de Objetos para Bases de Datos) que se utilizó es Microsoft Jet 3.51 OLE DB dado que la base de datos es de Access.

Para conectar cada una de las tablas Conceptos y Dudas donde el experto captura su conocimiento y además señala sus materiales, se utilizó un control Microsoft DataGrid Control 6.0 de los componentes de Visual Basic, el cual se enlazó con el objeto ADO descrito con anterioridad, a continuación se muestra la conexión en la siguiente figura:

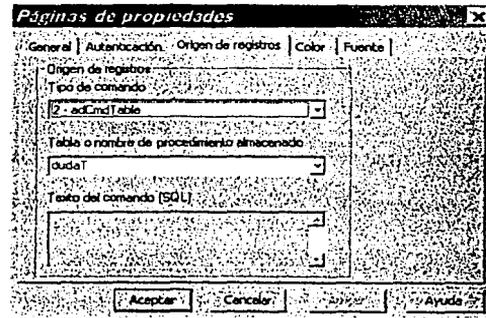


Figura 13. Alta de tablas en DataGrid.

Mediante la Creación de esta Base de Datos el experto puede:

- Capturar, recuperar y Guardar su información en el momento que lo desee.

Finalmente podemos decir que este módulo le permite al experto:

- Generar sus propios materiales.
- Adquisición de conocimiento en forma automática.
- Verter su información sin depender directamente del equipo de desarrollo.
- Agilizar la producción de software educativo.

4.4 CASCARÓN DEL TUTOR.

El objetivo general establecido para la realización de esta parte del trabajo, consistió en desarrollar un cascarón que permitiera:

- Generar sistemas tutoriales inteligentes bajo un esquema probado.
- Recuperar la información capturada en el Módulo del profesor.
- Presentarle al alumno la información de manera coordinada.
- Facilitar al desarrollo de materiales multimedia bajo la supervisión de alumnos, profesores y equipo de desarrollo.

Por lo anterior, se diseñó el módulo "Cascarón del tutor" que permite interactuar con el alumno, de manera que identifica sus deficiencias y pueda subsanarlas de acuerdo a sus necesidades.

Para mostrar la información necesitamos que este cascarón integre de manera automática diferentes materiales multimedia, y los presente al alumno de manera íntegra y coherente.

Este módulo permite cambiar el contenido del tutor ya sea por actualizaciones, ampliaciones o inclusive por tratar un tema nuevo.

En el CCADET se ha desarrollado un esquema para el desarrollo de software educativo centrado en el alumno, de manera que este tipo de software le permite al alumno alcanzar su meta, que es aprender, promoviendo su reflexión y análisis sobre fenómenos particulares permitiéndole reparar o reafirmar sus propias creencias hasta llegar a una verdadera comprensión de los conceptos presentados. Además de

cubrir la meta principal del alumno, este software debe ser atractivo y de fácil manejo, de tal forma que el alumno lo pueda utilizar tantas veces como sea necesario sin perder su atención.

Este esquema ha sido aplicado en el desarrollo de muchos productos y sus resultados son muy alentadores en términos de la calidad de los productos, por lo que el diseño de nuestro cascarón del tutor se basa en este esquema probado que a continuación describimos.

Este cascarón del tutor consta de dos partes:

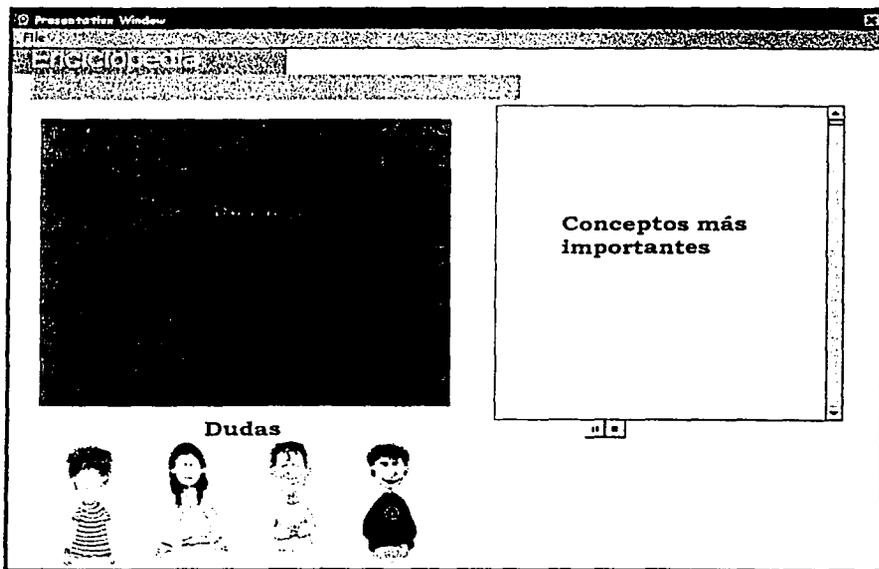


Figura 14. Pantalla del "Cascarón del Tutor".

- Explicación

En esta parte se le presenta al alumno un tema en particular de manera que todos los conceptos sean explicados de forma detallada,

coherente y fácil de comprender, como lo haría un profesor en el salón de clase.

Conforme se explica el tema, el sistema hace énfasis en los conceptos más importantes que requieren especial atención por parte de los alumnos.

Es importante que el alumno se sienta cómodo al revisar cada uno de los temas y que al igual que en un aula, pueda contar con compañeros que le ayuden en la comprensión de un tema cuando lo requiera. Es por ello que en este esquema se hace indispensable la presencia de compañeros virtuales que le ayudaran al alumno a interactuar con el sistema.

En esta parte de "Explicación" se muestran a los compañeros virtuales como compañeros pasivos, es decir, se encuentran escuchando la clase sin interactuar con el alumno ni con el contenido del tema. Nuestros compañeros virtuales son cuatro: Chuy, Ana, Benja y Lalo.

- Dudas

En esta parte cada compañero virtual le muestra una duda al alumno, y la manera en que puede resolverla. Estas dudas son proporcionadas por los expertos de la materia en cuestión. Su experiencia en impartir clases, les permite detectar las dudas que presentan los alumnos, referentes a un tema en específico, y la forma en que se le puede explicar nuevamente un concepto no comprendido. Los compañeros virtuales sirven como intermediarios entre el sistema y el alumno, gracias a ellos, los alumnos pueden elegir la forma de resolver sus dudas, ya sea por medio de un juego o mediante una nueva explicación. Estas opciones le permiten al alumno guiar su propio conocimiento.

4.4.1 HERRAMIENTAS DE DESARROLLO.

Las características del cascarón nos llevaron a elegir una herramienta multimedia que permitiera una fácil comunicación con los materiales multimedia creados en software Flash, además de ser flexible y proporcionarnos diversos servicios como búsqueda de texto, sincronización de medios etc .En base a estos parámetros es que se eligió la herramienta Authorware.

Authorware es un programa orientado a objetos para desarrollar programas de tipo multimedia, especialmente pensado para desarrollar manuales, enciclopedias interactivas entre otros programas y que permite combinar imágenes, sonido, animaciones digitales y video.

Conforme se reunía la información para el desarrollo de este módulo, también se puso en marcha el diseño e implementación del mismo. Los prototipos que surgieron fueron evaluados y modificados por los expertos conforme a las observaciones que éstos proporcionaban.

4.4.2 PROTOTIPO DEL MÓDULO CASCARÓN DEL TUTOR.

Dada las características del esquema probado anteriormente, y la necesidad de desarrollar sistemas tutoriales inteligentes, desarrollamos un módulo llamado "**Cascarón del Tutor**", la cual, nos permite recuperar la información que el profesor captura en el "Módulo del profesor" y presentarla al alumno de manera organizada. También el profesor se ve beneficiado al visualizar la información, ya que de esta manera, el profesor puede observar su propio tutor y modificarlo, si así lo requiere.

Es importante hacer notar que el profesor desarrolla y corrige su propio tutor, sin necesidad de estar en contacto con los programadores, lo que permite acortar el tiempo de desarrollo.

El módulo "Cascarón del tutor" funciona de la siguiente manera:

Para revisar un tema en específico, el cascarón requiere el número de tema y subtema que el usuario debe proporcionar.

EXPLICACIÓN

Del archivo de texto generado por el "Módulo del profesor", el cascarón lee la información línea por línea que contiene dicho archivo, la primera línea contiene el nombre del archivo de la animación flash que se presentará para explicar un tema en específico y lo despliega en pantalla.

Durante la presentación de la animación, el cascarón obtiene cada uno de los conceptos importantes que se encuentran en el archivo de texto del tema que se están estudiando y los muestra al alumno al mismo tiempo que la animación hace referencia a cada uno de los conceptos.

Una vez concluida la explicación, el cascarón pasa a la parte de Dudas.

El código de programación de la parte de explicación se muestra en el apéndice B, No.2.

DUDAS

Del archivo de texto externo generado por el "Módulo del profesor", el cascarón obtiene el número total de dudas que se presentarán, así como las respuestas a cada una de ellas. Para asignar las dudas, se eligen al azar a uno de los compañeros virtuales, se le asigna una duda y la manera en que se puede resolver. Todos los personajes pueden mostrar una duda, aunque esto dependerá del número de dudas que tengamos disponibles para el tema que se esté estudiando. El módulo "Cascarón del tutor" acepta de una a cuatro dudas.

El alumno puede seleccionar la duda que más le interese y se identifique, así como la forma de responderla.

Existen dos formas de responder a una duda:

- Por medio de un juego (Tú) se le presenta al alumno un juego en flash que le permitirá interactuar con el sistema para aclarar su duda.
- Por medio del sistema (sistema) se le presenta al alumno una animación en flash que le explica nuevamente el concepto no comprendido.

El alumno puede seleccionar todas las dudas existentes para un tema en específico, y responderla como él lo desee.

En base a estas ideas fue posible especificar, depurar e implementar el siguiente algoritmo de funcionamiento.

4.4.3 ALGORITMO DEL MÓDULO "CASCARÓN DEL TUTOR".

1. Obtener el número de tema y subtema proporcionado por el usuario
2. Leer archivo de texto externo correspondiente al tema subtema.
3. Obtener datos (nombre de la animación flash a presentar, número total de frames que contiene dicha animación, nombre del archivo de texto que hace referencia a las dudas)

4. Desplegar animación flash.
5. Recorrer todas las líneas del archivo de texto para obtener los conceptos importantes que se van a desplegar en cada frame.
6. Comparar si el frame del archivo de texto es igual al frame de la animación flash que se está desplegando o no.
Si son iguales se despliega el concepto.
Si no son iguales no se realiza sigue comparando los frames
7. Comparar si el número total de frames que se obtuvo en el archivo de texto es igual al número total de frames de la animación flash.
Si son iguales Pasa a la parte de dudas.
8. Abrir archivo de texto externo referente a las dudas.
9. Obtener el número total de dudas existentes para el tema en cuestión.
10. Elegir aleatoriamente un personaje multimedia.
11. Obtener duda del archivo de texto externo.

12. Asignar la duda al personaje elegido.
13. Verificar por cada personaje, si se le asignó una duda o no.
14. Obtener la respuesta a cada una de las dudas asignadas.
15. Desplegar dudas
16. Verificar selección de la duda.
Desplegar forma de responder a una duda.
17. Verificar selección de respuesta.
Desplegar juego.
Desplegar animación.

El diagrama de flujo de este algoritmo se puede observar en el apéndice A, Figura 2, 3, y 4.

4.5 MÓDULO DEMOSTRATIVO DE JUEGOS DIDÁCTICOS Y MATERIAL MULTIMEDIA.

El objetivo primordial de trabajar con juegos es verificar que sean efectivos en su labor de capturar la atención del usuario; es decir, que introduzcan al alumno en el estudio de los temas de forma atractiva, amena, y, que al mismo tiempo, evite que tenga presente que va a ser evaluado. Además de confrontar al alumno con las creencias (ideas previas) que tiene, y utilizarlas de diferentes maneras:

En nuestro caso, se decidió trabajar un solo tema, de modo que nos permitiera centrar los esfuerzos para garantizar una interfaz de calidad. El tema seleccionado fue el de " Los Sistemas de coordenadas".

El texto en el que nos basamos para el desarrollo, fue elaborado por el Dr. Fermín Viniegra Heberlein. Este texto se puede ver en el Apéndice C.

A través de reuniones periódicas con el Dr. Viniegra, se analizó el texto y se identificaron los conceptos fundamentales del mismo para elaborar lo que posteriormente se convertiría en el guión que se utilizó en el desarrollo del material didáctico.

Se nos brindaron tres dudas que son las más comunes al tratarse el tema en el salón de clase, que, al parecer del experto, debían presentarse en los alumnos virtuales

En base al guión elaborado previamente, se realizó una animación en Flash de dicho tema, se grabó el audio correspondiente, y se les dio especial énfasis a los conceptos importantes del texto.

Las respuestas correspondientes a las dudas se darán por dos distintos medios:

- 1 Juego
- 1 Nueva explicación, La nueva explicación a su vez contiene:
 - Animación.
 - Audio
 - Conceptos importantes.

Los juegos consisten de un sistema experto integrado a un sistema multimedia que contiene los elementos teóricos y prácticos de un tema de física de nivel medio superior.

Los dos juegos que se plantearon presentan dos características importantes: permiten al estudiante cometer errores cuyas consecuencias no son amenazadoras y, permiten al estudiante ejercer cierto control sobre el proceso de aprendizaje. Estos juegos son: "Mi Pueblito" y "Morita la Pulguita".

El juego interactivo "Mi Pueblito" se ideó basándonos en la difundida afición a los llamados videojuegos de la mayoría de los adolescentes desde hace ya, algún tiempo. Y tomando en cuenta algunas consideraciones fundamentales que se consideran en el desarrollo de Software Educativo Centrado en el Estudiante, tales como: Cuál es el objetivo de nuestros usuarios? Cuáles son las características de su tarea? Que busca nuestro usuario? Que necesita para alcanzar su meta?

Buscamos en todo momento que el juego fuera útil (Que integre las funciones necesarias para el usuario, que haga exactamente lo que se pensó que hiciera), que fuese utilizable (se organizara y diseñara de tal forma que el usuario fuese capaz de entender su estructura y cómo trabaja) y que promueva un profundo entendimiento (es decir que el software debe ser una guía que no solamente repita los contenidos de los libros, sino que además, ofrezca al estudiante los elementos (ejercicios, experimentos y metodologías) que lleven a una reflexión sobre un fenómeno en particular y su abstracción, mientras que analiza y revisa sus propias consideraciones, guiándolo a un verdadero entendimiento del material expuesto).

Para construir el juego interactivo "Mi Pueblito" se plantearon el siguiente algoritmo y diagrama de flujo:

4.5.1 ALGORITMO DEL JUEGO INTERACTIVO "MI PUEBLITO".

1. Establecer origen cursor.
2. Lanzar primera pregunta. Ejemplo: ¿Dónde está la casa de la tía Lencha?
3. Introducir respuesta del usuario a las coordenadas de posición x, y con respecto a la figura que corresponda la pregunta anterior.
4. Registrar respuesta del usuario.
5. Comparar respuesta del usuario contra la del sistema.

Si Respuesta es correcta entonces:

Muestra mensaje
Incrementa Respuesta Correcta
Lanza la siguiente pregunta.

Si No

Muestra mensaje
Incrementa Respuesta Incorrecta
Lanza la siguiente pregunta.

6.- Compara si el número de preguntas lanzadas es igual al número de preguntas totales del juego.

Si son iguales entonces:

Compara Respuestas del Usuario Correctas e Incorrectas.

Si Respuestas Correctas son mayores que las incorrectas entonces:

Muestra mensaje "Muy Bien".
Si No

Muestra mensaje "Estas seguro(a) que no quieres preguntar al Profesor".

Si No

Regresa al punto 5.

El diagrama de flujo de este pseudocódigo se muestra en el apéndice A, Figura 5.

4.5.2 ALGORITMO DEL JUEGO INTERACTIVO "MORITA LA PULGUITA"

- 1.-Lanza pregunta.
- 2.- El usuario deberá escoger combinación de imágenes para dar solución.
- 3.- Compara la solución del usuario con la del sistema.
 - Si la Respuesta del usuario es correcta entonces:
 - Lanza mensaje de que encontró la respuesta correcta.
 - Termina Juego.
 - Si No
 - Muestra una sugerencia.
 - Regresa al paso 2.

El diagrama de flujo de este pseudocódigo se muestra en el apéndice A, Figura 6

4.5.3 HERRAMIENTAS DE DESARROLLO.

Decidimos utilizar el siguiente software de desarrollo:

Macromedia Flash 5.0.

Flash es una tecnología desarrollada por Macromedia, que nos permite crear y reproducir contenido multimedia basado en vectores. Gracias a esta característica, el texto puede utilizar cualquier tipo de fuente y determinadas animaciones se reproducen en menos tiempo del habitual. Además Flash permite comprimir audio y, en caso que se requiera, utilizarlo en la web.

Para reproducir estas animaciones se necesita el reproductor Macromedia Flash Player, el cuál es gratuito.

4.5.4 PROTOTIPOS DE JUEGOS DIDÁCTICOS.

De acuerdo al diagrama de flujo descrito anteriormente se procedió a crear la siguiente interfaz.

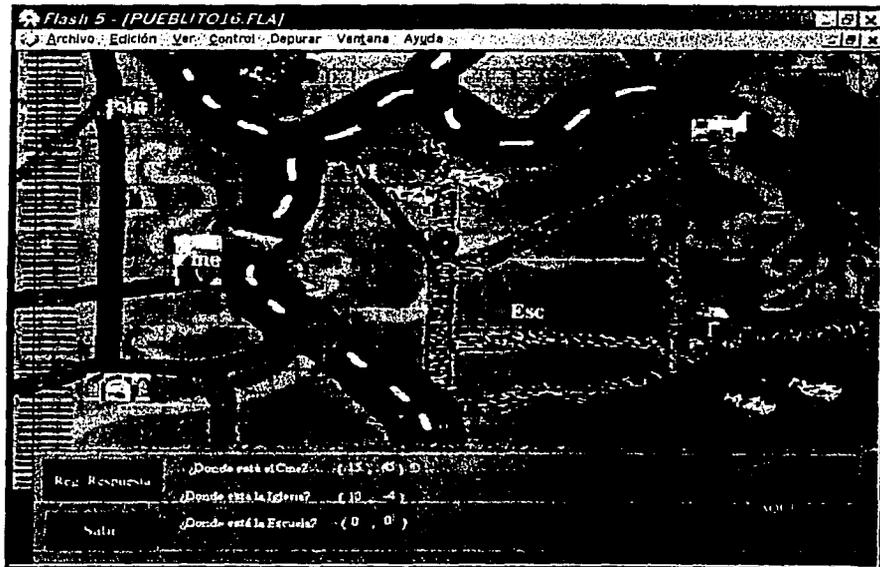


Figura 15. Pantalla de "Mi Pueblito"

PANTALLA DE JUEGO DE "MI PUEBLITO".

En esta pantalla se presenta el mapa de un lugar llamado "Mi Pueblito", donde se encuentran los objetivos; el Cine, la Tiendita, la Escuela, la Casa de la Tía Lencha, la Panadería, la Iglesia y el Mercado. Ahí podemos encontrar nuestro origen de referencia, una esfera que es posible mover por medio de los botones de arriba, abajo, derecha e izquierda, hacia cualquier dirección del mapa. Cuando el usuario oprime el botón de "Aquí", en ese momento se toma su posición como nuestro origen de coordenadas (0,0) , y aparece la

primera de 7 preguntas que se le harán al alumno, solicitándole la posición de los objetivos definidos anteriormente.

Se captura la respuesta del alumno en un par de coordenadas, e inmediatamente aparece una marca azul si ésta es correcta y una roja

en caso contrario. Al finalizar, se hace una comparación de el número de respuestas correctas e incorrectas y aparece un mensaje de "Muy Bien" en caso que haya logrado más respuestas correctas y de "Estas seguro que no quieres preguntarle al profe?" en el otro caso.

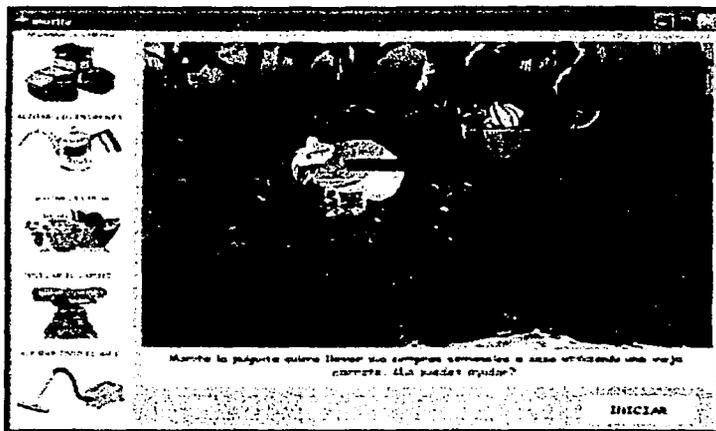


Figura 16. Pantalla de "Morita la Pulguita"

PANTALLA DE JUEGO DE "MORITA LA PULGUITA"

En esta pantalla encontramos una situación inicial, en donde encontramos a "Morita La Pulguita" junto a la carreta que se desea mover. Al lado izquierdo de la pantalla, encontramos varios elementos que el alumno puede elegir para ayudar a Morita a lograr su objetivo. De acuerdo a ésta elección el alumno puede ver algunas animaciones, y al mismo tiempo aparecen mensajes que orientan al alumno sobre

la dirección que debe seguir, hasta que finalmente se logre mover la carreta de nuestro personaje.

A través de la experiencia en el desarrollo de estos materiales se logró identificar 5 fases fundamentales de una metodología:

1.- Especificación de Estrategias y Metas Pedagógicas: Es decir, qué queremos que los estudiantes aprendan y entiendan? Para que? ¿En que contexto?

2.-Análisis de la situación de trabajo de los estudiantes: Cuáles son las creencias, ideas generales que los estudiantes tienen sobre el material que los profesores quieren que aprendan? Tienen experiencias personales relacionadas a esos temas? Pueden las estrategias pedagógicas elegidas trabajar exitosamente sin la computadora? Cuando los estudiantes están trabajando, que clase de herramientas utilizan? Que tipo de información o soporte están buscando? Cuando? Porqué?

3.- Análisis de Requerimientos: (Selección de la mejor suite de tecnología): Entre los distintos paradigmas de tecnología disponibles, cuál nos da el mejor resultado y es el más apropiado para nuestras necesidades prácticas y pedagógicas?

4.- Implementación de Estrategias Pedagógicas en el Software Educativo: Observación y Análisis de los estudiantes al utilizar el software. Prototipos, Pruebas y Refinamiento, que tipo de problemas prácticos y educacionales presenta el prototipo desarrollado.

5.- Comparación de Resultados y Evaluación: Se alcanzaron los objetivos pedagógicos? Si o no y porqué?

El código de programación de estos prototipos de juegos interactivos multimedia se encuentra en el apéndice B. No. 3 y 4.

5. PRUEBAS.

Teniendo los archivos que se generaron en la parte de Generación de Materiales Multimedia, se eligió un miembro del laboratorio, el Dr. Fernando Gamboa Rodríguez, para probar los módulos. El usuario procedió a manejar los módulos, mismos que le permitieron vaciar conocimiento en forma de texto sobre el tema "Sistemas de Coordenadas". Además de la sincronización de la explicación con la animación desarrollada para ese tema, se integraron las dudas frecuentes de los alumnos, y se señalaron los materiales para ser llamados para resolver dichas dudas.

A continuación se muestra y describe el uso del "Módulo del profesor" y el módulo "Cascarón del tutor" en el tema "Sistema de Coordenadas".

5.1.- CAPTURA DE CONOCIMIENTO.

a) Explicación.

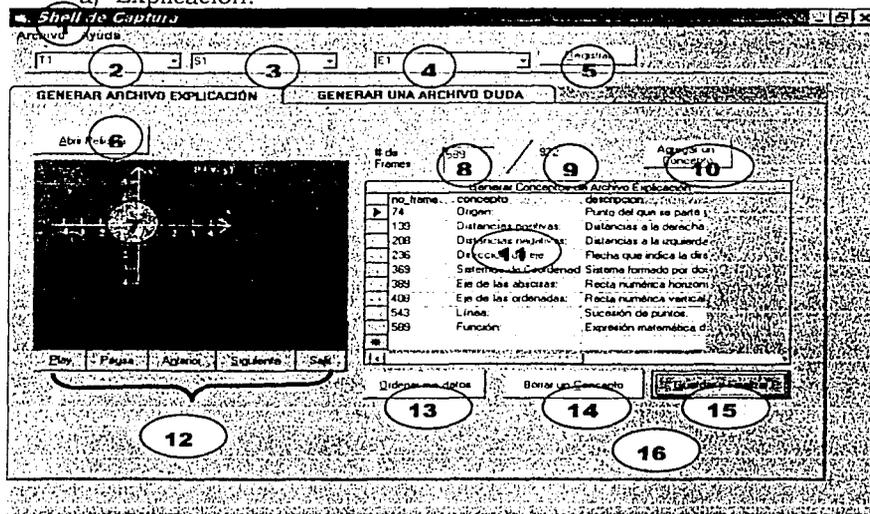


Figura 17. Pantalla de Explicación

En esta parte de la pantalla el profesor eligió en número de tema, el Tema uno, en el subtema uno, y explicación uno. Ver Figura 17, cuadros desplegables(2,3,4).

El botón “Registrar” (No. 11) Figura 17, le permite recuperar de la base de datos información previa, si ésta existe, sobre el tema, subtema y explicación seleccionado. En su caso el usuario no tenía información, por lo que solo se mostró vacía la tabla “Generar Conceptos de Archivo Explicación”.

Con el botón “abrir película” (No.6) Figura 17, se abrió una animación flash que explica el tema seleccionado, la cuál se visualizó en el cuadro de película(No. 7) Figura17.

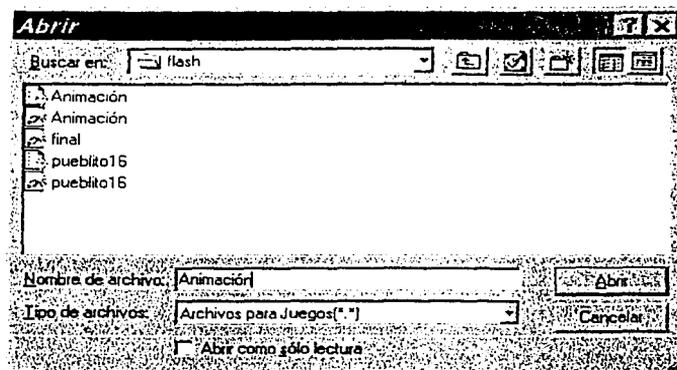


Figura 18. Elegir un archivo del Lenguaje Flash.

El experto pudo manipular esta animación flash con los botones Play, Pausa, Pausa, Anterior, siguiente.(No.12) Figura 17, los cuales le ayudaron a visualizar de manera detallada dicha animación, ya que la pudo observar cuadro por cuadro, parar e iniciar la animación cuantas veces lo requiera; además le permitió identificar los momentos en los cuales quería hacer referencia a un concepto, el cual fue identificado en el guión previo elaborado por el mismo experto.

En el cuadro de texto “# de frame” (No.8) Figura 17 se muestra el número de cuadro que se está mostrando en ese momento y en la

etiqueta (No.9), Figura 17, se mostró el número total de cuadros que contiene la animación flash.

En la zona de captura de información (No.11) Figura 17, el experto proporcionó la información necesaria para generar su tutorial como es número de frame, concepto y descripción.

Cuando el experto capturó toda la información requerida, la pantalla se observaba como se muestra a continuación en la figura 19.

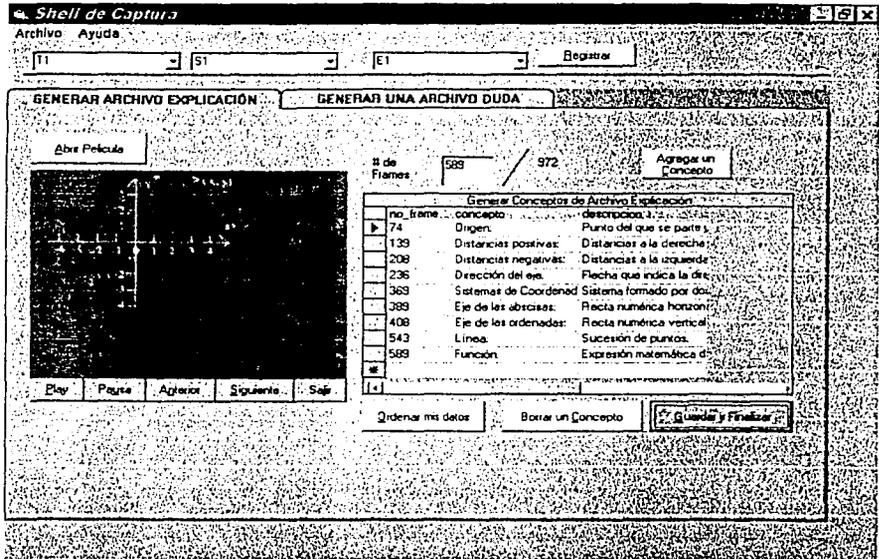


Figura 19. Pantalla de Explicación con información.

Finalmente, como la información capturada estaba ordenada por número de cuadro, el experto guardó su información para este tema. Esta información se guarda directamente cuando elige el botón "Guardar y Finalizar"(No.15) Figura 17, en la ruta c:\proyecto\textos, con el nombre que lo identifica, en este caso se trataba de un archivo explicación del tema que

eligió(Tema1, subtema1, Explicación 1), por lo que el nombre del archivo quedó como sigue: "C001001E001.txt".

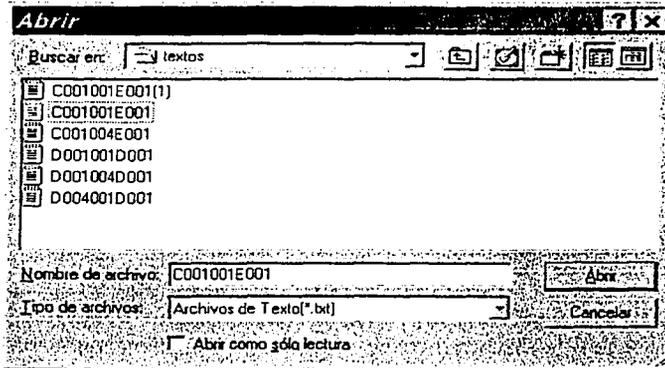


Figura 20. Guardar la información capturada en la pantalla Explicación.

b) Dudas

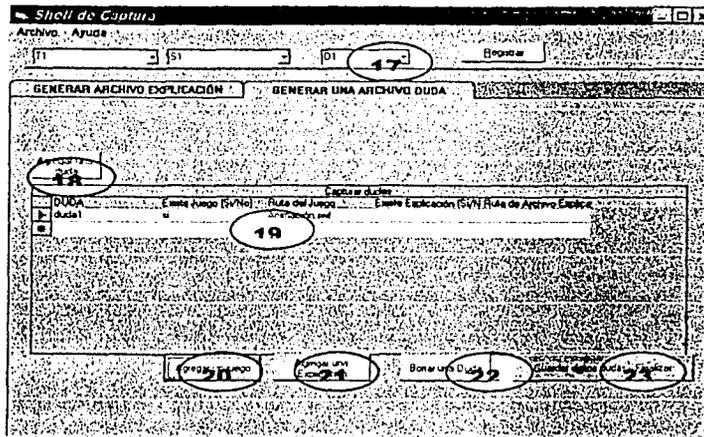


Figura 21. Pantalla de Dudas con información.

Para esta pantalla el experto eligió el tema uno en los cuadros de menú desplegable (No.2) Figura 21, el subtema uno, en (No.3) Figura 21, y la duda en (No. 17) Figura 21. Al igual que cuando eligió la pestaña "Generar Archivo Explicación" , y con ayuda del botón "Registrar"(No.5) de la Figura 17, el sistema nuevamente realizó una búsqueda en la base de datos para este tema, y subtema, no encontrándose nada.

Para cada duda, el experto eligió Agregar una duda, lo cual le daba un nuevo renglón donde le muestra la columna a ser llenada.

Así el usuario capturó sus dudas, y el lugar donde se encuentran los materiales a ser llamados para explicar cada una de ellas. Por ejemplo, para resolver una duda se puede elegir un archivo de explicación en el Botón "Agregar una explicación"(No.21) Figura 21, el cual le vuelva a explicar al alumno un concepto determinado como se muestra a continuación.

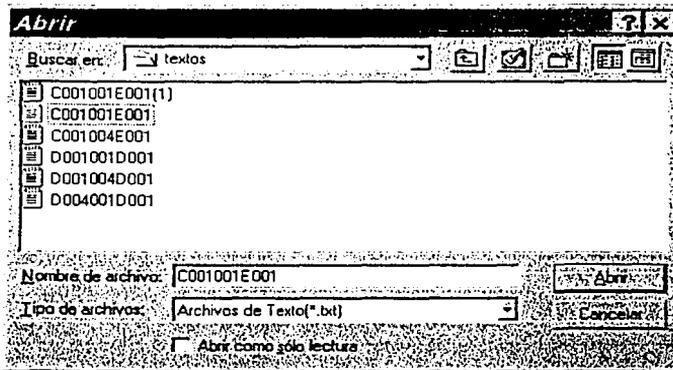


Figura 22. Abrir un archivo de texto para volver a explicar una duda.

La duda también se puede explicar mediante un juego interactivo. Este juego se puede elegir en el botón "Agregar Juego" (No.20) Figura 21, como se muestra a continuación.



Figura 23. Abrir un archivo de juego interactivo para volver a explicar una duda.

Cuando término de vaciar su información para este tema, la pantalla se observaba de la siguiente manera:

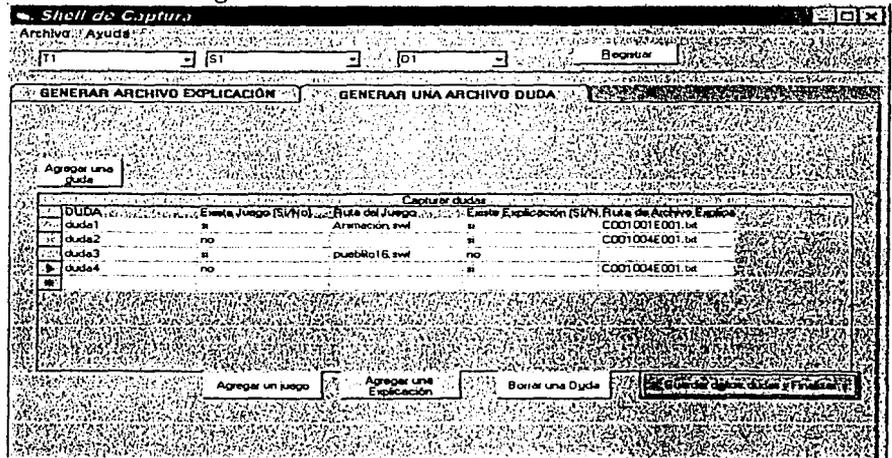


Figura 24. Pantalla de Dudas con información.

Finalmente el experto almacena su información. Los archivos que se generaron al almacenar dicha información serán utilizados por el módulo "Cascañón del tutor".

Cabe resaltar que encontró que hay que indicar la forma de guardar la información capturada, puesto que el módulo "Cascañón del tutor" requiere que los archivos generados por el "Módulo del profesor", sean guardados con el nombre que se le da por default, y no cambiarle el nombre, para conservar el orden de archivos en base al tema.

5.2.- INTEGRACIÓN DE LOS ARCHIVOS GENERADOS POR EL "MÓDULO DEL PROFESOR".

Para que el módulo "Cascañón del tutor" funcione, y le presente al profesor la información que capturo en el "Módulo del profesor", es indispensable que le proporcione el número de tema y subtema que desea revisar, para ello se le presenta una ventana en la cual debe de escribir dichos números como se muestra a continuación.

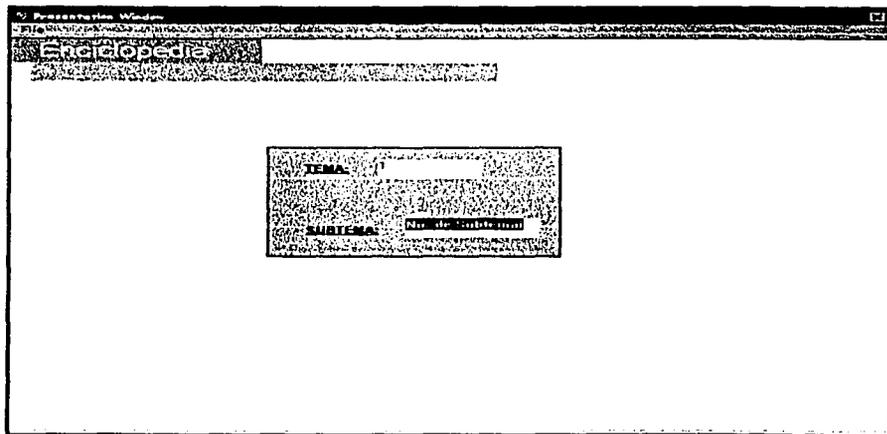


Figura 25. Elegir un tema y subtema del tutor.

Con los datos anteriores, el módulo “Cascarón del tutor” lee el archivo de texto generado por el “Módulo del profesor”, el cual contiene toda la información que el usuario observará de manera organizada.

En la pantalla que se muestra a continuación se puede observar la integración de la animación y la sincronización del texto explicación, generado con el “Módulo del profesor”.

The screenshot shows a software window titled "Presentation Window" with a standard Windows-style title bar. Inside the window, there is a central area on the left displaying a coordinate system with several points plotted. To the right of this area is a text box titled "Conceptos Importantes" (Important Concepts) containing the following definitions:

- Dirección del eje:** Flecha que indica la dirección en la que aumentan los valores.
- Sistemas de Coordenadas de dos dimensiones:** Sistema formado por dos ejes, perpendiculares y cuyos orígenes coinciden en el punto de intersección de ambas.
- Eje de las abscisas:** Recta numérica horizontal que se acostumbra denotar mediante la letra x .
- Eje de las ordenadas:** Recta numérica vertical que se acostumbra denotar mediante la letra y .
- Línea:** Sucesión de puntos.
- Función:** Expresión matemática de la forma $y=f(x)$.

Below the text box, there are two buttons: "Pausa" (Pause) and "Detener" (Stop). At the bottom of the window, there are four small, stylized human figures representing virtual companions, and the text "Compañeros virtuales" (Virtual companions) is displayed to their right.

Figura 26. Exposición del tema “Sistema de Coordenadas”.

Es esta parte de la pantalla se despliega la animación flash del tema “Sistemas de Coordenadas”. Al mismo tiempo se despliegan los conceptos más importantes que el profesor capturó en el “Módulo del profesor”. En la figura 26, podemos ver que en la parte de explicación se muestra el concepto de función y la forma en que se debe graficar; al mismo tiempo, en la parte de conceptos importantes, se da la definición de función. Aquí, los compañeros virtuales se encuentran de espaldas al usuario y no interactúan con el usuario ni con el sistema. Esto se debe a que en este momento ellos, al igual que el alumno, deben poner atención en lo que se les está explicando.

Los botones Pausa y Detener, permiten que el usuario pueda realizar un alto en la animación.

Después de concluir con la explicación del tema, el módulo pasa de manera automática a la parte de dudas.

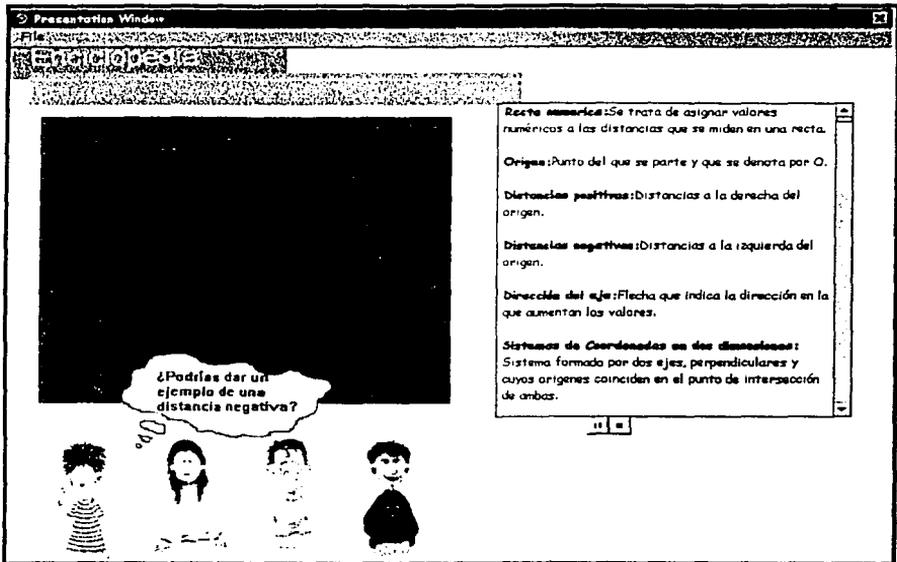


Figura 27. Dudas de los compañeros virtuales acerca del tema "Sistema de Coordenadas".

En la figura 27 se muestran las dudas que tienen cada uno de los personajes. Cada una de las dudas se visualizará cuando el alumno coloque el puntero del ratón sobre el personaje que él prefiera, en esta figura podemos ver que el compañero virtual denominado "Chuy" muestra la siguiente duda: "¿Podrías dar un ejemplo de una distancia negativa?".

Si el alumno decide responder la duda mostrada por el compañero virtual, debe dar un clic sobre la duda para visualizar la, o las formas de resolver dicha duda, como se muestra en la siguiente figura 28.

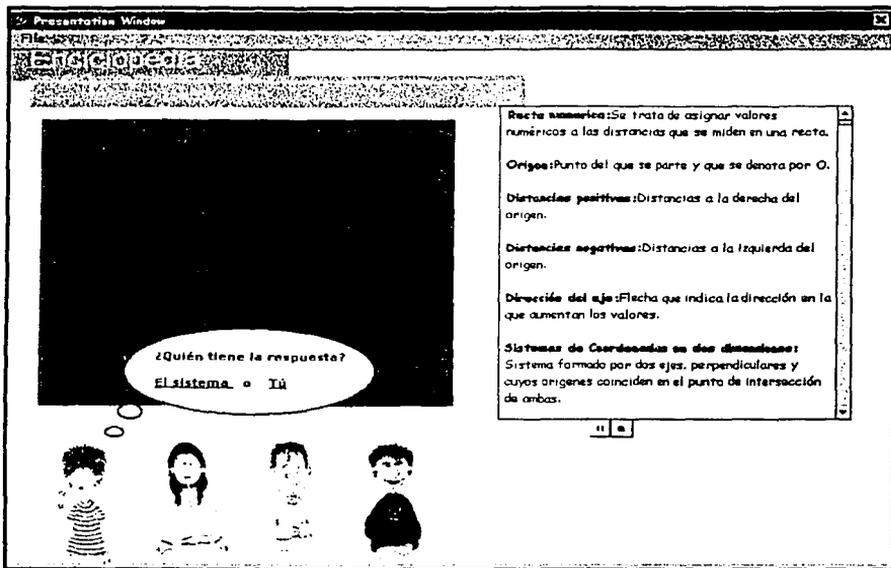


Figura 28. Opciones de responder la duda en los compañeros virtuales.

En esta pantalla se muestra las dos opciones para responder una duda.

- 1) El sistema ó
- 2) Tú

Si el alumno decide responder la duda por su cuenta (Tú) el módulo muestra un juego, en este caso es definido por el experto, llamado "Mi pueblito" como se muestra en la figura 29.

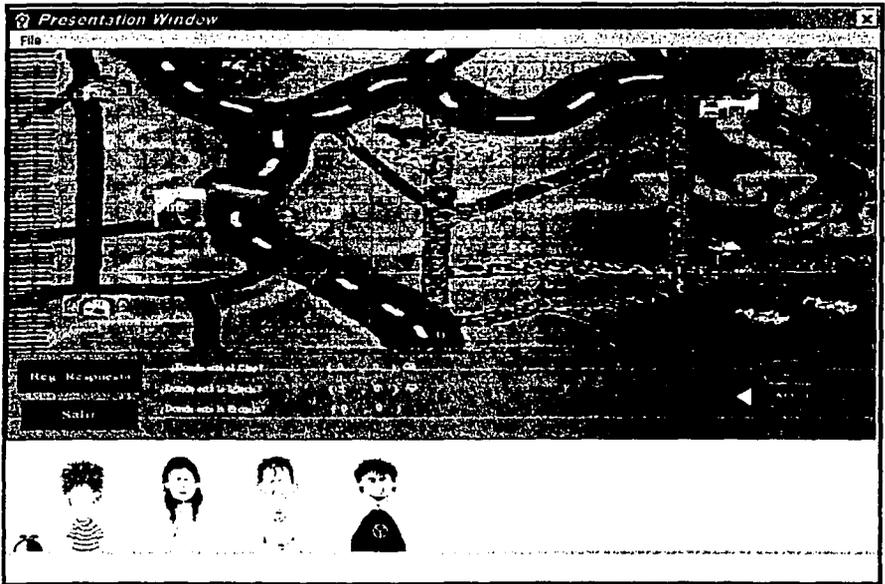


Figura 29. Presentación de un Juego Inteligente "Mi pueblito" para responder una duda.

El juego interactivo "Mi Pueblito", pretende que el alumno aprenda el concepto de coordenadas cartesianas, y para ello se le presenta el mapa de un pueblito donde debe de localizar y dar la posición de algunos lugares, como son el cine, la casa de la tía Lencha, la panadería, la escuela y la iglesia; todos con respecto a un origen. Para poder resolverlo debe de tener presente los conceptos que se revisaron en la primera presentación como son; ejes coordenados, distancias positivas y distancias negativas.

En esta figura observamos que el punto de referencia para localizar cada uno de los lugares que aparecen en la pantalla, es un círculo rojo, este será el origen de nuestro sistema de referencia. Sin embargo este punto se fija de manera dinámica por el alumno quien lo posiciona cuando pulsa el botón "AQUÍ".

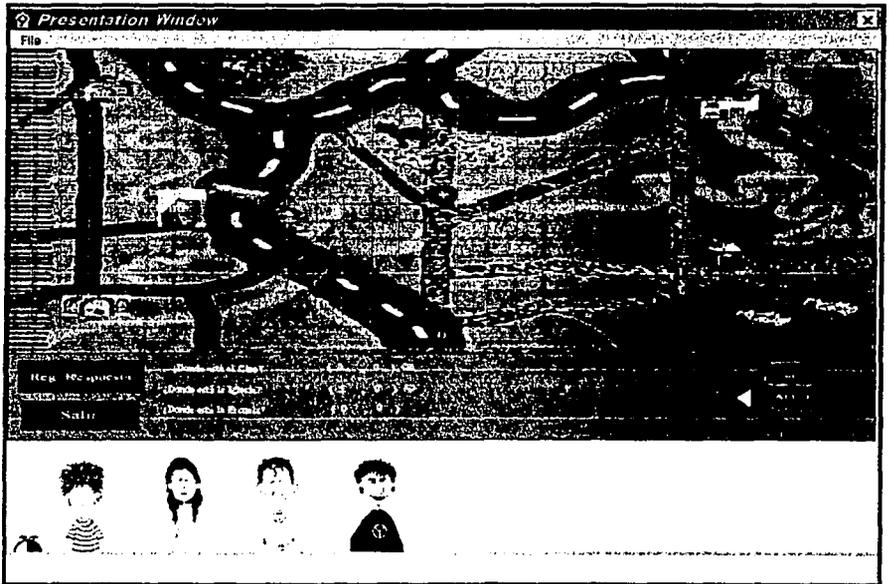


Figura 29. Presentación de un Juego Inteligente "Mi pueblito" para responder una duda.

El juego interactivo "Mi Pueblito", pretende que el alumno aprenda el concepto de coordenadas cartesianas, y para ello se le presenta el mapa de un pueblito donde debe de localizar y dar la posición de algunos lugares, como son el cine, la casa de la tía Lencha, la panadería, la escuela y la iglesia; todos con respecto a un origen. Para poder resolverlo debe de tener presente los conceptos que se revisaron en la primera presentación como son; ejes coordenados, distancias positivas y distancias negativas.

En esta figura observamos que el punto de referencia para localizar cada uno de los lugares que aparecen en la pantalla, es un círculo rojo, este será el origen de nuestro sistema de referencia. Sin embargo este punto se fija de manera dinámica por el alumno quien lo posiciona cuando pulsa el botón "AQUI".

Después de que se selecciona el origen, el juego muestra la siguiente pregunta ¿Dónde está el cine?, la cual debe ser contestada por el alumno.

El usuario puede revisar si su respuesta es correcta, pulsando el botón de "Reg. Respuesta". Si el alumno contesta bien, se le muestra una marca azul y si su respuesta es incorrecta, se le muestra una marca color rojo. Cada acierto y error se van registrando, para que al finalizar el juego se le proporcione al usuario una sugerencia.

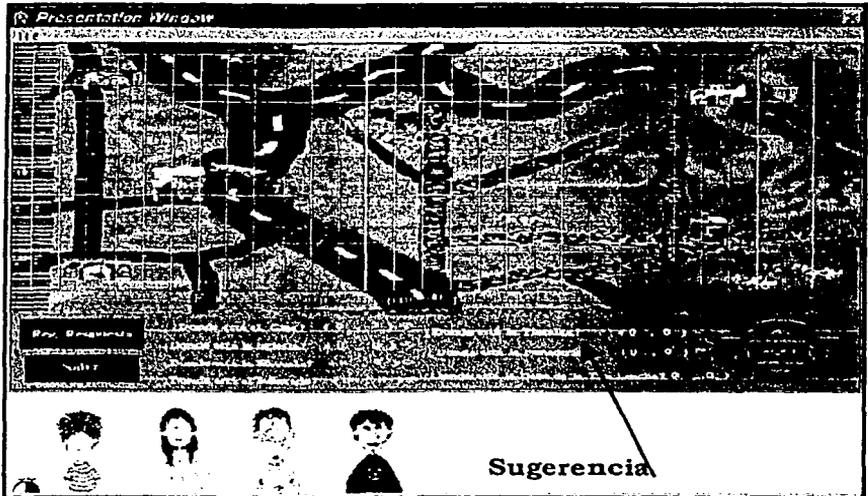


Figura 30. Sugerencia al alumno virtual por parte del Juego Inteligente "Mi Pueblito"

Los botones izquierda, derecha, arriba y abajo, le permiten al alumno navegar por el pueblito, de manera que pueda fijar el origen de su marco de referencia.

En el caso de que el alumno elija "Sistema", el módulo "Cascaarón del tutor" le vuelve a mostrar una nueva explicación del concepto no comprendido por medio de una nueva animación, semejante a la primera explicación.

6. RESULTADOS Y CONCLUSIONES

En esta tesis hemos estudiado el problema del desarrollo de software educativo de calidad, en particular para el caso específico de la física. El desarrollo de software educativo es de particular interés para países de escasos recursos, pues representan una tecnología que puede ser desarrollada a costos razonables y con potenciales beneficios sobre un gran número de estudiantes.

Sin embargo, desarrollar software educativo de calidad no es un problema sencillo.

En este trabajo hemos revisado los diversos requisitos que el software educativo debe satisfacer. Es por eso que el diseño de un software educativo requiere de instructores especializados, ingenieros de desarrollo, programadores, diseñadores, ilustradores, etc., todos trabajando bajo criterios y procedimientos rigurosos.

A este respecto existen cuatro puntos que nos parece importante remarcar:

- a. Hacer multimedia no puede ser visto como una actividad que consiste en generar materiales diversos, y reunirlos todos en una pantalla. Como aquí se mostró, una fase (normalmente larga) de análisis y diseño es indispensable.
- b. Hacer multimedia educativa no es un proceso que una sola disciplina pueda resolver. La discusión de quién debe guiar un proceso de esta naturaleza es falsa. Un sistema educativo multimedia de calidad sólo se puede lograr cuando se tienen en una mesa los diversos puntos de vista de manera organizada, de manera que no se convierta en una cacofonía de opiniones, sino que permitan acordar conclusiones de manera efectiva.
- c. Los procesos de desarrollo de software tradicionales no son adecuados para desarrollar sistemas como el que aquí se presenta. En ese sentido es importante desarrollar y validar propuestas metodológicas, en las que tantos expertos, equipos de desarrollo, diseñadores, artistas, etc. puedan converger, acordar y concluir proyectos. Dichos procesos sólo se vuelven viables, desde un punto de vista económico, cuando están soportados por herramientas especializadas.

Así, la experiencia recogida en el CCADET durante el desarrollo de este tipo de sistemas les ha permitido fundamentar que, si bien la intervención de distintos especialistas es la única manera de garantizar un producto final de calidad; dicho proceso puede volverse poco viable en la práctica fundamentalmente debido al tiempo requerido para su producción. Así, la necesidad de contar con herramientas que apoyen el trabajo de desarrollo, que faciliten el flujo de información entre los diferentes actores, que permitan generar prototipos que puedan ser evaluados, que permitan integrar técnicas de la inteligencia artificial, etc. se vuelve de primera importancia.

En esta Tesis se propusieron y fundamentaron módulos que permiten a los tres actores principales (profesor, estudiante, equipo de desarrollo), trabajar de manera coordinada, intercambiar información, y avanzar en el desarrollo del producto a velocidades (y por lo tanto, costos), razonables. El desarrollo de dichos módulos se basó e inspiró en las técnicas utilizadas en la Inteligencia Artificial, en particular de las relacionadas con la Ingeniería de Conocimientos.

Para demostrar la validez de la propuesta se desarrolló un caso de estudio, en el que se trataron los puntos principales que el equipo de desarrollo tuvo que seguir, y de cómo se organizó un equipo interdisciplinario para llevarlo a buen término.

Así, la propuesta aquí presentada es el resultado de analizar y concretar la experiencia acumulada por parte de los expertos en el CCADET en el desarrollo de software interactivo educativo. Consideramos que presenta las siguientes ventajas:

- *Ayuda al experto a crear su tutor de manera más independiente:* La herramienta brinda medios eficientes que permiten a los autores ingresar su material, generar sistemas que pueden ser probados con alumnos, e incluso, distribuirlos. De esta manera ayuda a profesores a desarrollar software educativo que sigue un esquema probado, dándoles flexibilidad e independencia, y permitiéndoles crear sus propios materiales sin ser totalmente dependientes de los equipos de la producción. Sin embargo, el sistema deja abierta la puerta a nuevas colaboraciones, permitiendo la integración de diversos materiales, desarrollada eventualmente por otras personas.
- *Se obtiene material interactivo en un formato probado:* El sistema presenta la información utilizando un formato que ha sido probado, y del que se tienen buenos resultados. Si bien esto puede hacer rígida

la manera de presentar los materiales, uno puede pensar en extensiones al presente trabajo que permitan utilizar diversos formatos.

- *Enfoque híbrido de técnicas interactivas y técnicas inteligentes:* La herramienta es capaz de unir, bajo una misma interfaz, mecanismos de hipertextos, así como inteligentes, de modo que el profesor pueda utilizar lo mejor de ambas disciplinas.
- *El enfoque atiende tanto a las necesidades del profesor como las del alumno:* Al estar dividida en dos herramientas diferentes, estamos considerando de hecho una herramienta que tiene las funciones y comandos que requiere el profesor, de un lado, y el alumno del otro.

Por supuesto hay muchos límites y restricciones a la clase de integraciones que se puedan manejar con los módulos. Estos módulos se deben ver como prototipos que permitirán evaluar nuevos paradigmas en la producción del software educativo.

APÉNDICE

Figura 1. Diagrama de Flujo del "Módulo del Profesor"

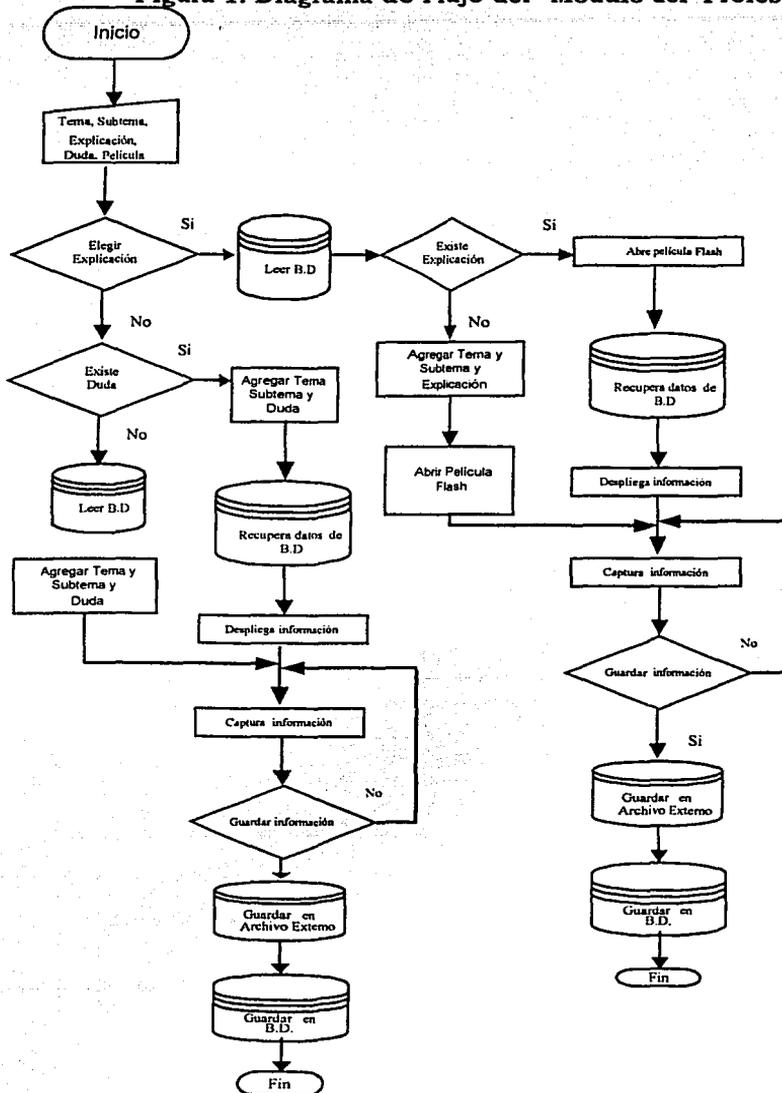


Figura 2. Diagrama de Flujo del módulo "Cascarón del Tutor"

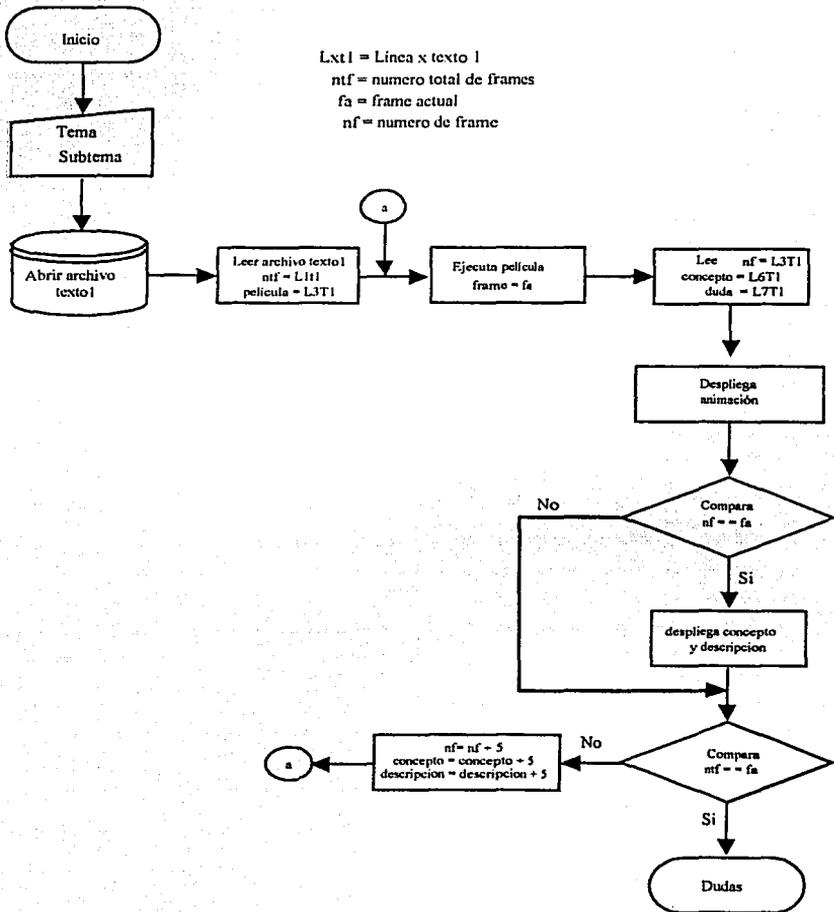


Figura 3. Diagrama de Flujo del módulo "Cascaón del Tutor"
Continuación.

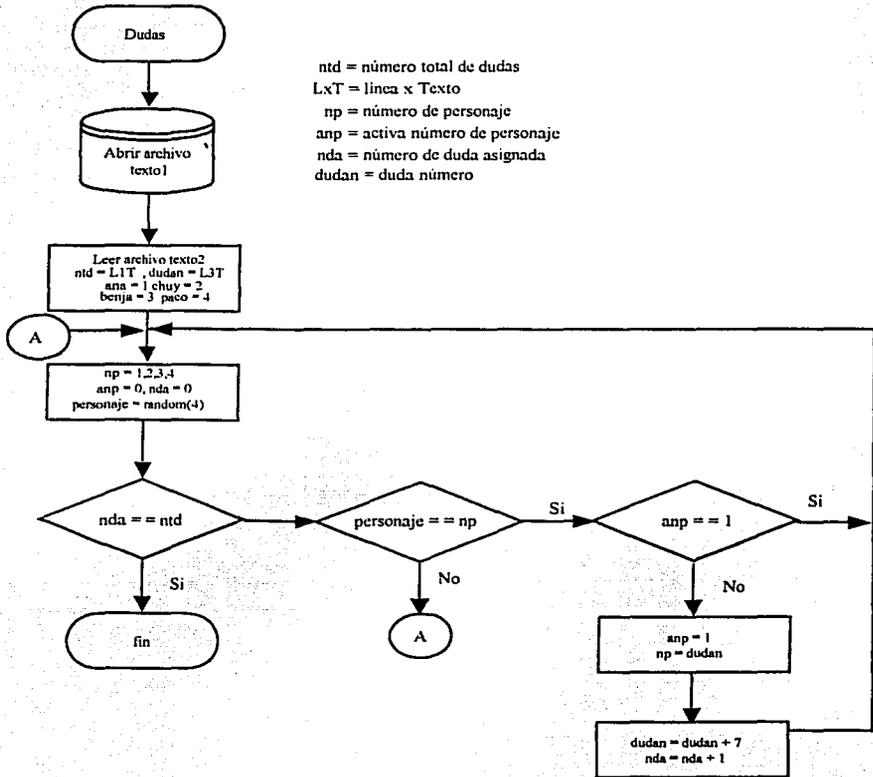


Figura 4. Diagrama de Flujo del módulo "Casarón del Tutor" Continuación.

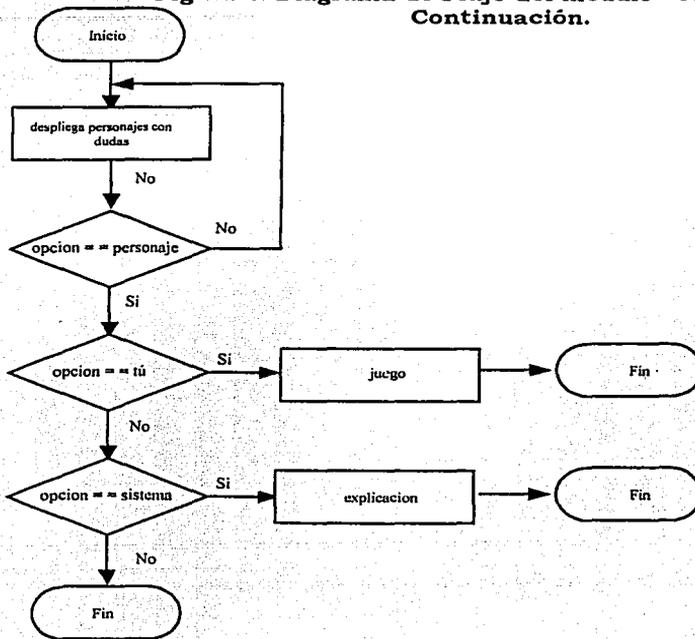
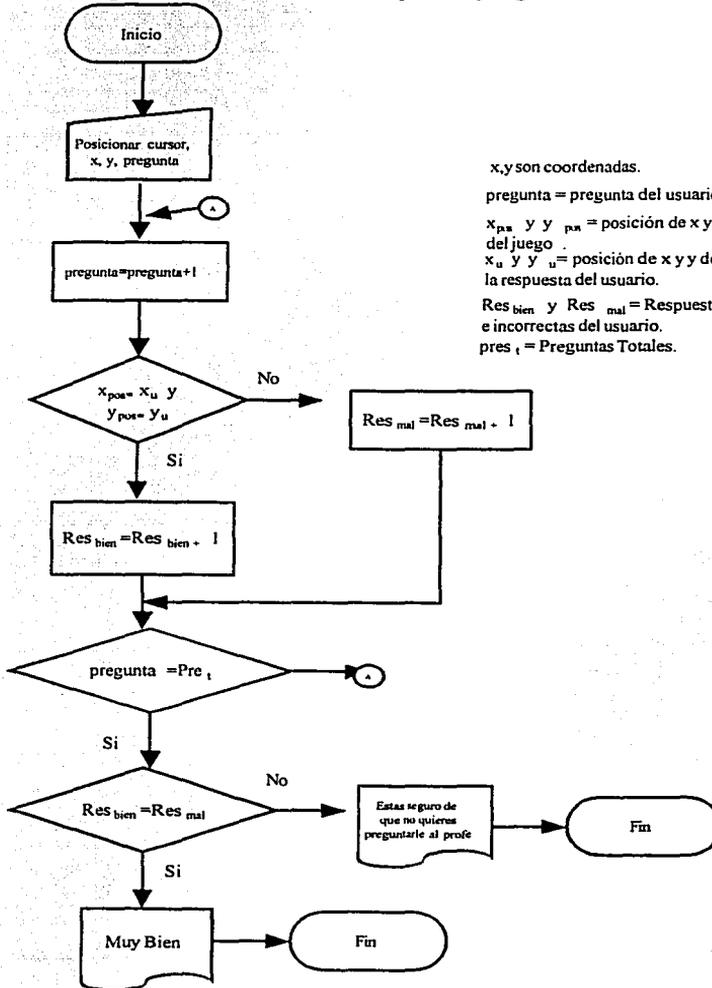
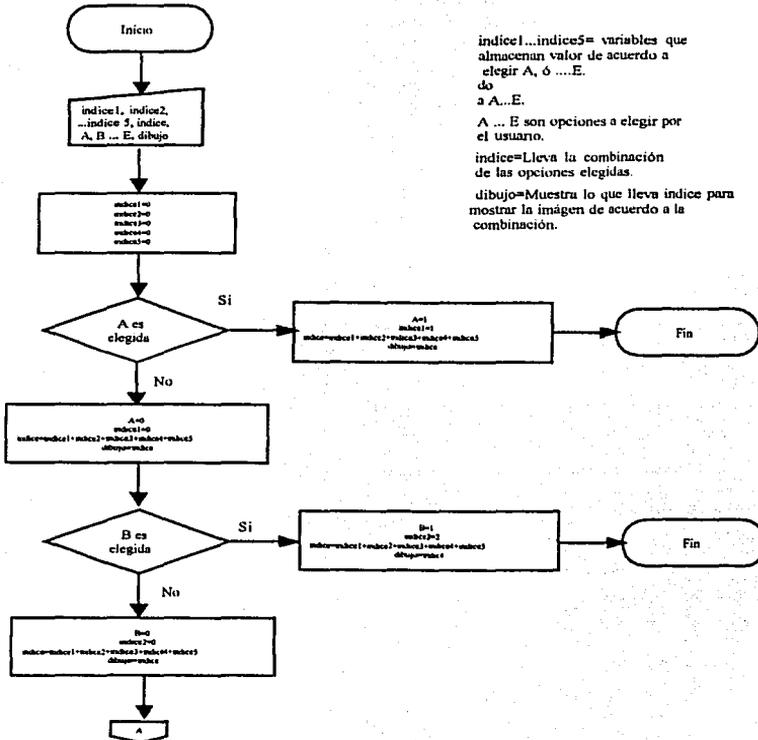


Figura 5 Diagrama de Flujo del juego interactivo "Mi Pueblito"



x,y son coordenadas.
 pregunta = pregunta del usuario
 x_{pos} y y_{pos} = posición de x y y del juego
 x_u y y_u = posición de x y y de la respuesta del usuario.
 Res bien y Res mal = Respuesta correctas e incorrectas del usuario.
 pres t = Preguntas Totales.

Figura 6. DIAGRAMA DE FLUJO DEL JUEGO INTERACTIVO "MORITA LA PULGUITA".



índice1...índice5= variables que almacenan valor de acuerdo a elegir A, ó ...E.

do a A...E.

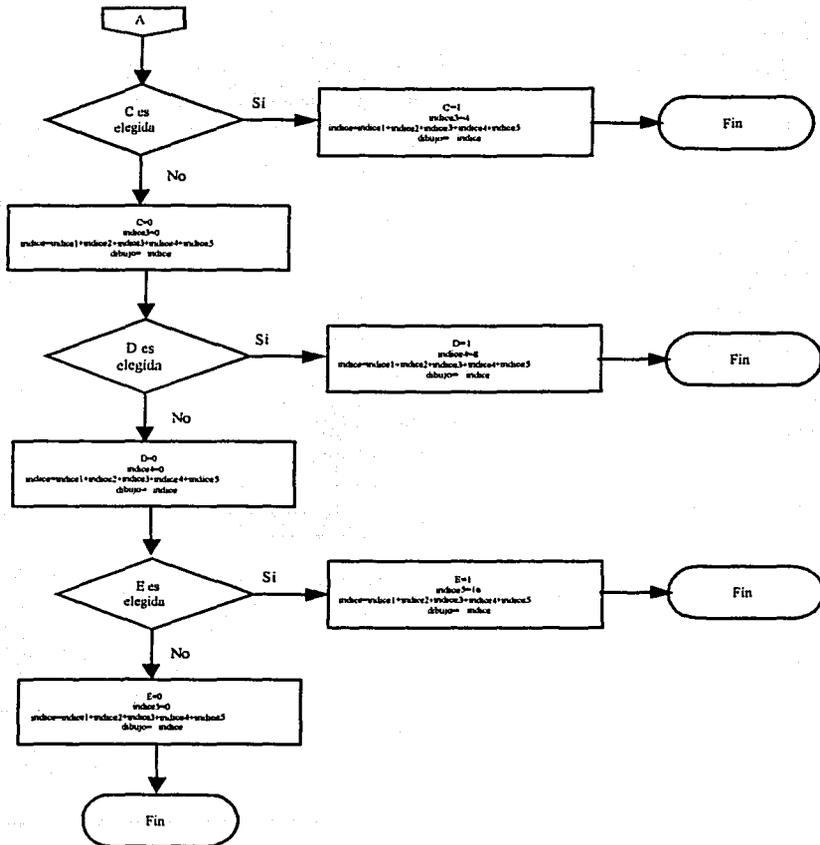
A ... E son opciones a elegir por el usuario.

índice=Lleva la combinación de las opciones elegidas.

dibujar=Muestra lo que lleva índice para mostrar la imagen de acuerdo a la combinación.

Figura 7. DIAGRAMA DE FLUJO DEL JUEGO INTERACTIVO "MORITA LA PULGUITA".

CONTINUACIÓN



TESIS CON
FALLA DE ORIGEN

1.-Código Fuente del "Módulo del Profesor".

```

'Codigo para Abrir la Pelicula Flash
Private Sub cmdAbrir_Click()
Dim k As Integer
Dim l As Integer
Dim temp As String
Abrir.InitDir = "c:\proyecto\Flash"
Abrir.Filter = "Películas de Flash(*.swf)|*.swf"
Abrir.ShowOpen
If Abrir.FileName <> "" Then
    Ruta.Caption = Abrir.FileName
    PeliculaFlash.Movie = Ruta.Caption
    PeliculaFlash.GotoFrame (-1)

    k = Val(PeliculaFlash.FrameNum)
    l = Val(PeliculaFlash.TotalFrames)
    txtframe.Text = k
    txtframes.Caption = l
    'Captura el nombre del archivo
    txtnarchivo.Text = Abrir.FileTitle

Else
    Ruta.Caption = "No se selecciono ningún archivo"
End If
End Sub
'*****Sirve para agregar la clave de la duda *****
Private Sub cmdagregarduda_Click()
'Agrega un elemento al datagrid
Dim i As Integer
If SStab1.Tab = 1 Then

    i = Adodc2.Recordset.RecordCount
    For i = Adodc2.Recordset.RecordCount To 0 Step -1
        If i = 0 Then
            MsgBox "Vas a agregar una duda "
            Adodc2.Recordset.AddNew
        End If
        Next
    txtconcaarchivo.Text = ""
    txttextduda.Text = ""
    txtnarchivo1.Text = ""
    txtnarchivo2.Text = ""
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    txtmain2.Text = ""

    'dgrid2.Row = dgrid2.VisibleRows
    dgrid2.Columns(0) = (txttextduda.Text)
    dgrid2.Columns(3).Text = "0"
    dgrid2.Columns(4) = txtnarchivo1.Text
    dgrid2.Columns(6) = txtnarchivo2.Text
    dgrid2.Refresh
    DoEvents
End If

```

Apéndice B. Código Fuente de los módulos.

```

.....
'Agregar en la parte de duda un archivo explicación
.....
Private Sub cmdagregarexplicación_Click()
Dim Titulomsg As String
Dim titulomsg1 As String
Dim msg As String
Dim msg1 As String
Dim respuesta As String
Dim respuesta1 As String

titulomsg1 = "Explicaciones"
msg1 = msg1 + vbCrLf + "Quieres continuar"
respuesta1 = MsgBox("Vas ha anexar un archivo", vbOKCancel, titulomsg1)
If respuesta1 = vbOK Then
'-----Codigo para abrir archivo Flash

Abrir1.Filter = "Archivos de Texto(*.txt)|*.txt"
Abrir1.InitDir = "c:\proyecto\textos"
Abrir1.ShowOpen
If Abrir1.FileName <> "" Then
Ruta.Caption = Abrir1.FileName
'Captura el nombre del archivo
txtnarchivo2.Text = Abrir1.FileTitle
dgrid2.Columns(6) = txtnarchivo2.Text
Else
Ruta.Caption = "No se selecciono ningún archivo"
End If
dgrid2.Refresh
' dgrid2.Row = dgrid2.VisibleRows
DoEvents
Elseif respuesta1 = vbCancel Then
Ruta.Caption = "No se selecciono ningún archivo"
MsgBox "no hay archivo, la siguiente columna debera permanecer vacia"
dgrid2.Columns(6).Text = ""

End If
DoEvents
End Sub
.....
'Sirve para agregar un juego en la parte de dudas
.....
Private Sub cmdagregarjuego_Click()
Dim k As Integer
Dim l As Integer
Dim temp As String
Dim j As Variant
Dim h As Variant
Dim respuesta1 As String
Dim sArray As Variant
Dim a As Integer
Dim b As Integer
Dim c As Integer

titulomsg1 = "Explicaciones"
msg1 = msg1 + vbCrLf + "Quieres continuar"
respuesta1 = MsgBox("Vas ha anexar un archivo", vbOKCancel, titulomsg1)
If respuesta1 = vbOK Then
Abre una ruta de archivo Flash
Abrir1.InitDir = "c:\proyecto\FIash"

```

Apéndice B. Código Fuente de los módulos.

```

Abrir1.Filter = "Archivos para Juegos(*.*)|*.*"
Abrir1.ShowOpen
If Abrir1.FileName <> "" Then
    Ruta.Caption = Abrir1.FileName
    'Captura el nombre del archivo
    txtnarchivo1.Text = Abrir1.FileTitle
Else
    Ruta.Caption = "No se selecciono ningún archivo"
End If
DoEvents
Dim z As Variant
dgrid2.Columns(4) = txtnarchivo1.Text
z = txtnarchivo1.Text
*****Funcion Split para separar cadenas*****
sArray = Split(z, ".", 2, vbTextCompare)
b = UBound(sArray)
For a = 0 To b
    Dim o As String
    o = sArray(1)
    txtextduda.Text = o
Next a
****decide de acuerdo a la extensión del archivo
If txtextduda.Text = "swf" Then
    dgrid2.Columns(3).Text = "1"
Elseif txtextduda.Text = "jpg" Then
    dgrid2.Columns(3).Text = "2"
Elseif txtextduda.Text = "avi" Then
    dgrid2.Columns(3).Text = "3"
Elseif txtextduda.Text = "gif" Then
    dgrid2.Columns(3).Text = "4"
Elseif txtextduda.Text = "" Then
    dgrid2.Columns(3).Text = "0"
End If
Elseif respuesta1 = vbCancel Then
    Ruta.Caption = "No se selecciono ningún archivo"
    MsgBox "no hay archivo, Esta columna deberá permanecer en blanco"
    dgrid2.Columns(3).Text = "0"
    dgrid2.Columns(4).Text = ""
End If
DoEvents
End Sub
*****
"Agrega un elemento al datagrid que es concepto
*****
Private Sub cmdagregarts_Click()
Dim i As Integer
i = Adodc1.Recordset.RecordCount
For i = Adodc1.Recordset.RecordCount To 0 Step -1
    If i = 0 Then
        MsgBox "Vas a agregar un concepto"
        Adodc1.Recordset.AddNew
    End If
Next
'dgrid1.Refresh
dgrid1.Row = dgrid1.VisibleRows - 2
dgrid1.Columns(0) = (txtMain1.Text)
dgrid1.Columns(1) = txtnarchivo.Text
dgrid1.Columns(2) = txtframe.Text
dgrid1.Refresh

```

Apéndice B. Código Fuente de los módulos.

```
DoEvents
End Sub
'.....
' Manipula controles de avanzar en la película Flash
'.....
Private Sub cmdanterior_Click()
If PeliculaFlash.FrameNum > 0 Then
    PeliculaFlash.GotoFrame (PeliculaFlash.FrameNum - 1)
    txtframe.Text = Val(PeliculaFlash.FrameNum)
End If
End Sub
'Borra un elemento de la base de datos
Private Sub cmdborrar_Click()
Dim i As Integer
i = Adodc1.Recordset.RecordCount
For i = Adodc1.Recordset.RecordCount To 0 Step -1
    If i = 0 Then
        MsgBox "Vas a eliminar tu concepto"
        Adodc1.Recordset.Delete
        Adodc1.Recordset.MoveFirst
    End If
End Sub
DoEvents
Next
End Sub
'.....
Borra un renglón duda
'.....
Private Sub cmdBorrarduda_Click()
Dim i As Integer
i = Adodc2.Recordset.RecordCount
For i = Adodc2.Recordset.RecordCount To 0 Step -1
    If i = 0 Then
        MsgBox "Vas a eliminar tu duda"
        Adodc2.Recordset.Delete
    End If
    DoEvents
Next
End Sub
'.....
'Sirve para guardar el archivo de Conceptos
'.....
Private Sub cmdarchivoexterno_Click()
Dim strfilename As String 'Cadena del archivo abrir
Dim strText As String 'Contenido del archivo
Dim strFilter As String 'Cadena de Filtro del common dialog
Dim strBuffer As String 'Variable de búfer de la cadena
Dim FileHandle% 'Variable que contiene el número de archivo
Dim strText1 As String
Dim strtext2 As String
Dim strtext3 As String
Dim i As Integer, j As Integer, k As Integer
Dim strregistro As String
Dim strregistro1 As String
Dim l As Integer
Dim res As String
Dim Ruta As String

res = MsgBox("Debes tener abierto tu archivo de película con el que estas trabajando",
vbInformation)
```

Apéndice B. Código Fuente de los módulos.

```
res = MsgBox("Tienes abierto tu archivo Flash ", vbYesNo)
If res = vbYes Then

    'Establece el filtro del Common Dialog
    cdmain.InitDir = "c:\proyecto\txtos"
    cdmain.Filter = "Archivos(*.txt)|*.txt"
    cdmain.DefaultExt = ".txt"
    cdmain.FileName = "C" & txtMain1.Text
    cdmain.ShowSave
    texto1.Text = cdmain.FileName

'Se asegura de que el nombre de un archivo recuperado no es una cadena en blanco
    If cdmain.FileName <> "" Then
        'Si no esta en blanco, abre el archivo
        strfilename = cdmain.FileName
        Dim fso, MiArchivo
        Set fso = CreateObject("Scripting.FileSystemObject")
        Set MiArchivo = fso.CreateTextFile(cdmain.FileName, True)
        k = dgrid1.VisibleRows - 2 ' Se toma -2 porque ve un renglon en blanco al
        final y ademas cuenta desde 0
        ' k = Adodc1.Recordset.RecordCount - 2
        strregistro = ""
        For j = 0 To k
            dgrid1.Row = j
            For i = 0 To 4
                strregistro = strregistro + dgrid1.Columns(i).Text + vbCrLf
            Next i
        Next j
        l = Val(PeliculaFlash.TotalFrames)
        strregistro = txtframes.Caption + vbCrLf + strregistro + vbCrLf
        txtmain2.Text = strregistro
        MiArchivo.WriteLine(txtmain2.Text)
        MiArchivo.Close
    End If

'Guarda en la base de datos Conceptos T
    On Error GoTo Err_SeleccionaVia
    Dim SQL1, SQLcampo(5), SQLcomando As String
    Dim MiBDD As Database, MiTabla As Recordset
    Dim AUX As Integer
    Dim ERROR As String
    Dim clave As String
    clave = Format(Combo1.ListIndex + 1, "000") & Format(Combo2.ListIndex + 1,
    "000") & "E" & Format(Explicaciones.ListIndex + 1, "000")

'clave = Combo1.ListCount + 1 & Combo2.ListCount + 1

    txtMain.Text = clave

    ' abrir BDt y borrar contenido
    Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
    'Abrir la tabla t para insertar en m
    SQLcampo(1) = "INSERT INTO conceptoM (no_de_tema, flash, no_frame,
    concepto, descripcion)"
    SQLcampo(2) = " SELECT conceptoT.no_de_tema, conceptoT.flash,
    conceptoT.no_frame, conceptoT.concepto, conceptoT.descripcion FROM
    conceptoT"
    SQLcomando = SQLcampo(1) & SQLcampo(2)
    MiBDD.Execute SQLcomando, dbFailOnError
    MiBDD.Close
    dgrid1.Refresh
    Adodc1.Refresh
```

Apéndice B. Código Fuente de los módulos.

```

exit Sub
err_SeleccionaVia:
  MsgBox ERROR & ": " & Err
  Establece el cursor normal
  Screen.MousePointer = 0
Exit Sub
  DoEvents
Elseif res = vbNo Then
  MsgBox "No tiene archivo Flash"

  DoEvents

End If
End Sub
.....
'Busca en la base de datos si es una duda o explicación en base a tema, subtema, explicación o
duda.
.....
Private Sub cmdbuscar_Click()
Dim clave As String
  On Error GoTo Err_SeleccionaVia
  Dim SQL1, SQLcampo(5), SQLcomando As String
  Dim MiBDD As Database, MiTabla As Recordset
  Dim AUX As Integer
  Dim ERROR As String
If ((Combo3.ListIndex + 1 <> 0)) Then
  clave = Format(Combo1.ListIndex + 1, "000") & Format(Combo2.ListIndex + 1, "000") &
"D" & Format(Combo3.ListIndex + 1, "000")
  txtduda.Text = clave
  txtnarchivo1.Text = ""
  txtnarchivo2.Text = ""
  txttxtduda.Text = ""
  txtconcaarchivo.Text = ""
  Text1.Text = ""
  Text2.Text = ""
  Text3.Text = ""
  txtmain2.Text = ""
Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
  SQL1 = "Delete dudaT.clv_duda FROM dudaT"
  MiBDD.Execute SQL1, dbFailOnError

Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
' abrir BDM y copiar de la BDM a la BDT todo lo que sea = clave
SQLcampo(1) = "INSERT INTO dudaT (clv_duda, duda, ejuego, tipo_juego,
nombre_juego, eexplicacion, explicacion)"
' SQLcampo(2) = " SELECT (clave_duda, texto_duda, juegosn, tipo_juego, ruta_juego,
explicacionn, ruta_explicacion) FROM dudaM"
SQLcampo(2) = "SELECT dudaM.clv_duda, dudaM.duda, dudaM.ejuego,
dudaM.tipo_juego, dudaM.nombre_juego, dudaM.eexplicacion, dudaM.explicacion FROM
dudaM"
SQLcampo(3) = " WHERE clv_duda = " & clave & ""
SQLcomando = ""
For AUX = 1 To 3
  SQLcomando = SQLcomando & SQLcampo(AUX)
Next AUX
MiBDD.Execute SQLcomando, dbFailOnError
'Borrar de la BDM todo lo que sea = clave
SQL1 = "Delete FROM dudaM WHERE dudaM.clv_duda = " & clave & "";"
MiBDD.Execute SQL1, dbFailOnError
MsgBox ERROR & ": " & Err
Establece el cursor normal

```

```

Adodc2.Refresh
dgrid2.Refresh
DoEvents

```

Else

```

    'clave = Format(Combo1.ListIndex + 1, "000") & Format(Combo2.ListIndex + 1, "000")

    clave = Format(Combo1.ListIndex + 1, "000") & Format(Combo2.ListIndex + 1, "000")
    & "E" & Format(Explicaciones.ListIndex + 1, "000")

```

```

    txtMain1.Text = clave
    txtMain2.Text = ""
    txtmain2.Text = ""
    txtframe.Text = ""
    txtarchivo.Text = ""
    ttlframes.Caption = ""
    ' abrir BDT y borrar contenido
    Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
    SQL1 = "Delete conceptoT.no_de_tema FROM conceptoT;"
    MiBDD.Execute SQL1, dbFailOnError

```

```

    ' abrir BDM y copiar de la BDM a la BDT todo lo que sea = clave
    SQLcampo(1) = "INSERT INTO conceptoT (no_de_tema, flash, no_frame, concepto,
descripcion)"
    SQLcampo(2) = " SELECT conceptoM.no_de_tema, conceptoM.flash,
conceptoM.no_frame, conceptoM.concepto, conceptoM.descripcion FROM conceptoM"
    SQLcampo(3) = " WHERE no_de_tema = " & clave & ""

```

```

    SQLcomando = ""
    For AUX = 1 To 3
        SQLcomando = SQLcomando & SQLcampo(AUX)
    Next AUX
    MiBDD.Execute SQLcomando, dbFailOnError

```

```

    ' Borrar de la BDM todo lo que sea = clave
    SQL1 = "Delete FROM conceptoM WHERE conceptoM.no_de_tema = " & clave & ";"
    MiBDD.Execute SQL1, dbFailOnError

```

```

    DoEvents
    Err_SeleccionaVia:
    'MsgBox ERROR & ": " & Err
    'Establece el cursor normal
    Screen.MousePointer = 0

```

```

    Adodc1.Refresh
    dgrid1.Refresh
    Adodc2.Refresh
    dgrid2.Refresh
    Dim k As Integer
    Dim l As Integer
    Dim temp As String
    Dim s As String

```

```

If Err.Number <> 6160 Then
    ' Indica al usuario lo que ha ocurrido. Luego borra el objeto Err.
    ' MsgBox msg, "Prueba de error retardada"
    Err.Clear ' Borra campos del objeto Err

```

```

Text4.Text = dgrid1.Columns(1).Text

```

Apéndice B. Código Fuente de los módulos.

```

Text5.Text = "c:\proyecto\FIash\" & Text4.Text

If Text4.Text <> "" Then
Abrir.InitDir = "c:\proyecto\FIash\"
Abrir.Filter = "Películas de Flash (*.swf) | *.swf"
'Abrir.ShowOpen
'If Abrir.FileName <> "" Then
Text4.Text = Abrir.FileName
PelículaFlash.Movie = Text5.Text
PelículaFlash.GotoFrame (-1)

k = Val(PelículaFlash.FrameNum)
l = Val(PelículaFlash.TotalFrames)
txtframe.Text = k
ttlframes.Caption = l
'Captura el nombre del archivo
xtxnarchivo.Text = Abrir.FileTitle
txtnarchivo.Text = Text4.Text
End If
Else
MsgBox ("Selecciona tu archivo de Flash")
cmdAbrir.SetFocus

End If
Exit Sub
End Sub
'.....
'Genera el Archivo Duda
'.....
Private Sub cmdgenerararchivoduda_Click()
Dim strfilename As String 'Cadena del archivo abrir
Dim strText As String 'Contenido del archivo
Dim strFilter As String 'Cadena de Filtro del common dialog
Dim strBuffer As String 'Variable de búfer de la cadena
Dim FileHandle% 'Variable que contiene el número de archivo
Dim i As Integer, j As Integer, k As Integer
Dim strregistro As String
Dim strregistro1 As String
Dim Ruta As String

'Establece el filtro del Common Dialog
cdmain.InitDir = "c:\proyecto\textos"
cdmain.FileName = "D" & txtduda.Text
cdmain.Filter = "Archivos (*.txt) | *.txt"
cdmain.ShowSave
texto2.Text = cdmain.FileName
'Se asegura de que el nombre de un archivo recuperado no es una cadena en blanco
If cdmain.FileName <> "" Then
'Si no esta en blanco, abre el archivo
strfilename = cdmain.FileName
'Abrir un archivo para escritura
Dim fso, MiArchivo
Set fso = CreateObject("Scripting.FileSystemObject")
Set MiArchivo = fso.CreateTextFile(cdmain.FileName, True)
k = dgrid1.VisibleRows - 2 ' Se toma -2 porque ve un renglon en blanco al final y ademas
cuenta desde 0
'k = Adodc2.Recordset.RecordCount
strregistro = ""
For j = 0 To k

```

```

dgrid2.Row = j
For i = 0 To 6
    strregistro = strregistro + dgrid2.Columns(i).Text + vbCrLf
Next i
Next j
cdmain.FileName = txt duda.Text
Dim l As String
txtconcaarchivo.Text = strregistro
strregistro = k & vbCrLf & txtconcaarchivo.Text
Text3.Text = strregistro
MiArchivo.WriteLine (Text3.Text)
MiArchivo.Close
End If
*****Parte de envio a la base de datos de T a M *****
On Error GoTo Err_SeleccionaVia
Dim SQL1, SQLcampo(5), SQLcomando As String
Dim MiBDD As Database, MiTabla As Recordset
Dim AUX As Integer
Dim ERROR As String
Dim clave As String
clave = Format(Combo1.ListIndex + 1, "000") & Format(Combo2.ListIndex + 1, "000") & "D" &
Format(Combo3.ListIndex + 1, "000")
txt duda.Text = clave
Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
'Abrir la tabla t para insertar en m
SQLcampo(1) = "INSERT INTO dudaM (civ_duda, duda, juego, tipo_juego, nombre_juego,
explicacion, explicacion)"
SQLcampo(2) = "SELECT dudaT.civ_duda, dudaT.duda, dudaT.juego, dudaT.tipo_juego,
dudaT.nombre_juego, dudaT.explicacion, dudaT.explicacion FROM dudaT"
SQLcomando = SQLcampo(1) & SQLcampo(2)
MiBDD.Execute SQLcomando, dbFailOnError
MiBDD.Close
dgrid1.Refresh
Adodc1.Refresh
Exit Sub
Err_SeleccionaVia:
MsgBox ERROR & ": " & Err
'Establece el cursor normal
Screen.MousePointer = 0
Exit Sub
DoEvents
End Sub
*****
'Inserta una linea.
*****
Private Sub cmdinsertar_Click()
On Error GoTo Err_SeleccionaVia
Dim SQL1, SQLcampo(5), SQLcomando As String
Dim MiBDD As Database, MiTabla As Recordset
Dim AUX As Integer
Dim ERROR As String
Dim clave As String
Dim i, j, h As Integer
Dim listat As String
i = dgrid2.Bookmark
MsgBox (i)
j = dgrid2.VisibleRows - 1
MsgBox (j)
h = j - i
dgrid2.ZOrder (1)

```

Apéndice B. Código Fuente de los módulos.

```

i = Adodc2.Recordset.RecordCount
For i = Adodc2.Recordset.RecordCount To 0 Step -1
  If i = 0 Then
    MsgBox "Vas a agregar una duda "
    Adodc2.Recordset.AddNew
  End If
Next
txtconcaarchivo.Text = ""
txttxtduda.Text = ""
txtnarchivo1.Text = ""
txtnarchivo2.Text = ""
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
txtmain2.Text = ""
dgrid2.Row = dgrid2.VisibleRows
dgrid2.Columns(0) = (txtduda.Text)
dgrid2.Columns(3).Text = "0"
dgrid2.Columns(4) = txtnarchivo1.Text
dgrid2.Columns(6) = txtnarchivo2.Text
dgrid2.Refresh
For j = dgrid2.VisibleRows To h Step -1
  For l = 6 To 0
    List1.List(l) = dgrid2.Columns(l).Text
  Next l
Next j
For m = dgrid2.VisibleRows To h Step -1
  For l = 6 To 0
    dgrid2.Columns(l).Text = List1.ItemData(l)
  Next l
Next m
DoEvents
clave = Format(Combo1.ListIndex + 1, "000") & Format(Combo2.ListIndex + 1, "000") & "D" &
Format(Combo3.ListIndex + 1, "000")
' txtduda.Text = clave
' Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
' Abrir la tabla t para insertar en m
' SQLcampo(1) = "SELECT dudaT.clv_duda, dudaT.duda, dudaT.ejuego, dudaT.tipo_juego,
dudaT.nombre_juego, dudaT.eexplicacion, dudaT.explicacion FROM dudaT ORDER BY
dudaT.duda ASC;"
' SQLcomando = SQLcampo(1)
' MiBDD.Execute SQLcomando, dbFailOnError
' MiBDD.Close
' dgrid1.Refresh
' Adodc1.Refresh
Exit Sub
Err_SeleccionaVia:
' MsgBox ERROR & "; " & Err
' Establece el cursor normal
Screen.MousePointer = 0
Exit Sub
DoEvents
i = Adodc2.Recordset.RecordCount
For i = Adodc2.Recordset.RecordCount To 0 Step -1
  If i = Adodc2.Visible Then
    MsgBox "Vas a agregar una duda "
    Adodc2.Recordset.AddNew
  End If
Next
txtconcaarchivo.Text = ""

```

Apéndice B. Código Fuente de los módulos.

```
txttextduda.Text = ""
txtnarchivo1.Text = ""
txtnarchivo2.Text = ""
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
txtmain2.Text = ""
'dgrid2.Row = dgrid2.VisibleRows
dgrid2.Columns(0) = (txttextduda.Text)
dgrid2.Columns(3).Text = "0"
dgrid2.Columns(4) = txtnarchivo1.Text
dgrid2.Columns(6) = txtnarchivo2.Text
dgrid2.Refresh
DoEvents
End Sub
'.....
Controles de la Pelicula Flash Pausa, Play, Stop , Anterior y Siguiente.
'.....
Private Sub cmdpausa_Click()
    PeliculaFlash.Stop
End Sub
'.....
Private Sub cmdplay_Click()
    PeliculaFlash.Play
    While PeliculaFlash.IsPlaying
        txtframe.Text = Val(PeliculaFlash.FrameNum)
        DoEvents
    End While
Exit Sub
Wend
End Sub
'.....
Private Sub cmdsiguiente_Click()
If PeliculaFlash.FrameNum < PeliculaFlash.TotalFrames Then
    PeliculaFlash.GotoFrame (PeliculaFlash.FrameNum + 1)
    txtframe.Text = Val(PeliculaFlash.FrameNum)
End If
End Sub
'.....
Private Sub cmdstop_Click()
    PeliculaFlash.FrameNum = 0
    txtframe.Text = Val(PeliculaFlash.FrameNum)
End Sub
'.....
Ordena la información de la tabla Genera Explicación.
'.....
Private Sub Command1_Click()
Dim clave As String
    Dim SQL1, SQLcampo(5), SQLcomando As String
    Dim MiBDD As Database, MiTabla As Recordset
    Dim AUX As Integer
    Dim ERROR As String
Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
' 'Abrir la tabla dudas para insertar en dudaO
' SQL1 = "Delete * FROM concepto0"
' MiBDD.Execute SQL1, dbFailOnError
' SQL1 = "Delete * FROM concepto0"
' MiBDD.Execute SQL1, dbFailOnError
' Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
SQLcampo(1) = "INSERT INTO concepto0 (no_de_tema, flash, no_frame, concepto, descripcion)"
```

```

SQLcampo(2) = "SELECT conceptoT.no_de_tema, conceptoT.flash, conceptoT.no_frame,
conceptoT.concepto, conceptoT.descripcion FROM conceptoT ORDER BY [no_frame] ASC;"
SQLcomando = SQLcampo(1) & SQLcampo(2)
MiBDD.Execute SQLcomando, dbFailOnError
    SQL1 = "Delete * FROM conceptoT"
    MiBDD.Execute SQL1, dbFailOnError
SQLcampo(1) = "INSERT INTO conceptoT (no_de_tema, flash, no_frame, concepto, descripcion)"
SQLcampo(2) = " SELECT conceptoO.no_de_tema, conceptoO.flash, conceptoO.no_frame,
conceptoO.concepto, conceptoO.descripcion FROM conceptoO"
SQLcomando = SQLcampo(1) & SQLcampo(2)
MiBDD.Execute SQLcomando, dbFailOnError
MiBDD.Close
dgrid1.Refresh
Adodc1.Refresh
'Err_SeleccionaVia:
' MsgBox ERROR & ": " & Err
' Establece el cursor normal
' Screen.MousePointer = 0
Exit Sub
DoEvents
' txtMain1.Text = clave
End Sub
.....
Al correr el programa se muestra la primera pantalla de Explicación y Borra algunas tablas de la
Base de Datos
.....
Private Sub Form_Load()
    SSTab1.Tab = 0
    Form1.Caption = "Shell de Captura"
    Dim clave As String
    'On Error GoTo Err_SeleccionaVia
    Dim SQL1, SQL2, SQLcampo(5), SQLcomando As String
    Dim MiBDD As Database, MiTabla As Recordset
    Dim AUX As Integer
    Dim ERROR As String
    Dim temp As String
    ' abrir Bdt y borrar contenido
    Set MiBDD = Workspaces(0).OpenDatabase("C:\proyecto\bdatos.mdb")
    SQL1 = "Delete conceptoT.no_de_tema FROM conceptoT;"
    MiBDD.Execute SQL1, dbFailOnError
    SQL2 = "Delete dudaT.clv_duda From dudaT"
    MiBDD.Execute SQL2, dbFailOnError
    MiBDD.Close
    Adodc1.Refresh
    dgrid1.Refresh
    Adodc2.Refresh
    dgrid2.Refresh
End Sub
.....
'Código para el Menú Ayuda
.....
Private Sub mnuDuda_Click()
    Dim s As String
    Load Form2
    Form2.Visible = True
    Form2.Caption = "Generar Archivo Duda"
    s = "1.- Escoger Escoger Tema, Subtema y Duda, Dar click en el botón Buscar."
    s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
    s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
    s = s + "2.- Agregar una duda"

```

Apéndice B. Código Fuente de los módulos.

```

s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Cada vez que se introduzca una nueva duda, se deberá dar click en el botón Agregar Duda."
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Cuando agregas una nueva duda, se agrega el no de tema, subtema y duda."
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "3.- Borrar una duda "
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Para borrar una duda es necesario posicionarse en el renglón que se quiere borrar."
s = s + "Cabe resaltar que borra toda la línea."
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "4.- Guardar un Archivo de Duda"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Click en el botón de Generar Archivo y Finalizar, este Archivo se guarda en el directorio c:\proyecto\textos"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Los archivos que se generan se guardan con el nombre de D de Duda, 6 números consecutivos que indican el Tema y subtema, siendo los tres primeros que indican el Tema, y los otros tres del Subtema, además van seguidos de una D que indica Duda y tres números más que indican que se trata de una duda al que corresponden"
Form2.Text1.Text = s
End Sub
*****
Private Sub mnuExplicacion_Click()
Dim s As String

Load Form2
Form2.Visible = True
Form2.Caption = "Generar Archivo Explicación"
Form2.Text1.Text = "Escoger Tema, Subtema y Duda, Dar click en el botón Registrar, a continuación Abrir el Archivo de la Película., Cada vez que se agregue un nuevo Concepto, Dar click en el botón Agregar Concepto, para guardar el archivo dar click en Guardar Achivo, No debes Olvidar tener tu archivo Película Abierto, pues de lo contrario, se generara un error."
s = "1.- Escoger Escoger Tema, Subtema y Explicación, Dar click en el botón Elegir."
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Abrir Película"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "En el botón Abrir Película, en el directorio c:\Proyecto\Flash\la película a cargar. "
s = s + "Esta película puedes manipularla con los botones de la Play, Siguiente, Anterior, Stop, Pausa para manipular la película "
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "2.- Agregar un concepto"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Cada vez que se introduzca un nuevo concepto se deberá dar click en el botón Agregar Concepto"
s = s + "Cuando agregas un nuevo concepto se agrega el no de tema, subtema y explicación, y el nombre del archivo de película"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "3.- Borrar un concepto "
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Para borrar un concepto es necesario posicionarse en el renglón que se quiere borrar. Cabe resaltar que borra toda la línea."
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "4.- Guardar un Archivo de Explicación"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Click en el botón de Guardar y Terminar, Debe estar abierto el archivo de película que estas utilizando para este Tema."

```

Apéndice B. Código Fuente de los módulos.

```
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Los archivos que se generan se guardan con el nombre de C de Concepto 6 numeros
consecutivos que indican el Tema y subtema, siendo los tres primeros que indican el Tema, y los
otros tres del Subtema, además van seguidos de una E que indica Explicación y tres números
más que indican el número de explicación al que corresponden"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Siempre se debe tener abierto el Archivo de Pelicula, si es que se quiere volver a Editar
un archivo."
s = s + "5.- Ordenar un los datos por número de Frame "
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "Click en el botón de Ordenar datos, y automáticamente se ordenaran.
Form2.Text1.Text = s
End Sub
.....
Private Sub mnuGeneral_Click()
Dim s As String
Load Form2
Form2.Visible = True
Form2.Caption = "General"

s = "Al cargar el proyecto debes tener en cuenta lo siguiente:"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "1.- Agregar en c:\ La carpeta Proyecto, que contiene carpetas llamadas: Flash y Textos,
además de la base de datos llamada bdatos."
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "En la carpeta Flash se deberán guardar los archivos de película que se generaron en
Macromedia Flash. "
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "En la carpeta textos se guardarán los archivos que se generaran en el Shell hecho en
Visual Basic"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + "2.- Ejecutar el proyecto shell_Fisica desde el programa Visual Basic. Con la tecla F5"
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
s = s + Chr$(13) + Chr$(10) + Chr$(13) + Chr$(10)
Form2.Text1.Text = s
End Sub
.....
Private Sub mnuSalir_Click()
End
End Sub
.....
'Elegir una pestaña.
.....
Private Sub SStab1_Click(PreviousTab As Integer)
If SStab1.Tab = 0 Then
Combo3.Visible = False
Explicaciones.Visible = True
Elseif SStab1.Tab = 1 Then
Explicaciones.Visible = False
Combo3.Visible = True
End If
End Sub
```

2.- Código Fuente del módulo "Cascarón del Tutor"

```

INTRODUCCION DE NUMERO TEMA, SUBTEMA.
subtema:=UpperCase(EntryText)
lontema:= CharCount(tema)
lonsubtema:=CharCount(subtema)
if lontema = 1 & lonsubtema=1 then
  concepto:=UpperCase("c" ^ "0" ^ tema ^ "00" ^ subtema ^ "e001")
  dudas:=UpperCase("d" ^ "00" ^ tema ^ "00" ^ subtema ^ "d001")
else
if lontema =2 & lonsubtema=1 then
  concepto:=UpperCase("c" ^ "0" ^ tema ^ "00" ^ subtema ^ "e001")
  dudas:=UpperCase("d" ^ "0" ^ tema ^ "00" ^ subtema ^ "d001")
else
if lontema =1 & lonsubtema=2 then
  concepto:=UpperCase("c" ^ "00" ^ tema ^ "0" ^ subtema ^ "e001")
  dudas:=UpperCase("d" ^ "00" ^ tema ^ "0" ^ subtema ^ "d001")
else
if lontema =2 & lonsubtema=2 then
  concepto:=UpperCase("c" ^ "0" ^ tema ^ "0" ^ subtema ^ "e001")
  dudas:=UpperCase("d" ^ "0" ^ tema ^ "0" ^ subtema ^ "d001")
end if
end if
end if
end if

```

LEE ARCHIVO DE TEXTO

```

archivo := ReadExtFile (FileLocation ^ "textos\\" ^ "concepto" ^ ".txt")
totalframe:= GetLine(archivo, 1)
despliega :=GetLine(archivo, 3)

```

DESPLIEGA ANIMACION

```

SetSpriteProperty(@ "anima", #Movie,(FileLocation ^ "\dash\\" ^ "despliega)
CallSprite(@ "anima", #Play)
SetSpriteProperty(@ "anima", #scale ,100)
--contiene numero de frame
linea:=4
renglon:=5
descripcion:=6
mensaje:=Array("",30)
definicion:= Array("",30)
cont:=1

```

PAUSA

```

Test(pausa=0, pausa:=1, pausa:=0)
if pausa=1 then
  CallSprite(@ "anima", #stop)
else
  CallSprite(@ "anima", #play)
end if

```

PLAY

```

stop:=0
CallSprite(@ "anima", #Play)

```

STOP

```

CallSprite(, #stop)
stop:=1

OBTIENE NUMERO DE FRAME
b:= CallSprite(IconID@"anima",#currentframe)
frame:=GetLine(archivo, línea)
if frame >= (b-5) & frame <= (b +5 ) then
    mensaje[cont] := GetLine(archivo, renglon )
    definicion [cont]:= GetLine (archivo, descripcion)
--numero de frame
    línea:=línea+5
    --concepto
    renglon:=renglon+5
    --definición
    descripcion:=descripcion+5
    cont:= cont+1
end if
when true
b>=(totalframe-5) & b<=( totalframe+5)
Exit interaction

RANDOM(ASIGNA DUDAS)
--archivo_dudas := ReadExtFile (FileLocation@"textos\\d12.txt")
archivo_dudas:= ReadExtFile (FileLocation@"textos\\"^dudas".txt")
numdudas:= GetLine(archivo_dudas, 1)
ana:=1
benja:=2
chuy:=3
lalo:=4
contador:=1
--renglon:=0
obtiene_duda:= 3
repeat while contador <= 4
    personaje:=Random (1,4, 1)
    if personaje = 1 then
        if activana=0 then
            d1:=GetLine (archivo_dudas, obtiene_duda)
            respana:= obtiene_duda
            obtiene_duda := obtiene_duda+7
            dana:= d1
            contador := contador+1
            activana:= 1
        end if
    end if
    if personaje = 2 then
        if activabenja=0 then
            d1:=GetLine (archivo_dudas, obtiene_duda)
            respbenja:= obtiene_duda
            obtiene_duda := obtiene_duda+7
            dbenja:= d1
            contador := contador+1
            activabenja:=1
        end if
    end if
    if personaje = 3 then
        if activachuy= 0 then
            d1:=GetLine (archivo_dudas, obtiene_duda)
            respchuy:= obtiene_duda
            obtiene_duda := obtiene_duda+7
            dchuy:=d1
        end if
    end if
    contador:=contador+1
end repeat

```

Apéndice B. Código Fuente de los módulos.

```
        contador := contador+1
        activachuy:=1
    end if
end if
if personaje = 4 then
    if actualalo= 0then
        d1:=GetLine (archivo_dudas, obtiene_duda)
        respalo:=obtiene_duda
        obtiene_duda := obtiene_duda+7
        dlalo:=d1
        contador := contador+1
        actualalo:=1
    end if
end if
end repeat

ASIGNACION DE RESPUESTAS
if dana <> "" then
    juego:= respaña+3
    explicacion:= respaña+5
    resp:= ReadExtFile (FileLocation^"textos\\"^"dudas^".txt")
    sistema:= GetLine(resp, explicacion)
    tu:= GetLine(resp, juego)
else
    dana:= ""
end if
if dlalo <> "" then
    juego:= respalo+3
    explicacion:= respalo+5
    resp:= ReadExtFile (FileLocation^"textos\\"^"dudas^".txt")
    sistema:= GetLine(resp, explicacion)
    tu:= GetLine(resp, juego)
else
    dlalo:= ""
end if
if dbenja <> "" then
    juego:= respbenja+3
    explicacion:= respbenja+5
    resp:= ReadExtFile (FileLocation^"textos\\"^"dudas^".txt")
    sistema:= GetLine(resp, explicacion)
    tu:= GetLine(resp, juego)
else
    dbenja:= ""
end if
if dchuy <> "" then
    juego:= respchuy+3
    explicacion:= respchuy+5
    resp:= ReadExtFile (FileLocation^"textos\\"^"dudas^".txt")
    sistema:= GetLine(resp, explicacion)
    tu:= GetLine(resp, juego)
else
    dchuy:= ""
end if

ELECCION DE COMPAÑERO VIRTUAL
Conditional
ObjectClicked="AnaMano"
OnFalseToTrue
Test(sistema="",2,1)
Test(tu="",2,1)
```

Apéndice B. Código Fuente de los módulos.

DESPLIEGA JUEGO

```
SetSpriteProperty(@"juegoana", #Movie, FileLocation^"flash\\"^"tu")
CallSprite(@"juegoana", #Play)
SetSpriteProperty(@"juegoana", #scale ,100)
ENTRAR:=0
```

DESPLIEGA EXPLICACION NUEVA

```
archivo := ReadExtFile (FileLocation^"textos\\"^"sistema")
--archivo := ReadExtFile (FileLocation^"textos\\"conceptoG.txt")
totalframe:= GetLine(archivo, 1)
despliega :=GetLine(archivo, 3)
SetSpriteProperty(@"anima3", #Movie, FileLocation^"flash\\"^"despliega")
CallSprite(@"anima3", #Play)
SetSpriteProperty(@"anima3", #scale ,100)
concepto := ReadExtFile (FileLocation^"textos\\"^"sistema")
--concepto := ReadExtFile (FileLocation^"textos\\"conceptoG.txt")
linea:=4
renglon:=5
descripcion:=6
mensaje:=Array("",30)
definicion:= Array("",30)
cont:=1
b:=0
```

PAUSA

```
Test(pausa=0, pausa:=1, pausa:=0)
if pausa=1 then
  CallSprite(, #stop)
else
  CallSprite(, #play)
end if
```

PLAY

```
stop:=0
CallSprite(, #Play)
```

STOP

```
CallSprite(, #stop)
stop:=1
b:= CallSprite([conID@"anima3",#currentframe)
concepto := ReadExtFile (FileLocation^"textos\\"^"sistema")
frame:=GetLine(concepto, linea)
if frame >= (b-5) & frame <= (b +5 ) then
  texto := ReadExtFile (FileLocation^"textos\\"^"sistema")
  mensaje[cont] := GetLine(texto, renglon )
  definicion [cont]:= GetLine (texto, descripcion)
  linea:=linea+5
  renglon:=renglon+5
  descripcion:=descripcion+5
  cont:= cont+1
end if
when true
b:=(totalframe-5) & b<= (totalframe+5)
Exit interaction
```

3.-Código Fuente del juego interactivo "Mi Pueblito"

PARTE 1.

Para comenzar se establecen las características iniciales del entorno de juego.

Se ocultan los elementos que posteriormente utilizaremos:

a) Se ocultan los paréntesis.

```
setProperty ("parentesis1",_visible,false);
setProperty ("parentesis2",_visible,false);
setProperty ("parentesis3",_visible,false);
setProperty ("parentesis4",_visible,false);
setProperty ("parentesis5",_visible,false);
setProperty ("parentesis6",_visible,false);
setProperty ("parentesis7",_visible,false);
```

b) Se oculta la cuadrícula que nos ayuda a que el alumno pueda tener una unidad de referencia.

```
setProperty ("cuadrícula",_visible,false);
```

c) Se ocultan las casillas donde se leera el valor para las variables x1, x2, y1 y y2.

```
setProperty ("cero_x1",_visible,false);
setProperty ("cero_x2",_visible,false);
setProperty ("cero_y1",_visible,false);
setProperty ("cero_y2",_visible,false);
```

d) Se ocultan marcadores de respuesta correcta e incorrecta para cada lugar (objetivo)

```
setProperty ("carita_feliz_Cine",_visible,false);
setProperty ("carita_triste_Cine",_visible,false);
setProperty ("carita_feliz_Iglesia",_visible,false);
setProperty ("carita_triste_Iglesia",_visible,false);
setProperty ("carita_feliz_Escuela",_visible,false);
setProperty ("carita_triste_Escuela",_visible,false);
setProperty ("carita_feliz_Mercado",_visible,false);
setProperty ("carita_triste_Mercado",_visible,false);
setProperty ("carita_feliz_Tiendita",_visible,false);
setProperty ("carita_triste_Tiendita",_visible,false);
setProperty ("carita_feliz_CasaTL",_visible,false);
setProperty ("carita_triste_CasaTL",_visible,false);
setProperty ("carita_feliz_Panaderia",_visible,false);
setProperty ("carita_triste_Panaderia",_visible,false);
```

A continuación inicializamos variables que llevan el control de las Respuestas Correctas e Incorrectas, así como el origen de referencia (Aquí).

```
Resp_Correctas=0;
Resp_Incorrectas=0;
Aqui=0;
```

A partir de este punto se inicializan las posiciones de los lugares (objetivos) dentro de la pantalla de juego. (coordenadas x, y)

```
Resp_Iglesia_x=455;
Resp_Iglesia_y=221;

Resp_Escuela_x=329;
Resp_Escuela_y=203;
```

Apéndice B. Código Fuente de los módulos.

```
Resp_Mercado_x=221;  
Resp_Mercado_y=95;
```

```
Resp_Tiendita_x=455;  
Resp_Tiendita_y=59;
```

```
Resp_CasaTL_x=59;  
Resp_CasaTL_y=257;
```

```
Resp_Cine_x=113;  
Resp_Cine_y=167;
```

```
Resp_Panaderia_x=59;  
Resp_Panaderia_y=41;
```

Se inicializan ocultan los paréntesis

```
setProperty ("/target_item", _x, "275");  
setProperty ("/target_item", _y, "149");  
setProperty ("/target_item", _xscale, 50);  
setProperty ("/target_item", _yscale, 50);  
width_value = int(getProperty("/target_item", _width));  
height_value = int(getProperty("/target_item", _height));
```

```
distancia_x = 18;  
distancia_y = 18;
```

Estas variables contienen el valor de la posición (x,y) y los convierten en números enteros:

```
A=(int(getProperty("/target_item", _x)/distancia_x));  
B=(int(getProperty("/target_item", _y)/distancia_y));
```

Para efectos de pruebas, mando imprimir en pantalla los valores (ahora enteros) de A y B.
trace(A);
trace(B);

```
x_value = (int(getProperty("/target_item", _x)/distancia_x)-int(getProperty("/target_item",  
_x)/distancia_x));  
y_value = (int(getProperty("/target_item", _y)/distancia_y)-int(getProperty("/target_item",  
_y)/distancia_y))*(-1);
```

```
stop ();
```

FIN DE LA PARTE 1.....

PARTE 2.

No se que opinen pero para mi, la parte 2 podría ser cuando el alumno se mueve, por lo tanto, a continuación les agrego el listado de los botones de movimiento en el siguiente orden: Arriba, Abajo, Derecha e Izquierda.

ARRIBA.

```
on (release) {  
if (getProperty("/target_item",_y)<"41")  
{  
  
}  
}  
else  
{  
    setProperty ("/target_item", _y, getProperty("/target_item", _y)-distancia_y);  
    x_value = (int(getProperty("/target_item", _x)/distancia_x)- A);  
    y_value = (int(getProperty("/target_item", _y)/distancia_y)- B)*(-1);  
    trace(getProperty("/target_item",_x));  
}
```

Apéndice B. Código Fuente de los módulos.

```
        trace(getProperty("/target_item",_y));
    }
}

ABAJO.
on (release) {
if (getProperty("/target_item",_y)>"257")
{
}
else
{
    setProperty ("/target_item", _y, getProperty("/target_item", _y)+distancia_y);
    x_value = (int(getProperty("/target_item", _x)/distancia_x)- A);
    y_value = (int(getProperty("/target_item", _y)/distancia_y)- B)*(-1);
    trace(getProperty("/target_item",_x));
    trace(getProperty("/target_item",_y));
}
}

DERECHA.
on (release) {
if (getProperty("/target_item",_x)>"518")
{
}
else
{
    setProperty ("/target_item", _x, getProperty("/target_item", _x)+distancia_x);
    x_value = (int(getProperty("/target_item", _x)/distancia_x)- A);
    y_value = (int(getProperty("/target_item", _y)/distancia_y)- B)*(-1);
    trace(getProperty("/target_item",_x));
    trace(getProperty("/target_item",_y));
}
}

IZQUIERDA.
on (release) {
if (getProperty("/target_item",_x)<"36")
{
}
else
{
    setProperty ("/target_item", _x, getProperty("/target_item", _x)-distancia_x);
    x_value = (int(getProperty("/target_item", _x)/distancia_x)- A);
    y_value = (int(getProperty("/target_item", _y)/distancia_y)- B)*(-1);
    trace(getProperty("/target_item",_x));
    trace(getProperty("/target_item",_y));
}
}
}
```

PARTE 3.....

Es la parte en que se establece el origen de coordenadas, es decir, el usuario presiona el botón AQUI.

```
on (release) {
```

Nuevamente se establecen las propiedades de los objetos paréntesis, cuadrícula, marcas de correcto e incorrecto (puntos de color), etc. Pero ahora con valores de visible para que aparezcan en la pantalla.

```
setProperty ("parentesis1",_visible,false);
setProperty ("parentesis2",_visible,false);
setProperty ("parentesis3",_visible,false);
setProperty ("parentesis4",_visible,false);
setProperty ("parentesis5",_visible,false);
setProperty ("parentesis6",_visible,false);
setProperty ("parentesis7",_visible,false);

setProperty ("cuadrícula",_visible,true);

setProperty ("carita_feliz_Cine",_visible, false);
setProperty ("carita_triste_Cine",_visible, false);
setProperty ("carita_feliz_Iglesia",_visible, false);
setProperty ("carita_triste_Iglesia",_visible, false);
setProperty ("carita_feliz_Escuela",_visible, false);
setProperty ("carita_triste_Escuela",_visible, false);
setProperty ("carita_feliz_Mercado",_visible, false);
setProperty ("carita_triste_Mercado",_visible, false);
setProperty ("carita_feliz_Tiendita",_visible, false);
setProperty ("carita_triste_Tiendita",_visible, false);
setProperty ("carita_feliz_CasaTL",_visible, false);
setProperty ("carita_triste_CasaTL",_visible, false);
setProperty ("carita_feliz_Panaderia",_visible, false);
setProperty ("carita_triste_Panaderia",_visible, false);

_root.pregunta2 = " ";
_root.pregunta3 = " ";
_root.pregunta4 = " ";
_root.pregunta5 = " ";
_root.pregunta6 = " ";
_root.pregunta7 = " ";
```

También se inicializan los valores de las respuestas (coordenadas x,y) para cada caso, así como el número de respuestas correctas e incorrectas.

```
Respuesta_Iglesia_x=" ";
Respuesta_Iglesia_y=" ";
Respuesta_Escuela_x=" ";
Respuesta_Escuela_y=" ";
Respuesta_Mercado_x=" ";
Respuesta_Mercado_y=" ";
Respuesta_Tiendita_x=" ";
Respuesta_Tiendita_y=" ";
Respuesta_CasaTL_x=" ";
Respuesta_CasaTL_y=" ";
Respuesta_Panaderia_x=" ";
Respuesta_Panaderia_y=" ";

Resp_Correctas=0;
Resp_Incorrectas=0;
```

Apéndice B. Código Fuente de los módulos.

Valida que se oprima primero el botón de Aquí

```
Aquí=1;  
_root.OJO = " ";  
i=0;
```

```
A=(int(getproperty("/target_item", _x)/distancia_x));  
B=(int(getproperty("/target_item", _y)/distancia_y));
```

```
_x_value = (int(getProperty("/target_item", _x)/distancia_x)-int(getProperty("/target_item",  
_x)/distancia_x));  
_y_value = (int(getProperty("/target_item", _y)/distancia_y)-int(getProperty("/target_item",  
_y)/distancia_y))*(-1);
```

```
Respuesta_Cine_x=0;  
Respuesta_Cine_y=0;
```

```
setProperty("parentesis1",_visible,true);  
setProperty("Respuesta1",_visible,true);
```

```
_root.pregunta1 = " ¿Donde está el Cine? ";
```

Se establecen las coordenadas del nuevo origen:

```
New_origen_x = getProperty("/target_item", _x);  
New_origen_y = getProperty("/target_item", _y);
```

Se asignan las coordenadas del nuevo origen.

```
setProperty("/cero_x1", _x, New_origen_x);  
setProperty("/cero_x2", _x, New_origen_x);  
setProperty("/cero_y1", _y, New_origen_y);  
setProperty("/cero_y2", _y, New_origen_y);
```

```
setProperty ("cero_x1",_visible,true);  
setProperty ("cero_x2",_visible,true);  
setProperty ("cero_y1",_visible,true);  
setProperty ("cero_y2",_visible,true);
```

Se restan a las coordenadas del nuevo origen las coordenadas de la respuesta correcta.

```
Respuesta_Cine_x2 = int((New_origen_x - Resp_Cine_x)*(-1));    Respuesta_Cine_y2 =  
int(New_origen_y - Resp_Cine_y);  
trace("Resp");  
trace(int(Respuesta_Cine_x2/18));  
trace(int(Respuesta_Cine_y2/18));  
}
```

PARTE 4.....

Es la parte en que se registra la respuesta que brindó del usuario.

```
on (press)  
{
```

Valida que primero oprima el botón de Aquí.

```
f (Aquí==0) {_root.OJO = "OJO Primero tienes que establecer tu origen"}
```

```
if (Aquí==1)
```

```
{  
    i=i+1;
```

```

    if (i==1)
    {
        _root.OJO = " ";

        Se convierte la respuesta del usuario a pixeles para comparar los resultados.

        Usuario_Cine_x = int(Respuesta_Cine_x * distancia_x);
        Usuario_Cine_y = int(Respuesta_Cine_y * distancia_y); //
        if (Respuesta_Cine_x2==Usuario_Cine_x &&
Respuesta_Cine_y2==Usuario_Cine_y)
        {
            carita_Feliz_Cine._visible = true;
            Resp_Correctas=Resp_Correctas+1;
            carita_triste_Cine._visible = false;
        }

        else

        {
            carita_Feliz_Cine._visible = false;
            carita_triste_Cine._visible = true;
            Resp_Incorrectas=Resp_Incorrectas+1;
        }

        Desde aquí se inicializa la siguiente pregunta

        setProperty("parentesis2",_visible,true);
        _root.pregunta2 = " ¿Donde está la Iglesia? ";
        setProperty("Respuesta2",_visible,true);

        Respuesta_Iglesia_x=0;
        Respuesta_Iglesia_y=0;

        Se establecen las coordenadas del nuevo origen
        New_origen_x = getProperty("/target_item", _x); New_origen_y =
getProperty("/target_item", _y);

        Se restan a las coordenadas del nuevo origen las coordenadas de la respuesta correcta

        Respuesta_Iglesia_x2 = int((New_origen_x - Resp_Iglesia_x)*(-1)); Respuesta_Iglesia_y2 =
int((New_origen_y - Resp_Iglesia_y);

        trace("Resp");
        trace(int(Respuesta_Iglesia_x2/18));
        trace(int(Respuesta_Iglesia_y2/18));
    }

    if( i==2 )
    {
        Se convierte la respuesta del usuario a pixeles para comparar los resultados

        Usuario_Iglesia_x = int (Respuesta_Iglesia_x * distancia_x);
        Usuario_Iglesia_y = int (Respuesta_Iglesia_y * distancia_y);
        if (Respuesta_Iglesia_x2==Usuario_Iglesia_x &&
Respuesta_Iglesia_y2==Usuario_Iglesia_y)
        {
            carita_Feliz_Iglesia._visible = true;
            Resp_Correctas=Resp_Correctas+1;
            carita_triste_Iglesia._visible = false;
        }
    }

```

Apéndice B. Código Fuente de los módulos.

```
else
{
    carita_Feliz_Iglesia._visible = false;
    carita_triste_Iglesia._visible = true;
    Resp_Incorrectas=Resp_Incorrectas+1;
}

setProperty("parentesis3",_visible,true);
_root.pregunta3 = "¿Donde está la Escuela? ";

Respuesta_Escuela_x=0;
Respuesta_Escuela_y=0;

Se establecen las coordenadas del nuevo origen

New_origen_x = getProperty("/target_item",_x);           New_origen_y =
                getProperty("/target_item",_y);

Se restan a las coordenadas del nuevo origen las coordenadas de la respuesta
correcta
Respuesta_Escuela_x2 = int((New_origen_x - Resp_Escuela_x)*(-1));
Respuesta_Escuela_y2 = int(New_origen_y - Resp_Escuela_y);
trace("Resp");
trace(int(Respuesta_Escuela_x2 / 18));
trace(int(Respuesta_Escuela_y2 / 18));
}

if( i==3 )
{
    Se convierte la respuesta del usuario a pixeles para comparar los resultados
    Usuario_Escuela_x = Respuesta_Escuela_x * distancia_x;
    Usuario_Escuela_y = Respuesta_Escuela_y * distancia_y;

    if (Respuesta_Escuela_x2==Usuario_Escuela_x &&
Respuesta_escuela_y2==Usuario_Escuela_y)
    {
        carita_feliz_Escuela._visible = true;
        Resp_Correctas=Resp_Correctas+1;
        carita_triste_Escuela._visible = false;
    }

    else
    {
        carita_feliz_Escuela._visible = false;
        carita_triste_Escuela._visible = true;
        Resp_Incorrectas=Resp_Incorrectas+1;
    }

    setProperty("parentesis4",_visible,true);
    _root.pregunta4 = "¿Donde está el Mercado? ";

Respuesta_Mercado_x=0;
Respuesta_Mercado_y=0;
```

Apéndice B. Código Fuente de los módulos.

```
Se establecen las coordenadas del nuevo origen
New_origen_x = getProperty("/target_item", _x);
New_origen_y = getProperty("/target_item", _y);

Se restan a las coordenadas del nuevo origen las coordenadas de la respuesta
correcta
Respuesta_Mercado_x2 = int((New_origen_x - Resp_Mercado_x)*(-1));
Respuesta_Mercado_y2 = int(New_origen_y - Resp_Mercado_y);
trace("Resp");
    trace(int(Respuesta_Mercado_x2/18));
    trace(int(Respuesta_Mercado_y2/18));
// trace("Resp");
// trace(Resp_Incorrectas);
// trace(Resp_Correctas);
}

if( i==4 )
{
    Se convierte la respuesta del usuario a pixeles para comparar los resultados
    Usuario_Mercado_x = Respuesta_Mercado_x * distancia_x;
    Usuario_Mercado_y = Respuesta_Mercado_y * distancia_y;

    if (Respuesta_Mercado_x2==Usuario_Mercado_x &&
Respuesta_Mercado_y2==Usuario_Mercado_y)
    {
        carita_Feliz_Mercado._visible = true;
        Resp_Correctas=Resp_Correctas+1;
        carita_triste_Mercado._visible = false;
    }
    else
    {
        carita_Feliz_Mercado._visible = false;
        carita_triste_Mercado._visible = true;
        Resp_Incorrectas=Resp_Incorrectas+1;
    }

    setProperty("parentesis5",_visible,true);
    _root.pregunta5 = " ¿Donde está la Tiendita? ";

    Respuesta_Tiendita_x=0;
    Respuesta_Tiendita_y=0;

    Se establecen las coordenadas del nuevo origen

    New_origen_x = getProperty("/target_item", _x);
    New_origen_y =
getProperty("/target_item", _y);

correcta
    Respuesta_Tiendita_x2 = int((New_origen_x - Resp_Tiendita_x)*(-1));
    Respuesta_Tiendita_y2 = int(New_origen_y - Resp_Tiendita_y);
    trace("Resp");
    trace(int(Respuesta_Tiendita_x2/18));
    trace(int(Respuesta_Tiendita_y2/18));
```

Apéndice B. Código Fuente de los módulos.

```
    }  
  
    if( i==5 )  
    {  
        Se convierte la respuesta del usuario  
  
        Usuario_Tiendita_x = Respuesta_Tiendita_x * distancia_x;  
        Usuario_Tiendita_y = Respuesta_Tiendita_y * distancia_y;  
  
        if (Respuesta_Tiendita_x2==Usuario_Tiendita_x &&  
Respuesta_Tiendita_y2==Usuario_Tiendita_y)  
        {  
            carita_Feliz_Tiendita._visible = true;  
            Resp_Correctas=Resp_Correctas+1;  
            carita_triste_Tiendita._visible = false;  
        }  
        else  
        {  
            carita_feliz_Tiendita._visible = false;  
            carita_triste_Tiendita._visible = true;  
            Resp_Incorrectas=Resp_Incorrectas+1;  
        }  
  
        setProperty("parentesis6",_visible,true);  
        _root.pregunta6 = " ¿Donde está la Panaderia? ";  
  
        Respuesta_Panaderia_x=0;  
        Respuesta_Panaderia_y=0;  
  
        Se establecen las coordenadas del nuevo origen  
  
        New_origen_x = getProperty("/target_item", _x);  
        New_origen_y = getProperty("/target_item", _y);  
  
        Se restan a las coordenadas del nuevo origen las coordenadas de la respuesta  
correcta  
        Respuesta_Panaderia_x2 = int((New_origen_x - Resp_Panaderia_x))*(-1);  
        Respuesta_Panaderia_y2 = int((New_origen_y - Resp_Panaderia_y);  
        trace("Resp");  
        trace(int(Respuesta_Panaderia_x2/ 18));  
        trace(int(Respuesta_Panaderia_y2/ 18));  
    }  
  
    if( i==6 )  
    {  
        Se convierte la respuesta del usuario  
  
        Usuario_Panaderia_x = Respuesta_Panaderia_x * distancia_x;  
        Usuario_Panaderia_y = Respuesta_Panaderia_y * distancia_y;  
  
        if (Respuesta_Panaderia_x2==Usuario_Panaderia_x &&  
Respuesta_Panaderia_y2==Usuario_Panaderia_y)  
        {
```

Apéndice B. Código Fuente de los módulos.

```
        carita_feliz_Panaderia._visible = true;
        Resp_Correctas=Resp_Correctas+1;
        carita_triste_Panaderia._visible = false;
    }

    else
    {
        carita_feliz_Panaderia._visible = false;
        carita_triste_Panaderia._visible = true;
        Resp_Incorrectas=Resp_Incorrectas+1;
    }

        setProperty("parentesis7",_visible,true);
        _root.pregunta7 = "¿Donde está la Casa de la Tia Lencha? ";

Respuesta_CasaTL_x=0;
Respuesta_CasaTL_y=0;

    Se establecen las coordenadas del nuevo origen
    New_origen_x = getProperty("/target_item", _x);
    New_origen_y = getProperty("/target_item", _y);

    Se restan a las coordenadas del nuevo origen las coordenadas de la respuesta
correcta
    Respuesta_CTL_x2 = int((New_origen_x - Resp_CasaTL_x)*(-1);
Respuesta_CTL_y2 = int(New_origen_y - Resp_CasaTL_y);
    trace("Resp");
    trace(int(Respuesta_CTL_x2/18));
    trace(int(Respuesta_CTL_y2/18));
}

if( i==7 )
{
    Se convierte la respuesta del usuario a pixeles para comparar los resultados
    Usuario_CasaTL_x = Respuesta_CasaTL_x * distancia_x;
    Usuario_CasaTL_y = Respuesta_CasaTL_y * distancia_y;

    if (Respuesta_CTL_x2==Usuario_CasaTL_x &&
Respuesta_CTL_y2==Usuario_CasaTL_y)
    {
        carita_Feliz_CasaTL._visible = true;
        Resp_Correctas=Resp_Correctas+1;
        carita_triste_CasaTL._visible = false;
    }

    else
    {
        carita_Feliz_CasaTL._visible = false;
        carita_triste_CasaTL._visible = true;
        Resp_Incorrectas=Resp_Incorrectas+1;
    }

    if (Resp_Correctas<Resp_Incorrectas)
```

```

    { _root.OJO="Estas seguro que no quieres preguntarle al profe? ";
    }
    else
    { _root.OJO2="Muy bien!!! ";
    }
}
}
}

```

PARTE 5.....

Esta parte no esta terminada, pero se supone que sería el botón de salir.

```

on (press)
{
//stop
//quit{};
//break;
//exit;
stop {};
}

```

4.-Código Fuente del juego interactivo "Morita la Pulguita"

```

opcionA:=0
opcionB:=0
indice:=0
suma:=0
if opcionA= 0 then
    opcionA:= 1
    indice1 := 1
    indice:= indice1 + indice2 + indice3 + indice4 + indice5
    dibujo := "C:\\my documents\\Mis
documentos\\martha\\prototipos\\Dibujos\\"^indice^".gif"
else
    opcionA:= 0
    indice1:= 0
    indice:= indice1 + indice2 + indice3 + indice4 + indice5
    dibujo := "C:\\my documents\\Mis
documentos\\martha\\prototipos\\DIBUJOS\\"^indice^".gif"
end if
if opcionB= 0 then
    opcionB:= 1
    indice2 := 2
    indice:= indice1 + indice2 + indice3 + indice4 + indice5
    dibujo := "C:\\my documents\\Mis
documentos\\martha\\prototipos\\DIBUJOS\\"^indice^".gif"
else
    opcionB:= 0
    indice2:= 0
    indice:= indice1 + indice2 + indice3 + indice4 + indice5
    dibujo := "C:\\my documents\\Mis
documentos\\martha\\prototipos\\DIBUJOS\\"^indice^".gif"
end if
if opcionC= 0 then

```

Apéndice B. Código Fuente de los módulos.

```
opcionC:= 1
indice3 := 4
indice:= indice1 + indice2 + indice3 + indice4 + indice5
dibujo := "C:\my documents\Mis
documentos\martha\prototipos\DIBUJOS\^"indice^".gif"
else
opcionC:= 0
indice3:= 0
indice:= indice1 + indice2 + indice3 + indice4 + indice5
dibujo := "C:\my documents\Mis
documentos\martha\prototipos\DIBUJOS\^"indice^".gif"

end if
if opcionD= 0 then
opcionD:= 1
indice4 := 8
indice:= indice1 + indice2 + indice3 + indice4 + indice5
dibujo := "C:\my documents\Mis
documentos\martha\prototipos\DIBUJOS\^"indice^".gif"
else
opcionD:= 0
indice4 := 0
indice:= indice1 + indice2 + indice3 + indice4 + indice5
dibujo := "C:\my documents\Mis
documentos\martha\prototipos\DIBUJOS\^"indice^".gif"

end if
if opcionE= 0 then
opcionE := 1
indice5 := 16
indice:= indice1 + indice2 + indice3 + indice4 + indice5
dibujo := "C:\my documents\Mis
documentos\martha\prototipos\DIBUJOS\^"indice^".gif"
else
opcionE := 0
indice5 := 0
indice:= indice1 + indice2 + indice3 + indice4 + indice5
dibujo := "C:\my documents\Mis
documentos\martha\prototipos\DIBUJOS\^"indice^".gif"

end if
if inicio = 0 then
indice:= indice1 + indice2 + indice3 + indice4 + indice5
--mensaje := ReadExtFile(FileLocation^"mensajes\mensaje^"indice^".txt")

mensaje:= ReadExtFile(FileLocation^"mensajes\mensajehora.txt")
despliega := GetLine( mensaje, indice + 1)
--dibujo := "c:\Mis Documentos\martha\dibujos\^"indice^".jpg"

else
despliega := ""
end if
```

Guión del experto Dr. Fermín Viniestra Heberlein.

1 LA PRIMERA LEY DE LA MECÁNICA

1.1 LOS MARCOS DE REFERENCIA INERCIALES

1.1.1 LOS SISTEMAS DE COORDENADAS

1.1.1.1 La recta numérica fue introducida por René Descartes en el siglo XVII. Se trata de asignar valores numéricos a las distancias que se miden en una recta numérica, partiendo de un punto de ella que se le llama el origen y se denota por O. A la derecha de ese origen y a intervalos regulares se indica mediante una marca las distancias 1,2,3,4, ...etc desde el origen en algún sistema de unidades. Todas estas distancias, medidas a la derecha del origen se dice que son positivas. A la izquierda del origen =, en la recta numérica se marcan, igualmente, a intervalos regulares las distancias -1, -2, -3, -4, ... etc. Se acostumbra decir que estas distancias son negativas.

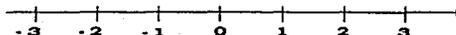


Fig. 1.1.1.1 La recta numérica.

Se acostumbra también indicar mediante una punta de flecha en el extremo derecho de la recta numérica, la dirección en que crecen los valores. Cualquier valor de una distancia se puede hallar sobre la recta numérica. Por ejemplo, el valor -2.5 se encuentra a la izquierda del origen; a la izquierda del valor -2 entre este valor y la marca que indica -3.

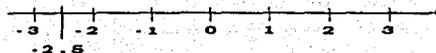


Fig. 1.1.1.2. Sobre la recta numérica se encuentra el valor -2.5
Cualquier valor, tanto positivo como negativo se encuentra sobre la recta numérica

1.1.1.2 Con dos rectas numéricas que se dibujan, una perpendicular a la otra y cuyos orígenes coinciden en el punto de intersección de ambas, se construye un SISTEMA DE COORDENADAS EN DOS DIMENSIONES.

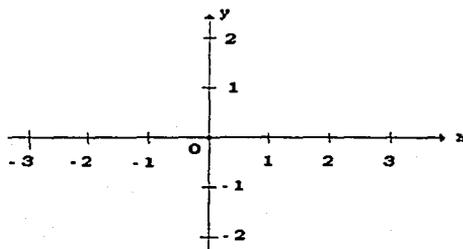


Fig. 1.1.1.3 Un sistema de coordenadas en dos dimensiones. A la recta numérica horizontal se le llama eje de las abscisas y a la recta numérica vertical se le llama eje de las ordenadas.

A estas dos rectas numéricas se le llama ejes. Al eje horizontal se le conoce como **eje de las abscisas**, en tanto que al vertical se le da el nombre de **eje de las ordenadas**.

También es costumbre que al eje de las abscisas, por facilidad se le denota mediante la letra x , en tanto que al eje de las ordenadas se le adscribe la letra y .

Un sistema de coordenadas permite ubicar puntos sobre un plano y asociar a cada uno de ellos, un par de números ordenados, se les llama las **coordenadas del punto en el plano**.

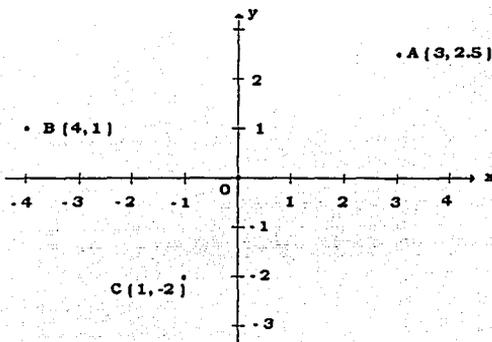


Fig. 1.1.1.4.: Tres puntos sobre el plano se ubican con respecto a un sistema de coordenadas de dos dimensiones. Estos puntos son $A(3, 2.5)$, $B(-4, 1)$, $C(-1, -2)$.

Pero además de puntos, se pueden trazar también curvas sobre el plano. No hay que olvidar que una línea no es otra cosa que una sucesión de puntos. Y los puntos se pueden ubicar y si una línea es una sucesión de punto, entonces las líneas se pueden describir en el plano.

Una función es una expresión matemática de la forma

$$y = f(x)$$

donde x es la variable independiente y y es la variable dependiente. A la variable independiente se le dan valores correspondientes. Por ejemplo, una función puede ser la siguiente:

Este es un ejemplo de una función. Si se le dan valores a la variable independiente; por ejemplo $-4, -2, 0, 2, 4, \dots$ etc., la variable dependiente va adquiriendo valores también:

x	$f(x) = \frac{1}{2}x + 1$	y
-4	$\frac{1}{2}(-4) + 1$	-1
-2	$\frac{1}{2}(-2) + 1$	0
0	$\frac{1}{2}(0) + 1$	1
2	$\frac{1}{2}(2) + 1$	2
4	$\frac{1}{2}(4) + 1$	3

Así, al dar valores a la variable independiente x , se obtienen valores de la variable dependiente y . Esto se puede visualizar como un conjunto de puntos aun respecto al sistema de coordenadas:

$(-4, -1), (-2, 0), (0, 1), (2, 2), (4, 3) \dots$

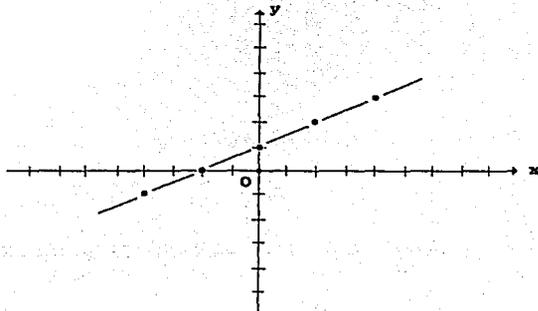


Fig. 1.1.1.5.: Una función $y=f(x)$ da como resultado un conjunto de puntos que generan una línea.

Lo realmente importante de estos conceptos para la física es que en el Universo, trayectorias como estas se pueden observar con frecuencia. Si se observa la foto de un asteroide, tomado desde alguno de los poderosos telescopios que hay actualmente en los observatorios del mundo, se advierte que ejecuta un movimiento muy parecido al que se muestra en la figura 1.1.1.5. Así, la geometría desarrollada por Descartes hace ya tantos años resulta ser una herramienta que permite la observación y el estudio sistemático de los movimientos en la Naturaleza.

Por supuesto, esta ayuda no nada más se aplica a asteroides observados al través de telescopios, sino a muchos, otros ejemplos de movimiento de cuerpos materiales. También debe ser claro que no nada más las rectas son las funciones que se pueden graficar en un sistema de coordenadas de dos dimensiones.

Los sistemas de coordenadas pueden ser también de tres dimensiones. Para construirlas se añade una recta más a las dos perpendiculares, que tenían en el caso dos dimensional. Esta recta numérica adicional debe hacerse coincidir con las otras dos en el origen O, pero debe ser perpendicular a ellas, tal como se ve en la figura 1.1.1.6.

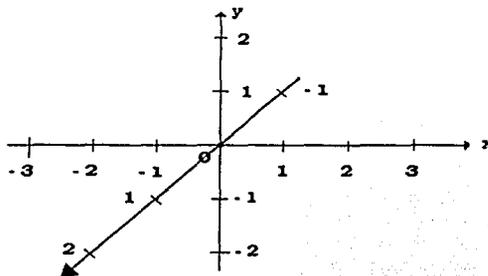


Fig. 1.1.1.6.: Un sistema de coordenadas de tres dimensiones.

Preguntas:

- 1) Define tal como tu lo hayas entendido, qué es la recta numérica
- 2) ¿Qué significa una distancia negativa? ¿Podrías dar algún ejemplo de una distancia negativa?

Dudas:

¿Entonces si yo estoy en el centro de un pueblo mirando el amanecer, puedo decir que el sol apareció por el Este, detrás de los cerros y que están a +1 km y por la tarde, cuando el sol desaparezca en el Poniente, lo veremos por última vez atrás del edificio de la escuela, que está a -200m del centro?

¿No se van a reír de mí? ¿No es más fácil decir que el cerro está a 18 km al este y la escuela a 200m al oeste del centro? ¿Qué ventajas tiene nombrar los números así?

Respuestas:

Un alumno: Mira, si tu estás en tu pueblo lo mejor es que no les hables de números positivos o negativos. Los vas a asustar. Pero si te encuentras en medio del desierto, donde no tengas puntos de referencia, más que la salida o puesta del sol, o si estás en el mar, te puede ayudar. Así se ve en las películas.

Maestro:

La ventaja de la recta numérica es que la puedes dibujar sobre una hoja de papel y en cualquier dirección. Lo único que necesitas, aparte del papel es un lápiz y una regla.

Preguntas:

- 3) ¿Para qué sirve dibujar puntos en el plano cartesiano? ¿Podrías dar algún ejemplo de este procedimiento?

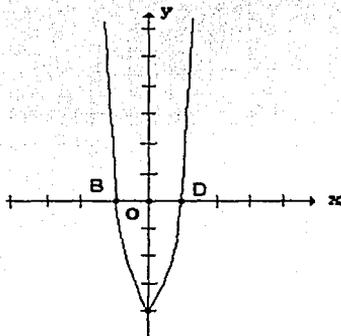
R: Cuando jugamos a los submarinos podemos hallar el barco enemigo usando sus coordenadas.

- 4) ¿Si te dan una función $y=f(x)$, donde $f(x)=4x-4$ puedes hacer una tabla como en el ejemplo anterior, hallar los puntos en el plano coordenado y luego dibujar la curva que corresponde a esta función?

R:

x	$f(x) = 4x - 4$	y
-3	$4(-3) - 4$	32
-2	$4(-2) - 4$	12
-1	$4(-1) - 4$	0
0	$4(0) - 4$	-4
1	$4(1) - 4$	0
2	$4(2) - 4$	12
3	$4(3) - 4$	32

Los puntos que se van a dibujar son entonces A(-2,12), B(-1,0), C(0,-4), D(1,0), E(2,12)



Esta curva es una parábola

Dudas:

¿Cualquier fórmula que se me ocurra la puedo convertir en una figura con este mismo procedimiento de hacer primero una tabla, dar valores a la variable independiente, calcular los valores a la variable dependiente, formar las parejas de valores y dibujarlos como puntos en el plano coordenado?

Respuestas:

Siempre lo puedes hacer cuando tienes una función de x y la igualas a " y ". Hay algunos casos muy complicados que te costará mucho trabajo hacerlos porque la fórmula puede ser muy difícil, pero con las modernas computadoras, casi siempre se puede.

Dudas:

¿Y a mí de qué me servirá saber todo esto?

Respuesta:

Si tu vas a ser un ingeniero, o un astrónomo, o bien si tu vas a convertirte en marino a piloto de aviones, o si vas a dedicarte a los negocios y quieres saber como van tus pérdidas y ganancias, tendrás que manejar esto.

¿Y si lo que quiero ser cuando sea grande es abogado?

Profesor:

Hoy en día casi no queda un solo rincón de la actividad humana donde no se utilicen las matemáticas. Aun los abogados a los médicos o los psicólogos y filósofos que tanto habían esquivado el uso de las matemáticas hoy en día tienen que aprenderlas para ser competitivos en sus profesiones.

Si vas a ser abogado tendrás que llevar, necesariamente, un registro de tus casos ganados y de los perdidos. Entonces será muy importante que dibujes gráficas como éstas para visualizar tu actividad.

Bibliografía.

(1995). Diccionario de las Ciencias de la Educación. México, D.F., Editorial Santillana.

D, A. (1987). "What century is this anyway?": 26,27,28.

Kemper Valverde, N. (1997). Manual de Sistemas Expertos. Lima-Perú.

López Ostio J. Sistemas Tutoriales Inteligentes (ITS). Conferencia mecanografiada. San Sebastián, España: 1993.

Martínez, A. (1991). Inteligencia Artificial, La Gran Guía. Madrid, Grupo Editorial Jackson.

Nebendahl, D. (1988). Sistemas Expertos. Barcelona, Marcombo.

Newman, B. D. (1996). Knowledge Management vs Knowledge Engineering.

Resselman, B. and R. Peasley (1999). Descubre Visual Basic 6.0. Madrid, España, Prentice Hall.

Rolston, D. W. (1992). Principios de Inteligencia Artificial y Sistemas Expertos. México, D.F., McGraw-Hill.

Páginas Visitadas:

[Asesores Inteligentes Para Apoyar El Proceso De Enseñanza De Lenguajes De Programación]
www.sinergia-web.com.mx/clases/tesis/propuesta.htm#Figura3

[Asesores Inteligentes Para Apoyar El Proceso De Enseñanza De Lenguajes De Programación]
www.quadernsdigitals.net/articuloquaderns.asp?idArticle=3743

[Aspectos de la Inteligencia Artificial]
www3.gratisweb.com/comarsc/inar2.htm

[BearEdu Technologies]
bearedu.com

[Comp.lang.prolog Frequently Asked Questions]
www.cis.ohio-state.edu/hypertext/faq/usenet/prolog/faq/faq.html

[De la educación Tradicional a la educación a distancia: cambio de paradigma]
www.uls.edu.mx/public_html/publicaciones/onteanqui/b7/paradigma2.html

[Edward Fredkin Los comienzos de la Inteligencia Artificial]
web1.cti.unav.es/asignaturas/ia/tsld002.htm

[Evolución de la enseñanza asistida por computadoras]
bvs.sld.cu/revistas/ems/vol11_1_97/ems05197.htm

[Genéticos Algoritmos]
www.cinefantastico.com/nexus7/ia/ageneticos.htm

[H., G. P. A. "Estrategias Heurísticas para el uso de Computadores para Enriquecer la Educación." 6: 12]
vig.prenhall.com/acadbook/0.2581.0132581.0132541033.00.html

[Inteligencia Artificial]
ciberconta.unizar.es/LECCION/IA/000F2.HTM

[Inteligencia Artificial]
w3.mor.itesm.mx/~rdec/node11.html

[Inteligencia Artificial]
www.geocities.com/NapaValley/4376/ia.htm

[INTELIGENCIA ARTIFICIAL: ¿DE QUÉ SE TRATA?]
cariari.ucr.ac.cr/~claudiog/Inteligencia_artificial.html

[Interactive Physics]
www.softwreshop.com/Productos/Interactive_Physics/interactive_physics.html

[Knowledgebase™ Series]
www.knowledgefactory.com

[Nuevas Herramientas Educativas: Sistema Integral de Prácticas de Laboratorio Asistidas por Computador]
www.c5.cl/ieinvestiga/actas/ribie96/EDSIPLAC.html

[Play Physics]

www.i-t.com/engsw/jason/pp2.htm

[Sistemas Tutores Inteligentes]

www.sinergia-web.com.mx/clases/tesis/stis.htm

[The Ultimate Einstein]

www.cdromshop.com/cdshop/desc/p.793763000218.html

[Visual Quantum Mechanical]

phys.educ.ksu.edu/