



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CONTADURIA Y ADMINISTRACION

**INFORMATICA EDUCATIVA:
DISEÑO E IMPLEMENTACION DE UN MODELO
PARA EL MANEJO DE FLUJOS DE
INFORMACION Y CONTROL EN UN
SISTEMA TUTORIAL**

SEMINARIO DE INVESTIGACION INFORMATICA

QUE EN OPCION AL GRADO DE
LICENCIADO EN INFORMATICA

P R E S E N T A N :

MONICA ARAOZ CASTILLO

RUBEN CAMPOS CAZARES

JORGE OROPEZA AVILA

DIRECTOR: DR. ANGEL DAVID GUEVARA POZAS

TESIS CON
FALTA DE ORIGEN

1992



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 está 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.

**INFORMATICA EDUCATIVA:
DISEÑO E IMPLEMENTACION DE UN MODELO PARA EL MANEJO DE FLUJOS DE
INFORMACION Y CONTROL EN UN SISTEMA TUTORIAL**

INDICE	Pag.
RESUMEN.	I
1. INTRODUCCION.	1
2. REVISION DE LA LITERATURA.	5
2.1 Flujos de Información y Control en Sistemas Educativos.	5
2.2 Tipos de Software Educativo.	8
2.3 Herramientas para la construcción de Software Educativo.	12
2.4 Metodología para el desarrollo de Sistemas Tutoriales.	16
2.5 Control en los sistemas de Instrucción por computadora.	23
2.6 Automatas de Estados Finitos.	25
3. MATERIAL Y METODOS.	28
4. RESULTADOS.	29
4.1 Esquema General sobre el Flujo y Transformación de la Información para la Construcción de Software Educativo.	29
4.2 Modelo para el manejo de flujos de Información y Control en un Sistema Tutorial.	32
4.3 Implementación del modelo.	35
4.3.1 Estructura del contenido temático en un Sistema Tutorial.	41
4.3.2 Un Automata de Estados Finitos en la definición de flujos de información y control.	43
5. DISCUSION.	47
5.1 La computadora en relación a otros medios educativos.	47
5.2 Los flujos de información y control en diversas implementaciones físicas.	48
5.3 Aplicación del modelo a la descripción de flujos en Juegos Educativos, Simuladores y Leccionarios.	50
5.4 La incorporación de la Instrucción Asistida por Computadora al proceso educativo.	52
6. CONCLUSIONES.	56
7. PROPUESTAS SOBRE INVESTIGACIONES FUTURAS.	58
APENDICES.	
BIBLIOGRAFIA.	

RESUMEN.

Las nuevas alternativas que provee la tecnología nos permiten asociar dos áreas de conocimiento inicialmente desconexas: la Educación y la Informática. La Instrucción Asistida por Computadora es el punto de convergencia entre ellas ya que se presenta como una opción viable para resolver problemas específicos en el ambiente educativo nacional. Desafortunadamente, en México no existen metodologías para el desarrollo de Software Educativo (SE) que respondan a las expectativas de nuestros educandos, ni se utilizan suficientemente las herramientas disponibles para este fin. En este trabajo: a) se identifican las fases de incorporación y transformación de la información durante la generación de software educativo; b) sobre esta base, se presenta un modelo que define tres mecanismos interactuantes en un sistema tutorial y se identifican las direcciones de los flujos de información y control que se establecen entre los mecanismos y; c) se aplica la teoría de autómatas de estados finitos como principio básico de ingeniería para implementar la función de control. Las estructuras teórico-conceptuales resultantes se aplican en un desarrollo concreto: el Sistema Tutorial ATOMO. Concluimos que la Instrucción debe ser asistida por computadora y que el ejercicio educativo necesita imperiosamente de profesionales que dediquen esfuerzos para promover el uso de la computadora como herramienta que conforme una nueva política educativa.

1. INTRODUCCION.

En este trabajo se interrelacionan dos áreas de conocimiento y desarrollo inicialmente desconexas: la educación y el ejercicio informático.

La Educación es un propósito social presente en todas las culturas. Su importancia se destaca en tanto que su misión es la preparación intelectual y social de los individuos con el fin de incorporarlos a las relaciones que ocurren en su comunidad.

Una concepción relevante en torno a la educación señala que, como receptor del ejercicio educativo, una persona encuentra su "*vocación verdadera*" y contribuye así, al desarrollo de la polis (Sócrates en: Zea, 1989). Bajo este planteamiento nosotros percibimos a la educación como el proceso que provee a cada individuo con los elementos esenciales para su desarrollo y le otorga, por tanto, la potencialidad de promover el desarrollo económico, social y cultural de su país.

La educación en nuestro país atraviesa por un estado crítico que se caracteriza por:

1. **Marginación social de un alto número de mexicanos.** Según datos tomados de Cueli, *et al* (1991), un millón de indígenas no hablan español, hay seis millones de adultos analfabetos, dos millones de niños en edad escolar no tienen acceso a la primaria y 26 millones de mexicanos no terminaron la primaria.

2. **Pauperización de los recursos dedicados a la educación.** En 1960 la fracción del Producto Interno Bruto (PIB) que se destinó a propósitos educativos fue del 1.2%. En 30 años este porcentaje se ha incrementado sólo en 1.6% (NAFINSA, 1990), mientras que la UNESCO recomienda que esta fracción del PIB sea del 8% (Méndez, 1991). Como consecuencia, existe un exceso de alumnos asignados a cada clase, escasez o ausencia

de material didáctico, carencia de recursos materiales y humanos en el campo, profesores improvisados, etc.

3. **Problemas magisteriales.** Estos van desde una pesada burocracia educativa, hasta raquílicas remuneraciones.

4. **Falta de continuidad en los planes y programas educativos.** Por ejemplo: La Educación Positivista de Gabino Barrera, el Renacimiento Educativo de Vasconcelos, la Educación Ultraica de Bassols y el Liberalismo Educativo impulsado en la presente administración.

Esta problemática ha promovido un interés en los grupos universitarios, de investigación y desarrollo, por generar alternativas metodológicas y técnicas que mejoren la calidad de la educación. Esto ha dado origen a la aplicación de diversos medios que apoyen la impartición de una clase, como el video con fines educativos, los sistemas audiovisuales y las teleconferencias, entre otros. En este sentido, la **Informática surge como una poderosa alternativa que enriquece este universo de opciones.**

La Informática nació como una herramienta para facilitar aplicaciones militares (Sanders, 1985), posteriormente algunas empresas vieron en ella una forma de sistematizar el control de sus crecientes volúmenes de información. Dadas la capacidad de almacenamiento, velocidad de procesamiento y facilidad en la transmisión de datos que proveen el Hardware y Software, la Informática tiene en el área de negocios una plataforma "natural" de aplicación. Sin embargo, en este momento el procesamiento automático de la información está presente en todos los campos de conocimiento. Vemos así que la computadora -herramienta informática- se utiliza como medio de procesamiento de datos para múltiples tareas entre las que destacan: industriales, procesos robotizados de manufactura, etc.; clínicas, análisis químicos, sistemas expertos de diagnóstico, etc.; científicas, cálculo numérico; económicas, simulaciones en modelos econométricos, y; educativas.

La Informática aplicada a la Educación genera un nuevo paradigma educativo: Instrucción Asistida por Computadora (IAC) donde el instructor está representado por una computadora alimentada con software *ex profeso*.

En México, es patente la falta de estrategias informáticas para el desarrollo de sistemas para IAC. En su mayoría, dichos desarrollos siguen un enfoque completamente heurístico y, en el mejor de los casos, utilizan metodologías extranjeras.

Con el propósito de establecer una primera vía de análisis del proceso educativo, nuestro grupo de investigación ha adoptado descripciones operativas para los términos de *instrucción* y formación. Instrucción describe la transferencia de conocimiento específico (con contenido temático) del educador (instructor) hacia el educando, a través de flujos de información y control entre ambos y; *formación* designa la generación de criterios y actitudes en el educando que le permitan contender con situaciones complejas en su entorno (Araoz *et al*, 1991; Guevara-Pozas, 1992).

De esta manera, nos hemos propuesto generar un modelo teórico que maneje los flujos de información y control en un sistema tutorial y aplicarlo en un desarrollo concreto. Pretendemos que este modelo sienta las bases para la elaboración ulterior de una metodología de construcción de sistemas para IAC.

Los objetivos específicos que persigue este trabajo son:

1. Identificar las fases de incorporación y transformación de la información durante la generación de software educativo.
2. Generar una estructura teórico-conceptual que defina los mecanismos interactuantes en un sistema tutorial y las vías de interacción entre los mismos mecanismos.
3. Explorar un desarrollo teórico que pueda utilizarse como un principio básico de ingeniería de software para controlar un sistema tutorial.

4. Implementar interfases entre herramientas de construcción de software educativo, que permitan aprovechar las facilidades que cada una de ellas presenta.

En la segunda parte de este trabajo presentamos el conocimiento generado en torno a: a) el establecimiento de flujos de información y control en sistemas educativos y; b) los diversos elementos para el desarrollo de software educativo. Examinamos también la función de control para sistemas de instrucción asistida por computadora, específicamente en sistemas tutoriales. Y revisamos los fundamentos teóricos de un modelo matemático, ya reconocido como un controlador simple y poderoso: el autómata de estados finitos.

En la tercera parte, mencionamos los materiales y métodos que se utilizaron para el desarrollo de la presente tesis.

Los resultados derivados de la investigación se presentan en la cuarta sección donde se incluye al sistema tutorial **ATOMO**. Este sistema trata sobre la historia y desarrollo de las teorías acerca de la composición de la materia.

En la quinta parte se discute sobre la aplicación de la computadora en relación a otros medios educativos y sobre la generalización del modelo, para el manejo de flujos de información y control en un sistema tutorial, hacia otros tipos de software educativo.

La sexta parte contiene las conclusiones derivadas de la presente investigación y, finalmente, proponemos algunos aspectos que pueden desarrollarse a futuro.

Una breve reseña de los acontecimientos más relevantes en materia educativa en nuestro país se integra en el Apéndice A. Y el Apéndice B contiene un listado de los programas fuente del Sistema Tutorial **ATOMO**.

2. REVISION DE LA LITERATURA.

2.1 Flujos de Información y Control en Sistemas Educcionales.

Una de las características generales a un sistema es la existencia de dos tipos de flujo: información y control. Bajo este principio, algunos desarrollos teórico conceptuales plantean que el proceso de enseñanza-aprendizaje es un *sistema* en el que se establecen flujos de información y control entre sus dos elementos constituyentes: el educador y el educando (Guevara-Pozas, 1992).

Los flujos de información y control se pueden detectar en los diferentes sistemas educacionales.

Sistema Libro-Lector.

Uno de los sistemas educativos más frecuentes se establece entre un libro impreso y su lector. La información está completamente contenida en el libro y fluye hacia el lector. Pero el control es ejercido totalmente por el lector. De esta manera el educador no puede evitar que el lector visite una lección avanzada aunque no haya adquirido el conocimiento previo necesario. Es imposible establecer algún grado de creencia sobre la transposición de conocimiento desde el educador al educando. La información está encriptada en forma de texto y se utilizan algunas gráficas para ilustrar el comportamiento de la parte del universo que se relata.

Sistema Cine-Espectador.

Un objeto genera una figura⁽¹⁾ cuando refleja un cierto conjunto de fotones que recibe desde una fuente luminosa. Los fotones reflejados re-forman la figura del objeto en el sistema nervioso de un sujeto y producen una imagen congruente, o no, con su acervo de conocimiento. Esta propiedad reflexiva de los objetos tiene, por sí misma, un valor educativo; debido a que un objeto determinado puede utilizarse

(1) Guevara-Pozas distingue los conceptos OBJETO, como la entidad que existe en el mundo real y, FIGURA como la abstracción que hace el humano del objeto.

para ejemplificar las características de todos los objetos de su categoría. Y la información luminosa es, mas o menos, compartida por un auditorio.

En algún momento se descubrió que a la imagen de un objeto se podía fijar sobre una película fotosensible y reproducirse sobre papel. Este avance resultó trascendente para la educación en tanto que permitió el análisis de objetos que no necesariamente forman parte en el entorno del sujeto. Y el conocimiento, al menos morfológico, se hizo accesible a un mayor número de sujetos.

Cuando el cine logra simular imágenes en movimiento se abre una nueva posibilidad para la educación. Los fenómenos pueden analizarse en su transcurso temporal, o al menos secuencial. Y establece un nuevo sistema educativo: cine-espectador.

En este sistema el flujo de información se establece desde la proyección hacia el educando. Pero también el control sobre la velocidad y secuencia de exposición del material educativo. El universo se explica a través de modelos icónicos en imágenes tridimensionales que se representan en un plano. El educando está obligado a mantener una actitud receptiva y no hay forma de determinar, desde el interior del mismo sistema, si el conocimiento ha pasado al educando o no. Si esto no ha ocurrido, el educando se ve expuesto a un material que no puede incorporar a su acervo cognitivo mas que en relación a la morfología de los eventos que representa.

Sistema Profesor-Clase.

El ejercicio educativo ha estado caracterizado, siempre, por la interacción entre un profesor y su grupo de alumnos o clase. En este sistema el flujo de información y control también se establece desde el educador hacia el educando, que en este caso es múltiple. Sin embargo, hay un cierto flujo de información desde el educando hacia el educador. Por ejemplo: cuando responde a un cuestionamiento formulado por el educador o cuando informa que no ha entendido alguna parte de la

lección. En este mismo ejercicio, genera una señal de control que modula la acción del educador en cuanto a la velocidad de exposición del material educativo. Pero, finalmente, la decisión sobre la continuidad del curso, el número de repeticiones sobre un material específico, y el juicio sobre el nivel de aprovechamiento por parte de la clase, descansa sobre el educador.

Sistema Computadora-Usuario.

El desarrollo de la física del estado sólido, concretizado en la tecnología computacional, establece la plataforma material para la aparición de un nuevo y privilegiado sistema educativo: la computadora y el usuario.

Consideramos privilegiado a este sistema porque es el único que permite un flujo de información y control, casi, recíproco entre el educador y el educando. El educando decide el tiempo de exploración de una pantalla antes de permitir que la computadora exponga la siguiente o le inste a continuar cuando un contenido le resulta conocido. La computadora explora continuamente el nivel de aprovechamiento del educando y registra si ocurre, o no, la adquisición de información. Cuando el estudiante no demuestra haber adquirido el mínimo necesario, la computadora no le permite avanzar sobre el resto del material.

La información tiene una representación eminentemente gráfica y la animación hace más sensible el comportamiento de los procesos. El educando tiene un control total sobre el número de repeticiones del material y sobre el sitio y hora adecuados para instruirse. Esto independiza al proceso instruccional del aula y del mentor.

En este nuevo sistema, formalmente llamado **Instrucción Asistida por Computadora (IAC)**, el instructor está representado por una computadora alimentada con *Software ex profeso*.

El *Software Educativo (SE)* es un conjunto de programas para computadora diseñados por el Ingeniero de *Software Educativo (ISE)*, que tiene el objeto de aplicar una estrategia de enseñanza, así como de establecer la interfase educando-

computadora. El ISE es una *entidad conceptual* donde participan expertos en la materia a exponer, en diseño gráfico, en pedagogía y en Informática (Guevara-Pozas, 1992). De esta manera, el SE es el producto final del trabajo multidisciplinario y constituye el elemento clave en la Instrucción Asistida por Computadora.

Existen diferentes tipos de Software Educativo, todos ellos siguen dos funciones instructivas básicas: a) servir como medio primario de transmisión de conocimiento; y b) reforzar un aprendizaje efectuado por otros medios.

En el apartado siguiente describimos los diferentes tipos de Software Educativo más reconocidos.

2.2 TIPOS DE SOFTWARE EDUCATIVO

Denis (en Hong, 1987) y Price (1991), coinciden en identificar 5 tipos de software educativo :

- 1) Sistemas Tutoriales.
- 2) Simuladores.
- 3) Sistemas de Resolución de Problemas.
- 4) Juegos Instruccionales.
- 5) Ejercitadores.

Adicionalmente, nuestro equipo de trabajo reconoce el Leccionario como un sexto tipo que se añade a esta clasificación.

A continuación se detallan las características de cada tipo de Software Educativo :

Sistemas Tutoriales.

Un sistema tutorial realiza las funciones de un tutor. Alessi y Trollip (en Smith, 1988) definen al Sistema Tutorial como "el programa de computadora que enseña a

través de un diálogo con el estudiante, presentándole información y haciéndole preguntas”.

Los Sistemas Tutoriales presentan la característica de individualizar la exposición del material para cada estudiante y llevar un registro del progreso de cada uno de ellos.

Simuladores.

Una simulación es una analogía de una situación real, con la cual el estudiante interactúa a través del manejo de las diferentes variables que determinan el estado de la simulación, con el propósito de identificar la relación existente entre éstas. Se presenta al estudiante un posible escenario con el objeto de que tome decisiones basadas en los datos proporcionados, y ejecute acciones precisas.

El ambiente de simulación se modifica acorde a las acciones tomadas por el alumno. La simulación continúa hasta que se llega al resultado deseado, o cuando las condiciones introducidas por el alumno son completamente erróneas, al grado de que no sea posible seguir con la simulación (*v.gr.*, se murió el paciente, explotó la fábrica, se cayó el avión, etc.)

Sistemas de Resolución de Problemas.

En este tipo de sistemas el estudiante utiliza las capacidades de cálculo de la computadora para resolver los problemas que se le presentan. En algunos modelos de resolución de problemas se instrumenta a la computadora como una calculadora “inteligente” que plantea problemas al estudiante, y lo retroalimenta en forma inmediata según éste vaya planteando la solución. En otros modelos, el estudiante propone solamente un método de solución y la computadora lleva a cabo los pasos.

Juegos Instruccionales.

El objetivo de este tipo de sistemas es reforzar el aprendizaje por medio de un juego. Los juegos instruccionales, como cualquier juego, tienen reglas y establecen la competencia entre sus jugadores; siempre resulta un ganador y un

perdedor. Están diseñados para entretener al alumno utilizando las capacidades de color, sonido y gráficos de la computadora. Riedsel y Clements (en Smith, 1988) escribieron: "Un buen juego instruccional requiere de la actividad física y mental de los jugadores. Involucra la oportunidad de que el niño con mayor cantidad de conocimientos y/o habilidad gane el juego más a menudo que uno que no cuente con estos atributos. La competencia que se crea en el juego, ya sea por la competencia individual entre jugadores, contra el tiempo, o contra la computadora, hace que la atención por parte del estudiante se mantenga constante en un nivel alto."

Ejercitadores.

El objetivo principal de los Ejercitadores es reforzar los conocimientos aprendidos por otros medios. Se presenta al estudiante un reactivo y se le proporciona algún tipo de reforzamiento inmediato.

Comunmente, este tipo de sistemas incluyen la presentación semi-aleatoria de tareas que requieren por parte del estudiante acciones muy simples pero precisas. Se utilizan con mayor frecuencia con materias tales como ortografía o aritmética elemental, por requerir éstas una ejercitación intensa (Burke, 1986).

Leccionarios.

Los sistemas de este tipo estructuran la información de manera lineal, y mantienen un bajo nivel de interacción con el educando⁽¹⁾. Debido a esto, el Leccionario está considerado dentro de la categoría de Software Educativo *No Interactivo* en contraste con el Software Educativo *Interactivo* al que pertenecen el resto de los tipos de sistemas presentados en este apartado. Su propósito fundamental es de exposición y pueden ser utilizados como material de apoyo a una clase o al contenido de un libro. (Campos, *et al* 1991).

(1) Price (1991) menciona otro tipo de Software Educativo llamado Page Turner ("Pasador de hojas") cuyas características coinciden con las del leccionario.

Hong (1987) propone clasificar estos tipos de Software Educativo de acuerdo al tipo de conocimiento que transmiten:

1. Conocimiento declarativo. El SE proporciona información acerca de una materia o tema específico. El contenido del SE es descriptivo y se espera que el estudiante aprenda las características del sistema objeto de conocimiento que se le presenta, y los principios que regulan la interacción entre sus elementos.
2. Conocimiento procedural. Se busca que el estudiante conozca los mecanismos y alternativas de interacción entre él y los sistemas en su ambiente.

En otras palabras, el conocimiento declarativo se enfoca más al *qué es*, mientras que el conocimiento procedural se enfoca hacia el *cómo interactuar* con el medio ambiente.

El SE que se genera hacia estos tipos de conocimiento puede presentar dos posibles estructuras:

a) *Vertical*, agrupa a los materiales educativos que presentan una secuencia ligada. El proceso instructivo se realiza concepto por concepto. Por ejemplo, si se desea enseñar la multiplicación, se tuvo que haber enseñado primero la suma. No es posible bajo esta estructura cambiar el orden conceptual.

b) *Horizontal*, bajo esta estructura, los materiales educativos se presentan en forma aleatoria, sin importar el orden conceptual. Por ejemplo, si se desea enseñar ortografía, da lo mismo se enseñe primero la regla ortográfica "m" antes de "p", que la transformación de la "z" en "c" en los plurales.

La clasificación que Hong proporciona que se ilustra en la figura 1.

De acuerdo con la figura 1, presentamos el caso de un Sistema Tutorial: el tipo de conocimiento que transmite este tipo de sistema es declarativo, sin ser esto definitivo, ya que puede ser también procedural; en el caso de su estructura, ésta es exclusivamente vertical.

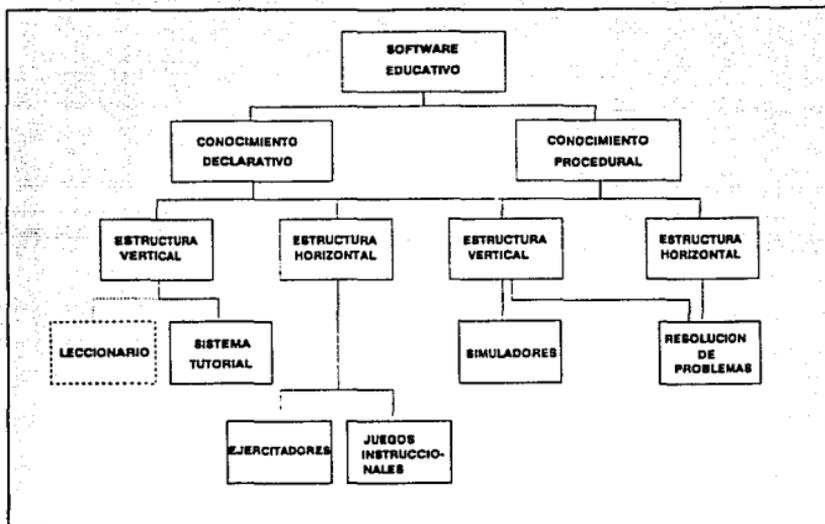


Figura 1. Clasificación de los tipos de Software Educativo según Hong (1987). Se presenta una estructura arbórea que relaciona a los diferentes tipos de SE con las características de conocimiento y estructura que poseen. Con línea punteada se señala el lugar que ocuparía, dentro de esta clasificación, el Leccionario (no contemplado por el autor).

La situación para un Sistema de Resolución de Problemas es distinta, ya que el tipo de conocimiento que transmite puede ser tanto declarativo como procedural y puede seguir cualquiera de las dos estructuras (horizontal y vertical).

Existe una gran variedad de herramientas computacionales para la construcción de cualquiera de los tipos de SE antes descritos; en el apartado siguiente discurremos en torno a ellas.

2.3 HERRAMIENTAS PARA LA CONSTRUCCION DE SOFTWARE EDUCATIVO

Price (1991) y Hudson (1986) identifican las siguientes herramientas dentro de la Ingeniería de Software Educativo:

- a) Lenguajes de propósito general.
- b) Lenguajes de Autor.
- c) Sistemas de Autor.

Las características principales de cada una de éstas herramientas son:

Lenguajes de Propósito General.

Este tipo de lenguajes se componen de un conjunto de instrucciones que se escriben con palabras comunes (v.gr., LEE, ESCRIBE, MUEVE) y que representan las acciones a llevar a cabo. Los programas escritos en estos lenguajes son independientes de la computadora en la que se ejecutan ya que se traducen a lenguaje de máquina; asimismo, proveen estructuras de control (secuencia, repetición y selección) y una amplia variedad de tipos de datos, lo que posibilita la construcción de aplicaciones de cualquier índole (v.gr., numéricas, gráficas, de manejo de archivos, etc.). Otra característica de estos lenguajes es que permiten realizar interfases con programas escritos en lenguaje ensamblador y ejecutar interrupciones de bajo nivel del Sistema Operativo, lo que amplía su potencia en la programación de computadoras.

Algunos de estos lenguajes son: a) BASIC (*Beginners' All-purpose System Information Code*), en sus versiones iniciales se empleó como lenguaje anfitrión de las primeras computadoras personales como la APPLE I, II, IIe, y la Commodore 16. BASIC fue un lenguaje ampliamente utilizado para el desarrollo de SE debido a que: incluía instrucciones *ex profeso* para el manejo de gráficos y animación; funcionaba a través de un intérprete y; a que no requería de una estructura formal de programación. b) Pascal y C son lenguajes cuyas versiones actuales (Turbo Pascal y Turbo C de Borland Inc., Quick C de Microsoft, etc.) proveen un gran número de bibliotecas de funciones predefinidas que facilitan el manejo de ambientes gráficos y de base de datos, convirtiéndolos en una opción poderosa para el desarrollo de SE.

Dado que la programación de SE requiere un manejo intensivo y extensivo de gráficas y animación, utilizar sólo un lenguaje de propósito general puede resultar complejo para personas no familiarizadas con su uso, ya que en algunos casos, se requiere una gran cantidad de código para implementar una sola pantalla; es aquí que surgen como alternativa los *Lenguajes de Autor*.

Lenguajes de Autor.

Son lenguajes de programación creados especialmente para personas que desean desarrollar SE sin tener que profundizar en técnicas de programación y manejo de tipos de datos propios de los lenguajes de propósito general. Estos lenguajes tienen como objetivo facilitar el desarrollo de SE.

La característica principal de estos lenguajes es que cuentan con un editor gráfico y con funciones de animación (*v.gr.*, MUEVE-ARRIBA, MUEVE-IZQUIERDA, GIRA, etc.), que facilitan la creación de gráficos y secuencias animadas, disminuyendo notablemente el tiempo de desarrollo. Sin embargo, se ha encontrado que la eficiencia de los lenguajes de autor es limitada, ya que no tienen todas las instrucciones y funciones de un lenguaje de propósito general, por lo que el autor debe amoldarse a las facilidades que provee este tipo de lenguajes.

Uno de los lenguajes de autor más populares es PILOT, desarrollado por APPLE Computer en 1982. Otra versión para computadoras IBM PC y compatibles es PC/Pilot desarrollado por Sperry Rand Co. (1983). Entre las características más notables de ésta versión se encuentran:

- Puede operar en un ambiente IBM PC.
- La animación de gráficos y textos se realiza por medio de funciones ya definidas, lo que hace de la animación una tarea realmente sencilla.
- Tiene interfase con otros productos gráficos como Story Board y Paint Brush, así como con Lotus 123.
- Habilita el uso del ratón para mejorar la interfase alumno computadora.
- Incluye un editor gráfico.

- Permite el manejo de archivos para registrar el avance de los alumnos.
- Cuenta con la utilidad "Capture", para tomar imágenes de pantallas de otros sistemas.

Se han desarrollado varios lenguajes de autor; Barker (en Price, 1991) lista aproximadamente 20, entre ellos incluye Writer III, Unison y Engol.

El nivel de refinamiento alcanzado en la construcción de herramientas para el desarrollo de SE ha permitido crear sistemas que funcionen como "Shells" (Caparazones o Estructuras vacías), los cuales permiten generar aplicaciones en forma automática eliminando así la necesidad de programación. A estos "Shells" se les conoce como *Sistemas de Autor*.

Sistemas de autor.

Esta herramienta funciona a través de una serie de "Menús" que presentan al usuario las opciones disponibles para generar el SE. Como ejemplo de estas funciones tenemos:

- "Crea Pantalla", lleva al usuario a un editor de gráficos
- "Crea Reactivo", registra una pregunta, su respuesta correcta, los posibles mensajes en caso de error y la pantalla a la que ésta va asociada;
- "Crea Lección", crea una lista de pantallas y reactivos y los agrupa en una unidad instruccional;
- "Crea Música", permite añadir música o sonido a una pantalla de reforzamiento.

El usuario va construyendo el SE sin escribir una sola línea de código, ya que una vez terminadas las especificaciones del autor, se genera el código ejecutable en forma automática.

Al igual que los Lenguajes de Autor, los Sistemas de Autor ahorran considerablemente el tiempo necesario para desarrollar SE; no obstante, este desarrollo queda restringido a las opciones disponibles de cada sistema de autor.

Scenario, producido por Teachbyte en 1984, es un ejemplo de este tipo de herramientas.

Barker (en Price, 1991) menciona más de 80 sistemas de autor; algunos ejemplos son ACCES para la Apple IIe, SAM IV para IBM PC y compatibles, COURSE OF ACTION para la Apple Macintosh.

En la práctica se ha demostrado que al utilizar sólo una herramienta de desarrollo se desaprovecha las facilidades que presentan cualquiera de las otras. En nuestro grupo proponemos la realización de interfases entre varias de estas herramientas, de manera que el Ingeniero de Software Educativo tome ventaja de cada una de ellas. Un ejemplo concreto de estas interfases se discute en capítulos posteriores.

2.4 METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS TUTORIALES

Independientemente de la herramienta de construcción que se utilice, es necesario seguir una serie de pasos para la elaboración de SE. Hudson (1986) propone la siguiente metodología para el desarrollo de Sistemas Tutoriales (ST).

Las fases que componen esta metodología son:

- I. Análisis.
 - Determinación del alcance del sistema.
 - Justificación del sistema.
- II. Representación del contenido de las lecciones del sistema.
 - Creación de la lista de temas.
 - Creación de las tareas de aprendizaje.
 - Análisis de las tareas de aprendizaje.
- III. Diseño.
- IV. Construcción del sistema.
 - Escritura de pantallas.
 - Escritura del guión.
 - Escritura de las directrices.
 - Codificación.
- V. Validación.
 - Prueba Piloto.
 - Aplicación de pruebas previas y posteriores.

VI. Documentación.

VII. Mantenimiento.

Describimos a continuación cada una de las fases que conforman esta metodología.

I. Análisis

Las actividades que se realizan dentro de esta fase son :

Determinación del alcance del sistema.

Se refiere al nivel de conocimientos que se desea adquiera el educando después de haber estudiado con el ST.

En esta etapa se determinan los objetivos instruccionales del ST, así como el grado de dificultad del mismo.

Justificación del sistema.

Se discuten los motivos originales del proyecto, sus expectativas y la importancia de llevarlo a cabo.

II. Representación del contenido de las lecciones del sistema.

Esta fase consta de las siguientes actividades:

Creación de la lista de temas.

Consiste en escribir en una tarjeta cada uno de los temas que conformarán las lecciones del ST. Posteriormente, las tarjetas se ordenan y se genera una lista definitiva de los temas.

La redacción del tema en la tarjeta debe breve se recomienda evitar el uso de adjetivos y calificativos para facilitar su manejo.

El sistema de tarjetas proporciona "movilidad" a los temas, de tal forma que resulta más sencillo manipular tarjetas que escribir nuevas listas cada vez que se plantee un cambio.

Creación de las tareas de aprendizaje.

Una tarea de aprendizaje es una acción que se desea el alumno realice; se expresa con una frase breve en infinitivo, por ejemplo: resolver ecuaciones, crear entradas, definir conceptos, etc.

La relación de Tareas de Aprendizaje puede crearse a partir de la lista de temas, aunque no debe asumirse que exista una correspondencia biunívoca. Un tema puede corresponder a varias tareas de aprendizaje y viceversa.

Análisis de las tareas de aprendizaje.

Los criterios que comprende el análisis de las tareas de aprendizaje son : frecuencia, importancia, dificultad de ejecución, nivel de conocimiento previo y dificultad de retención. Este análisis es necesario para definir los ejemplos y reactivos que deben utilizarse, cuándo presentarlos y el número de repeticiones necesarias para su retención.

III. Diseño.

El diseño del sistema tutorial se determina por la trayectoria que sigue el material durante su exposición, a partir de su interacción con el estudiante.

Hudson (1986) señala los siguientes diseños :

Diseño Lineal.

Un ST desarrollado con un diseño lineal es aquel que sigue una trayectoria única. Es decir, a cada estudiante se le presenta el mismo material independientemente de las respuestas que proporcione. Por tratarse de una trayectoria única, este diseño resulta económico y fácil de validar y revisar.

Diseño de Bifurcación.

Este diseño se basa en trayectorias alternativas en la exposición del material. Dependiendo de sus respuestas, el estudiante puede ver un material distinto al que vea otro estudiante, a pesar de haber seguido la misma trayectoria inicial. La bifurcación tiene lugar generalmente a continuación de un reactivo; todos los

estudiantes que respondan acertadamente seguirán por la misma rama, mientras que aquellos que lo hagan de manera errónea, seguirán una rama alternativa.

El atractivo de los diseños de bifurcación reside en que permiten una instrucción personalizada; en consecuencia, cada estudiante puede moverse a su propia velocidad a través del sistema.

Una característica de este diseño es que se deben construir n lecciones alternativas por cada reactivo que se presente, lo que puede alargar el tiempo de desarrollo.

Diseño de Múltiples Niveles.

Este diseño tiene material escrito para diferentes niveles de comprensión o habilidad, a fin de que la selección del material a exponer sea acorde con el nivel de conocimientos del educando. El nivel más alto es el más complejo, el material contiene menos explicaciones y ejemplos, y un menor número de preguntas, pero de mayor grado de dificultad.

Los niveles más bajos están escritos con explicaciones a mayor detalle, con más indicaciones y ayudas. La lógica del sistema para el diseño de múltiples niveles debe incluir una estrategia de evaluación para "pasar" al estudiante a otro nivel o dejarlo en el mismo, dependiendo de su desempeño.

Diseño Regenerativo.

En este tipo de diseño, cada vez que el sistema presenta un reactivo o ejemplo, lo selecciona aleatoriamente de un conjunto de posibles alternativas; de esta forma, el material de exposición para un mismo tema es distinto en cada ocasión que el estudiante pasa por el mismo lugar.

El tipo de diseño a seguir depende de múltiples criterios. Entre los más importantes están: a) Los objetivos instruccionales y; b) los recursos materiales, técnicos y de tiempo disponibles.

IV. Construcción del sistema.

La construcción del Sistema Tutorial consiste en la integración de imágenes, directrices y estrategia instruccional, en un conjunto de programas y datos utilizables por la computadora.

Las actividades de esta fase son:

Escritura de pantallas.

Comprende la selección y ubicación de texto y gráficos en una pantalla. Durante esta actividad la escritura se realiza sobre papel para facilitar las posibles modificaciones.

Existen dos formatos básicos de pantallas: de *Criterio* y de *Enseñanza*.

Las pantallas de Criterio tienen como objetivo medir el aprendizaje del alumno a través de uno o más reactivos. Se caracterizan por la ausencia de elementos que sugieran la respuesta correcta al alumno. La escritura de las pantallas de criterio está relacionada directamente con las tareas de aprendizaje definidas anteriormente: a cada una de estas tareas debe corresponder por lo menos una pantalla de criterio que proporcione datos acerca del cumplimiento de éstas.

Las pantallas de Enseñanza, por su parte, tienen como objetivo proporcionar al alumno la información necesaria para que sea capaz de cumplir con una tarea de aprendizaje. Las pantallas de enseñanza preceden siempre a las pantallas de criterio.

Escritura del guión.

El guión es una lista que indica la secuencia de presentación que siguen las pantallas de enseñanza y las pantallas de criterio. Es recomendable escribirlo sobre papel de manera tal que sea fácil reordenarlo y añadir y quitar secuencias antes de iniciar la codificación.

Escritura de las directrices.

Las directrices son un conjunto de indicaciones acerca del manejo del sistema tutorial que deben presentarse al estudiante antes de iniciar la sesión y durante la

misma. Ejemplos de algunas directrices son : [ENTER] = Continuar, [F1] = Ayuda, [ESC] = Salir, etc.

Codificación.

Consiste en la transcripción de las pantallas (en su formato en papel), de las directrices y del guión, a un código ejecutable por la computadora.

La codificación puede realizarse utilizando una o varias herramientas de construcción. Los pasos asociados a esta actividad dependen de las herramientas elegidas; por ejemplo, si se utiliza un lenguaje de propósito general, es necesario crear los programas y compilarlos; si se utiliza un sistema de autor, se introduce el guión de acuerdo con las instrucciones y opciones disponibles del menú.

V. Validación.

Esta fase consiste en probar en forma empírica el sistema, a través de la aplicación de pruebas con un grupo de estudiantes (que sea representativo del conjunto total de estudiantes que utilizarán el programa). Este proceso se repite hasta que el sistema alcance un grado aceptable de ejecución, mismo que define el Ingeniero de Software Educativo.

Las actividades que comprenden esta fase son :

Pruebas Piloto.

Cuando se ha completado el sistema con entera satisfacción por parte del autor, se necesita un estudiante voluntario que lo pruebe cuando todavía está en manuscrito. Una forma conveniente de hacerlo es escribir cada pantalla en una hoja por separado. El refuerzo se escribe en la parte superior de la hoja, y ésta se dobla para que resulte invisible al estudiante. Asimismo se indica al estudiante que, tan pronto escriba su respuesta, descubra el refuerzo.

Es aconsejable sentarse junto al estudiante, sin explicarle nada mientras trabaja. Habrá algunas pantallas que el estudiante no sea capaz de contestar; si este solicita la respuesta, el ISE debe escribir algunas pantallas adicionales y

mostrárselas en vez de darle la respuesta. Si se le proporciona al estudiante una explicación verbal, se corre el riesgo de invalidar una secuencia de pantallas. Cuando el estudiante conteste en forma correcta, es recomendable sonreír o decir alguna frase animosa (v.gr. ¡Bien hecho!).

Una vez realizada la prueba piloto en manuscrito, es necesario realizar otra prueba piloto, pero ahora con el sistema ya funcionando en la computadora, para lo cual se recomienda utilizar como herramienta de validación el registro de aciertos y errores de los estudiantes.

Aplicación de pruebas previas y posteriores.

Las pruebas previas sirven para determinar si los estudiantes tienen los conocimientos necesarios para tomar la lección.

Las pruebas posteriores son similares a las pruebas previas, a diferencia que su objetivo es verificar la transmisión de conocimiento específico. El éxito del sistema radica en el resultado positivo de la diferencia entre las pruebas previas y las posteriores.

El número de ciclos de validación está determinado por el número de correcciones y ajustes que necesite el sistema y de la cantidad de tiempo disponible.

Cuando el sistema haya pasado por una serie de ciclos de validación y funcione de manera adecuada, se puede considerar terminado y listo para utilizarse.

VI. Documentación.

Con este término se define al material descriptivo que acompaña al programa y explica su propósito y su utilización.

La documentación consiste en escritos impresos en papel, y/o comentarios y observaciones que se incluyan en los programas y que aparezcan en pantalla. Una buena documentación incorporada en las lecciones, es una ayuda inmensa aún cuando casi nunca elimina la necesidad de documentación en papel. Asimismo debe

describir claramente el ambiente en el que se ejecutará el sistema, es decir, el modelo de la computadora, discos ó cinta, el tamaño de la memoria, etc.

La documentación debe incluir las instrucciones necesarias para la ejecución del sistema; estar al nivel de la persona que menos conocimiento pueda tener y; emplear un lenguaje sencillo y completo.

VII. Mantenimiento.

En esta fase se realizan las mejoras, adaptaciones o ajustes que la fase de validación señale, así como aquellos cambios necesarios para que la información que se presenta en el Sistema Tutorial no se vuelva obsoleta.

Esta es una metodología detallada. Su seguimiento, paso a paso, supone un desarrollo exitoso de Sistemas Tutoriales. Sin embargo, Hudson no reporta resultados de la aplicación de su metodología en un sistema concreto. En nuestro caso, utilizamos las fases de análisis, de representación del contenido de las lecciones del sistema en su punto de creación de la lista de temas y, la fase de construcción del sistema.

2.5 CONTROL EN LOS SISTEMAS DE INSTRUCCION POR COMPUTADORA.

La función de control en los sistemas de instrucción por computadora consiste en determinar las acciones a ejecutar por el sistema en base a su interacción con el educando. Para implementar esta función, existen actualmente dos tendencias:

- a) El uso de programación procedural que emplea las estructuras de control: secuencia, selección y repetición (Lawrence, 1987) y;
- b) el uso de técnicas de Inteligencia Artificial (IA). Sleeman *et al* (1982) y Rich (1988), mencionan la aplicación de técnicas de IA para el control de sistemas de instrucción asistida por computadora⁽¹⁾ tales como : 1) Redes semánticas;

(1) A la utilización de técnicas de Inteligencia Artificial en la Instrucción Asistida por Computadora, se ha llamado Instrucción Inteligente Asistida por Computadora (IIAC).

2) Técnicas de inducción y deducción de propósito específico y; 3) Reglas de producción; entre otras.

Algunos ejemplos son :

- SOPHIE (Brown *et al*, en Sleeman, 1982) Sistema de resolución de problemas en electrónica.
- EXCHECK (Smith *et al*, en Sleeman, 1982): Resolución de problemas matemáticos.
- MACSYMA (Genesereth, en Sleeman, 1982). Es un sistema para manipulación algebraica.

La principal desventaja de la utilización de técnicas de IA radica en la necesidad de involucrar a personal altamente calificado en aspectos de IA para la creación y mantenimiento de sistemas IIAC. Y por otra parte, los lenguajes para procesamiento simbólico carecen de estructuras para representación gráfica y animación.

Aún cuando la programación procedural para el control en los sistemas IAC, particularmente en los sistemas tutoriales, ha sido la más ampliamente utilizada (Price, 1991), esta programación presenta una desventaja: cuando un sistema tutorial es considerablemente grande (o se le van añadir lecciones), su estructura se vuelve compleja, al igual que las funciones requeridas para su manejo (complejidad funcional); lo que se traduce en un arduo trabajo de re-ingeniería de software que implica analizar el código existente para determinar si los cambios alteran la estructura del sistema, insertar modificaciones, realizar nuevas pruebas, etc.

2.6 AUTOMATAS DE ESTADOS FINITOS.

Consideramos que un modelo matemático que permite que una estructura funcional permanezca constante e independiente de la complejidad estructural de un sistema, es el **Autómata de Estados Finitos**.

Un autómata de estados finitos es un modelo matemático cuya definición formal es la siguiente:

$$M = [S, I, O, F_s, F_o]$$

es un Autómata de Estados Finitos, donde:

S, es un conjunto finito de Estados;

I, es un conjunto finito de símbolos de entrada;

O, es un conjunto finito de símbolos de salida;

F_s es la función de estado siguiente :

$$\text{Estado}[t_{i+1}] = F_s(\text{estado}(t_i), \text{entrada}(t_i));$$

donde t es tiempo.

F_o es la función de salida :

$$\text{Salida}(t_i) = f_o(\text{estado}(t_i));$$

De esta manera, para un estado (S_n), en el que se encuentra el autómata en un tiempo t_i, el estado en el que se encontrará el autómata en un tiempo t_{i+1} está determinado por S_n y la entrada al autómata en el tiempo (t_i)⁽¹⁾. El autómata reconocerá como entradas válidas, sólo aquellas que pertenezcan al conjunto I. La función F_o determinará si existe una salida para el estado presente. El autómata debe empezar en un estado inicial (S₀).

Por ejemplo, considérese un autómata para el análisis de cadenas de ceros y unos, de manera tal que sea posible determinar si la cadena de entrada cumple con las siguientes condiciones :

(1) Dado que el estado presente depende del estado y la entrada anteriores, es posible retroceder hasta el estado inicial. Así, el estado en el que se encuentra el autómata en cualquier momento sirve como una *memoria* de las entradas pasadas.

- 1) contenga un número impar de unos y;
- 2) contenga por lo menos un cero.

El autómata que resuelve este problema es el siguiente :

$$S = \{ S_0, S_1, S_2, S_3 \}$$

$$I = \{ 0, 1 \}$$

$$F_0 = \{ \phi \}$$

$$F_s = \text{tabla 1}$$

Estado	Entrada	
	0	1
S_0	S_2	S_1
S_1	S_3	S_0
S_2	S_2	S_3
S_3	S_3	S_2

Tabla 1.- Función F_s (del estado-siguiente) en el Autómata de Estados Finitos del ejemplo. Dado un estado actual, el autómata pasa a un estado siguiente de acuerdo a la entrada en el tiempo presente.

En la figura 2 se muestra el gráfico del autómata de estados finitos que resuelve el problema del ejemplo.

Si el autómata llega al estado S_3 , indica que la cadena de entrada cumple con las condiciones establecidas.

No importa que tan grande sea el número de estados del autómata, el estado siguiente siempre estará determinado por el estado y entrada actuales. Este principio funcional permanece constante.

El autómata de estados finitos es útil en el estudio de los lenguajes formales, y a menudo se encuentra en compiladores e intérpretes de varios lenguajes de programación.

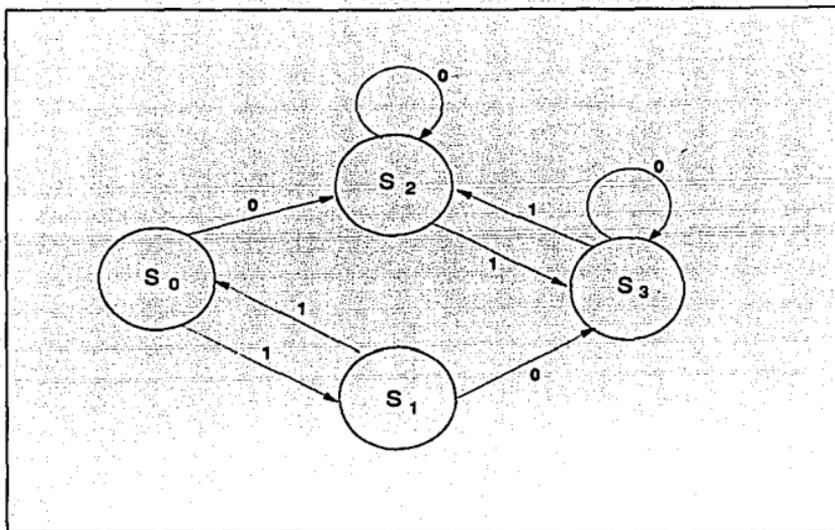


Figura 2. Gráfico del Automata de Estados Finitos que reconoce palabras con un número impar de unos y al menos un cero. Los círculos representan los estados y las líneas dirigidas la función de estado-siguiente; los números sobre las líneas representan las entradas.

3. MATERIAL Y METODOS.

Para el desarrollo de este trabajo, utilizamos una computadora ACER XT de 640 KB RAM, disco duro de 20 MB, manejador de disco de 5 ¼ pulgadas y monitor CGA.

El sistema tutorial ATOMO se desarrolló en una computadora modelo XT porque ésta es la plataforma más simple de que se dispone lo que permite ejecutar el sistema en cualquier PC compatible XT o mayor.

Otros periféricos utilizados son: un mouse logitech que facilitó la construcción de pantallas y secuencias animadas; una impresora Delta 15-180 y; una impresora HP LaserJet II.

Durante la fase de implementación física, se utilizó el paquete Story Board Plus (marca registrada por IBM, versión 1987) para la elaboración de imágenes y secuencias gráficas. Los lenguajes PC/Pilot (marca registrada por SPERRY Co., 1986) para la elaboración de imágenes y como evaluador de las respuestas; Turbo Pascal 5.0 (marca registrada por Borland Inc. 1988).

La metodología empleada para la construcción del sistema tutorial "ATOMO" fué la propuesta por Hudson (1986) en sus fases de análisis; representación del contenido de las lecciones del sistema (en su punto de creación de la lista de temas) y; la fase de construcción del sistema.

Las funciones de control y flujos de información entre los elementos que constituyen el sistema tutorial se implementaron de acuerdo a las propuestas que conforman el cuerpo principal de la tesis.

4. RESULTADOS.

En la primera parte de nuestros resultados reportamos un Esquema General sobre el Flujo y Transformación de la Información para la Construcción de SE; en la segunda parte, explicamos la estructura del Modelo para el Manejo de Flujos de Información y Control en un Sistema Tutorial; la tercera parte comprende la implementación del modelo descrito en un desarrollo concreto: el Sistema Tutorial ATOMO. Se presenta la manera de organizar el contenido temático de un sistema tutorial, así como la explicación del empleo de un Autómata de Estados Finitos como modulador de los flujos de información y control.

4.1. ESQUEMA GENERAL SOBRE EL FLUJO Y TRANSFORMACION DE LA INFORMACION PARA LA CONSTRUCCION DE SOFTWARE EDUCATIVO.

La figura 3. describe el esquema que nuestro grupo de trabajo propone para la construcción de software educativo.

El ISE revisa la información existente en torno al conocimiento formal (métodos educativos y teorías sobre el Aprendizaje) y a la Ingeniería de Software Educativo (tipos de diseño y metodologías), incorpora su heurística y genera una Base de Construcción de SE (proceso 1, figura 3).

La Base de Construcción, junto con los contenidos temáticos y las herramientas de construcción que el ISE elija, son los elementos necesarios para la Construcción del SE (proceso 2, figura 3).

Una vez concluido el SE, se aplica de manera experimental (proceso 3, figura 3). De esta aplicación se elabora un Reporte de Resultados.

Este reporte proporciona los elementos para el Planteamiento Formal de Hipótesis Específicas (proceso 4, figura 3) bajo las cuales se realiza un nuevo experimento. Los procesos 3 y 4 se repiten hasta obtener evidencia suficiente sobre la validez científica de las hipótesis.

Finalmente, el conocimiento generado modifica la Base de Construcción (proceso 5, figura 3); esta modificación puede consistir en la generación de nuevo conocimiento, de nuevas teorías educativas, y/o en la corroboración de lo existente.

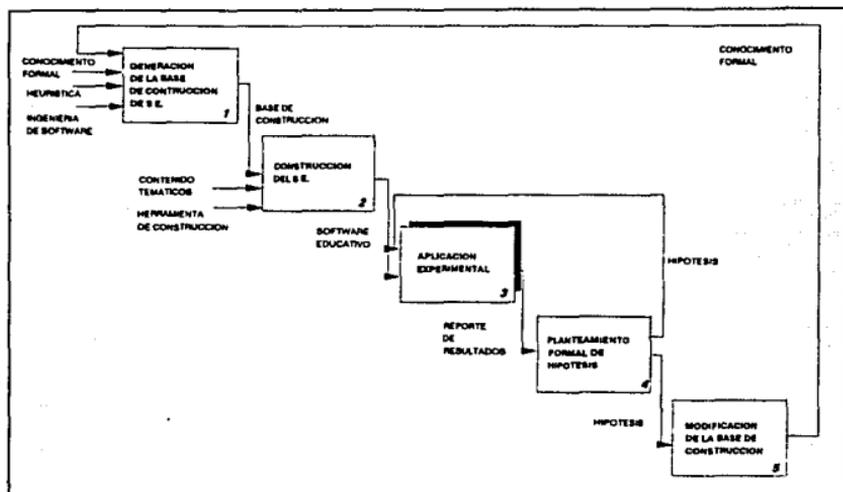


Figura 3. Esquema General sobre el Flujo de y Transformación de la Información para la Construcción de Software Educativo. (Descripción en el texto).

El proceso de aplicación experimental involucra a la interacción alumno-computadora y además presupone la participación más intensa de la máquina. Por tanto, presentamos el refinamiento de este proceso para describir los flujos de información que se establecen (figura 4).

El proceso de Selección de Material (proceso 3.1, figura 4) se encarga de determinar el conjunto de pantallas que se presentarán al educando. Las entradas a este proceso son: el software educativo y la calificación a la respuesta proporcionada por el educando.

El proceso de Despliegue del Material (proceso 3.2, figura 4) realiza la exposición del mismo a través de varios medios de transmisión (audio o video).

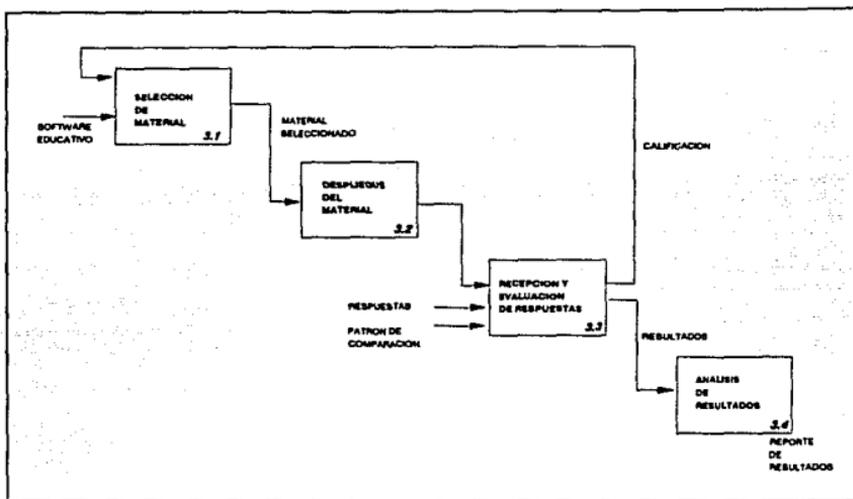


Figura 4. Refinamiento del Proceso de Aplicación Experimental en el Esquema General sobre el Flujo y Transformación de la Información para la Construcción de Software Educativo. (Descripción en el texto).

La Recepción y Evaluación de la Respuesta (proceso 3.3, figura 4) comprende la interacción del educando con el software educativo. La recepción se realiza a través de teclado, pantalla sensible al tacto, mouse, joystick, lápiz óptico, etc. La evaluación compara la respuesta recibida con un patrón interno y asigna una calificación a la misma. Este valor sirve como entrada al proceso de Selección de Material (proceso 3.1, figura 4).

El proceso de Análisis de Resultados (proceso 3.4, figura 4) reúne y verifica la información obtenida en un reporte de resultados.

El proceso de Experimentación resulta particularmente importante en nuestro país, debido a que permite delinear e incorporar las características específicas de los educandos mexicanos al desarrollo de SE.

En este esquema, existe un evidente *flujo de control* asociado al flujo de la información, que se establece entre los procesos. Dicho flujo de control determina:

- a) el siguiente material de exposición. Por ejemplo, si hay una respuesta errónea, se proporciona una explicación alternativa;
- b) el contenido del flujo de información. Por ejemplo, en algunos reactivos se presenta sólo una pista de la respuesta correcta y, tras varios intentos del estudiante, el sistema puede proporcionar la respuesta correcta y/o una explicación alterna (lo que modificará el contenido del flujo de información);
- c) la dirección del flujo de información. El software educativo presenta material al educando y éste proporciona respuestas al SE.

4.2 MODELO PARA EL MANEJO DE FLUJOS DE INFORMACION Y CONTROL EN UN SISTEMA TUTORIAL.

Hemos referido el enfoque del proceso educativo como un sistema donde se establecen flujos de información y control (Guevara-Pozas, 1992). Para el presente trabajo, los aspectos que adquieren mayor relevancia son: a) la identificación de las unidades interactuantes dentro del sistema tutorial y; b) la definición de los flujos de información y control que se establecen entre esas unidades.

El modelo que proponemos para el Manejo del Flujo de Información y Control en un Sistema Tutorial, se construye a partir de la identificación de tres mecanismos: Mecanismo de Control, Mecanismo de Evaluación y Mecanismo de Presentación (figura 5).

Mecanismo de Control.

Las funciones básicas de este mecanismo son:

1) Decidir, en base al estado actual del sistema y a la información recibida de los mecanismos de Evaluación y Presentación, la siguiente acción a realizar (v.gr. desplegar la explicación siguiente, salir del sistema, repetir explicación, aplicar reactivo).

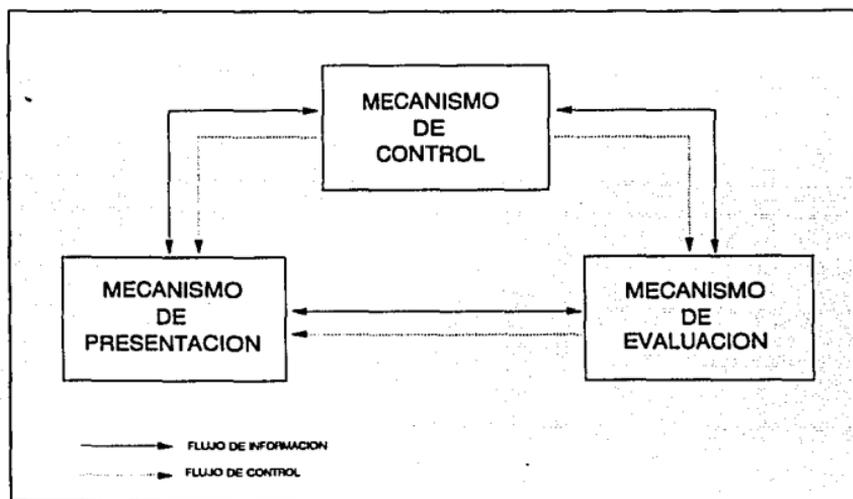


Figura 5. Diagrama con los flujos de información y control entre los tres Mecanismos que constituyen a un sistema tutorial. Un sistema tutorial incluye a los Mecanismos de Control, Presentación y Evaluación. Los flujos de información y control entre ellos se describen en el texto.

2) Establecer la coordinación entre los Mecanismos de Evaluación y de Presentación.

3) Llevar el registro de aciertos y errores del estudiante.

Otras funciones deseables en el Mecanismo de Control son:

1) Llevar la administración de usuarios (estudiantes). Esta función se recomienda cuando el sistema tutorial sea utilizado por más de un estudiante en una misma computadora.

2) Establecer el ambiente de ejecución idóneo para aprovechar al máximo las capacidades del equipo *v.gr.*, resolución gráfica, manejo de memoria y dispositivos (impresoras, graficadores, mouse, joystick).

Los flujos de Información y control que establece este mecanismo con los demás son:

Flujos de control

- Enviar orden de despliegue al Mecanismo de Presentación.
- Enviar orden de evaluación al Mecanismo de Evaluación.

Flujos de información

- Enviar nombre del módulo a desplegar al Mecanismo de Presentación.
- Enviar los parámetros necesarios (pregunta, respuesta, tipo de reactivo, número de intentos, etc.) al Mecanismo de Evaluación.

Mecanismo de Evaluación.

Las funciones básicas de este mecanismo son:

- 1) Recibir la respuesta del estudiante y almacenarla.
- 2) Interpretar la respuesta. Esta función se refiere a traducir la respuesta para poder compararla con el patrón interno enviado por el Mecanismo de Control.
- 3) Evaluar la respuesta. Consiste en comparar la respuesta contra el patrón interno, y emitir una calificación.

Los flujos de información y control que establece este mecanismo con los demás son:

Flujos de control

- Enviar orden de despliegue al Mecanismo de Presentación. Este flujo existe si y sólo si el Mecanismo de Evaluación ha recibido un flujo de control previo por parte

del Mecanismo de Control, que indique se utilizará el Mecanismo de Presentación (*vid infra*: Capítulo 5. Discusión).

Flujos de información

- Enviar resultado de la evaluación (calificación) al Mecanismo de Control.
- Enviar secuencia de imágenes a desplegar al Mecanismo de Presentación.

Mecanismo de Presentación.

Las funciones básicas de este mecanismo son:

- 1) Recuperar imágenes almacenadas en archivos gráficos.
- 2) Dirigir estas imágenes a un dispositivo de video.
- 3) Proveer los efectos especiales (movimiento, sonido, color) asociados a las imágenes.

Los flujos de información y control que establece este mecanismo con los demás son:

Flujos de control

- Recibir orden de despliegue del Mecanismo de Control.
- Recibir la orden de despliegue del Mecanismo de Evaluación

(*vid infra*: Capítulo 5. Discusión).

Flujos de Información

- Enviar aviso de terminación del despliegue al mecanismo que lo requirió.

4.3 IMPLEMENTACION DEL MODELO

El Modelo para el Manejo de Flujos de Información y Control descrito, permite identificar claramente las funciones básicas de cualquier Sistema Tutorial. Cada función puede constituir un programa (o conjunto de programas).

La ventaja de esta modularización es que es posible que cada función sea programada con diferentes herramientas de construcción de Software Educativo, lo que permite aprovechar las facilidades que cada herramienta brinda.

Hemos desarrollado el Sistema Tutorial **ATOMO**, como ejemplo concreto de la aplicación del Modelo para el Manejo de Flujos de Información y Control.

ATOMO trata sobre la historia y desarrollo de las teorías en torno a la estructura atómica. En la tabla 2 mostramos las lecciones y temas que aborda el sistema, y como ejemplos del material gráfico, las figuras 6 a 11.

SISTEMA TUTORIAL ATOMO

1. Antecedentes Históricos.

Se presentan las primeras teorías acerca de la composición de la materia y algunas de sus propiedades.

1.1 El átomo según los griegos.

Se presenta la primera noción del átomo como la partícula indivisible que conformaba toda la materia.

1.2 El átomo según Dalton.

La teoría del científico inglés John Dalton era que todos los átomos de un elemento eran iguales entre sí, pero diferentes a los de otro elemento.

1.3 Aportaciones de Tales de Mileto.

Tales de Mileto descubre que ciertos objetos, al ser frotados, eran capaces de atraer a otros objetos.

1.4 Benjamín Franklin y los tipos de electricidad.

Franklin define a los tipos de electricidad como Positiva y Negativa.

1.5 Ley General de Cargas Eléctricas.

Se presenta ésta ley con un módulo de experimentación.

2. Estructura Atómica.

Se presentan los descubrimientos que permitieron formular nuevas teorías acerca de la composición de la materia.

2.1 Electrón.

La primer partícula sub-atómica que se descubrió fue el electrón.

2.1.1 Experimentos Rayos Catódicos.

Estos experimentos ayudaron a determinar que el electrón poseía masa y carga eléctrica.

Tabla 2.

- 2.2 Modelo atómico de Kelvin Thompson.
Se muestra el primer modelo atómico.
- 2.3 Radioactividad.
Se relaciona el descubrimiento de la radioactividad con los estudios acerca de la composición de la materia.
 - 2.3.1 Experimentos de Henry Becquerel.
Se descubrió la capacidad de ciertas sustancias de emitir rayos "invisibles" al ojo humano.
 - 2.3.2 Contribución de Pierre y Marie Curie.
La principal contribución de los esposos Curie fue el aislamiento de sustancias radioactivas y el descubrimiento de nuevos elementos.
 - 2.3.3 Descubrimiento de las partículas alfa, beta y gama.
Experimentos con sustancias radioactivas permitieron determinar la existencia de estas partículas.
- 2.4 Modelo atómico de Rutherford.
Se presentan los experimentos que ayudaron a Rutherford a formular su modelo.
 - 2.4.1 Experimento de Geiger y Marsden.
Este experimento permitió determinar la existencia de espacios internucleares.
 - 2.4.2 Descripción del Modelo.
Rutherford formula su modelo atómico en analogía al modelo del sistema solar.

Tabla 2. (Continuación). Se muestra la relación de Lecciones y Temas del Sistema Tutorial ATOMO

En la implementación del Sistema ATOMO, se utilizaron tres herramientas:

En el Mecanismo de Presentación, hemos utilizado dos herramientas: Story Board Plus y Pilot, ambas con facilidades para el manejo de gráficos y animación. El Mecanismo de Evaluación está desarrollado en Pilot. El Mecanismo de Control está construido con un lenguaje de alto nivel: Pascal (figura 12.)

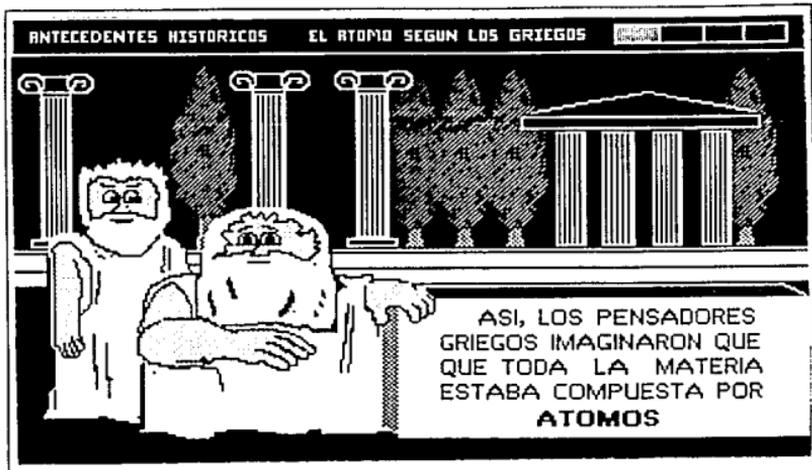


Figura 6. Pantalla del Sistema Tutorial ATOMO perteneciente a la Lección 1 ANTECEDENTES HISTORICOS, Tema 1.1 EL ATOMO SEGUN LOS GRIEGOS. Se describe la idea que tenían los antiguos griegos acerca de la composición de la materia.

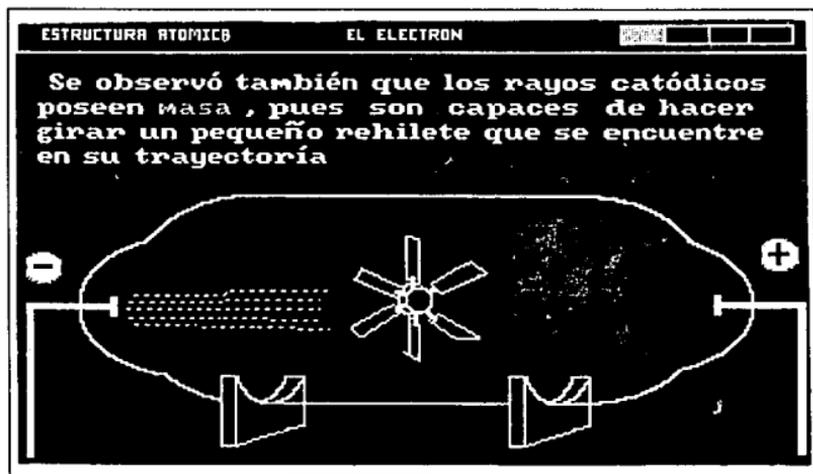


Figura 7. Pantalla perteneciente a la lección 2 ESTRUCTURA ATOMICA Tema. 2.1 EL ELECTRON, del Sistema Tutorial ATOMO. Se ilustra la trayectoria de los rayos catódicos dentro de un tubo de Crookes.



Figura 8. Pantalla perteneciente a la lección 2 ESTRUCTURA ATOMICA, Tema 2.1 EL ELECTRON, del Sistema Tutorial ATOMO. Se presenta el experimento que demuestra que los rayos catódicos se desplazan en línea recta.



Figura 9. Pantalla perteneciente a la lección 2 ESTRUCTURA ATOMICA, Tema 2.3 LA RADIOACTIVIDAD, del Sistema Tutorial ATOMO. Se ilustra la propiedad del uranio de emitir radiaciones.

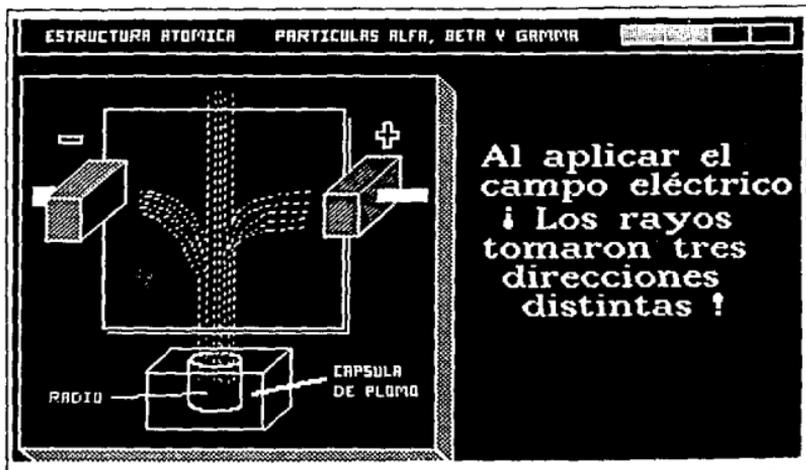


Figura 10. Pantalla perteneciente a la lección 2 ESTRUCTURA ATÓMICA, Tema 2.3.3. PARTICULAS ALFA, BETA Y GAMMA. Se muestra el comportamiento de la radiación que emite el radio en un campo eléctrico.



Figura 11. Pantalla perteneciente a la lección 2 ESTRUCTURA ATÓMICA, Tema 2.4 EL MODELO ATÓMICO DE RUTHERFORD. Se muestra el experimento que ayudó a determinar la existencia de espacios dentro del átomo.

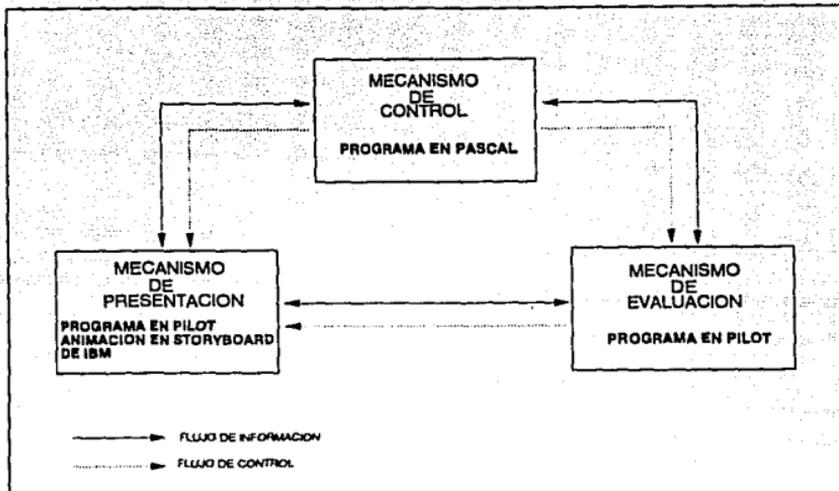


Figura 12. Implementación del Sistema Tutorial. Se muestran las herramientas con las que se implementaron los mecanismos del modelo propuesto.

4.3.1 Estructura del contenido temático de un sistema tutorial.

El contenido temático que se presenta en un Sistema Tutorial debe modularizarse de manera tal que sus temas puedan identificarse como unidades instruccionales independientes, con el fin de facilitar su manejo y operación. La figura 13 muestra la estructuración que proponemos para el contenido de un sistema tutorial.

Un Sistema Tutorial está compuesto de un conjunto finito de lecciones. La lección es la división más general que puede establecerse al contenido del sistema.

A su vez, cada lección se subdivide en partes que tiene un mayor grado de especificidad: los temas. Tendremos así que la lección engloba a un conjunto finito de temas.

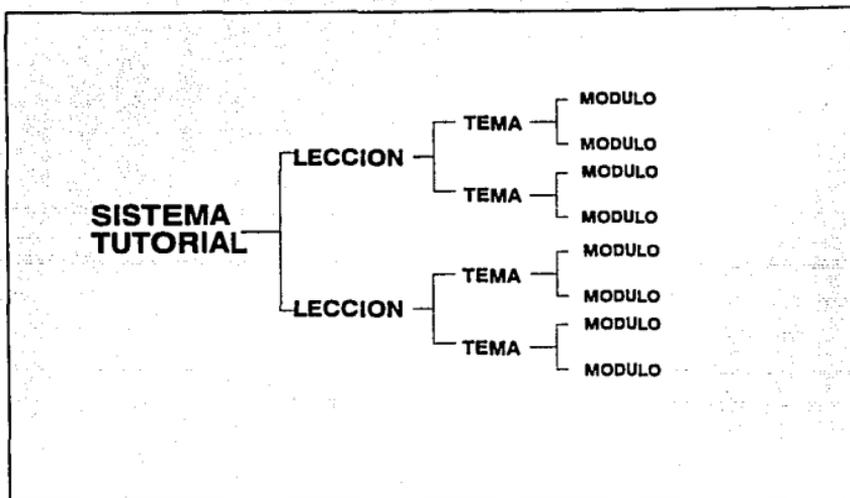


Figura 13. Estructura del Contenido Temático en un Sistema Tutorial. (Descripción en el texto).

Finalmente, cada tema está compuesto de unidades instruccionales mínimas que, por tener un objetivo específico, no pueden dividirse. A estas unidades las hemos llamado **módulos**. Un módulo consta de un conjunto de pantallas con una secuencia de despliegue específica. El módulo sólo tiene validez, instruccionalmente hablando, si se presenta en forma completa.

La estructuración del contenido temático en un sistema tutorial está determinada por el criterio del Ingeniero de Software Educativo. Es decir, las categorías antes descritas pueden ajustarse al contenido de un sistema tutorial de acuerdo a los requerimientos educativos concretos y a la complejidad de la temática a tratar (lo que para algunos puede ser un tema, para otros puede resultar una lección, o incluso un módulo).

4.3.2 Un Automata de Estados Finitos en la definición de flujos de información y control.

La función primordial del Mecanismo de Control es definir los flujos de información y control que se establecen entre los mecanismos (estos flujos se traducen en las acciones a realizar; es decir, en la estrategia de enseñanza que tendrá el sistema).

En **ATOMO** hemos diseñado un Automata de Estados Finitos para implementar la función que define los flujos de información y control entre la máquina y el usuario. Con este enfoque, integramos la plasticidad funcional de los autómatas de estados finitos al control de la interacción entre unidades de información complejas.

El autómata de Estados Finitos que emplea **ATOMO** es el siguiente:

$M = [S, I, O, F_s, F_o]$, donde

$S = \{ S_1, S_2, S_3, \dots, S_{23}, S_{24} \}$

$I = \{ \text{Acierto}, \text{Error} \}$

$O = \{ \emptyset \}$

$F_o = \{ \emptyset \}$

$F_s = \text{tabla 3}$

Estado	Entrada		Estado	Entrada	
	A	E		A	E
S_1	S_3	S_2	S_{13}	S_{14}	
S_2	S_4		S_{14}	S_{15}	
S_3	S_4		S_{15}	S_{16}	S_{15}
S_4	S_5	S_6	S_{16}	S_{17}	
S_5	S_7		S_{17}	S_{18}	S_{17}
S_6	S_7		S_{18}	S_{19}	S_{18}
S_7	S_8		S_{19}	S_{20}	
S_8	S_9	S_{10}	S_{20}	S_{22}	S_{21}
S_9	S_{10}	S_8	S_{21}	S_{20}	
S_{10}	S_{11}		S_{22}	S_{24}	S_{23}
S_{11}	S_{12}	S_{11}	S_{23}	S_{22}	
S_{12}	S_{14}	S_{13}	S_{24}		

Tabla 3. Conjunto de Funciones de transferencia al estado siguiente (F_s) en el Automata de Estados Finitos que controla a **ATOMO**. Dado un estado actual, el autómata pasa a un estado siguiente de acuerdo a la entrada que recibe en el tiempo presente. Las entradas están representadas por un Acierto o Error.

En la figura 14 se muestra el diagrama del Autómata de Estados Finitos que controla al sistema ATOMO.

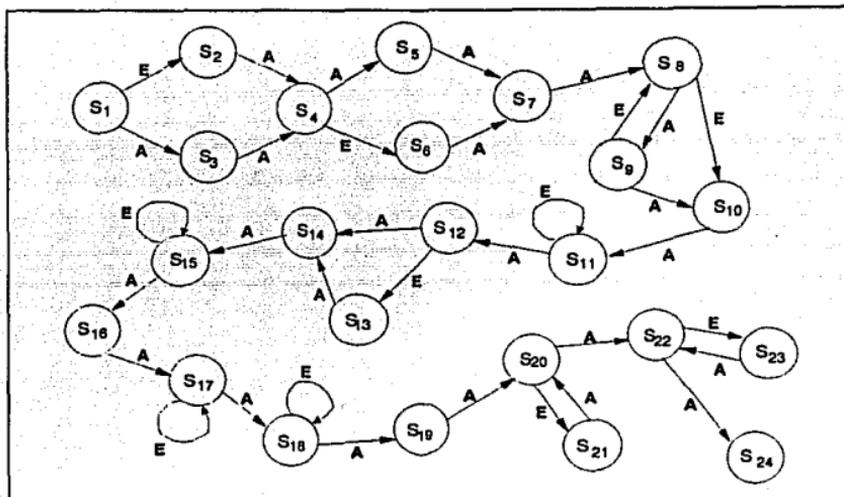


Figura 14. Diagrama del Autómata de Estados Finitos que controla a ATOMO.

Para efectos de la implementación del autómata se hacen corresponder a los módulos con estados. Eventualmente, y para esta aplicación concreta ha sido necesario agrupar los módulos de todo un tema en un solo estado. De esta manera, un estado puede incluir un módulo de despliegue, un reactivo y la evaluación de la respuesta. En otras ocasiones puede sólo incluir un reactivo.

La tabla 4 muestra el contenido temático de los estados. Por ejemplo, el estado S1 contiene el tema 1.1 "El átomo según los griegos"; el estado S14, agrupa los módulos para los temas: 2.2 "El modelo atómico de Kelvin", 2.3 "Radioactividad" y 2.3.1 "El experimento de Henry Becquerel". Varios temas se agrupan en un solo

estado cuando están relacionados entre sí y se desea conservar la continuidad en la exposición del material. La introducción de un nuevo concepto, idea, reactivo o ejercicio, puede determinar el cambio a otro estado.

ESTADO	ACCION	CONTENIDO
S ₁	ST 00100	Antecedentes Históricos.
	PILOT	El átomo según los griegos. Reactivo.
S ₂	PILOT	Reforzamiento a S ₁ en caso de error
S ₃	PILOT	Reforzamiento a S ₁ en caso de acierto.
S ₄	ST 00200	El átomo según Dalton.
	PILOT	Reactivo.
S ₅	PILOT	Reforzamiento a S ₄ en caso de acierto.
S ₆	PILOT	Reforzamiento a S ₄ en caso de error
S ₇	ST 00300	Aportaciones de Tales de Mileto
	ST 00400	y B. Franklin.
S ₈	ST 00500	Ley General de Cargas Eléctricas.
S ₉	PILOT	Reactivo Ley de Cargas Eléctricas.
S ₁₀	ST CARGAS	Experimento para Cargas Eléctricas.
	ST 00601	Rayos Catódicos y Experimentos con tubo de gas.
	PILOT	Explicación.
	ST 00604	Rayos Catódicos.
	ST 00605	Rayos Catódicos.
	ST 00606	Rayos Catódicos.
PILOT	ST 00607	Reactivo. Rayos Catódicos.
S ₁₁	ST 00800	Conceptos de Thomson.
	PILOT	Reactivo electrones, 1a. parte.
S ₁₂	PILOT	Reactivo electrones, 2a. parte.
S ₁₃	ST 00608R	Explicación alterna a reactivo S ₁₂ .

Tabla 4.

ESTADO	ACCION	CONTENIDO
S14	ST 00701 PILOT ST 00800 PILOT	Conceptos de Kelvin. Reactivo. Radioactividad. Experimentos de Becquerel. Reactivo.
S15	ST 00803 PILOT	Los Curie: Contribuciones. Reactivo.
S16	ST 00900 ST 00901	Partículas Alfa, Beta y Gama: experimentos. Posibles comportamientos.
S17	ST 00902 PILOT	Partículas Alfa, Beta y Gama: explicación de sus cargas eléctricas Reactivo Alfa, Beta, Gama: 1a. parte
S18	PILOT	Reactivo Alfa, Beta, Gama: 2a. parte
S19	ST 01001 PILOT ST 01001A PILOT ST 01002 ST 01003	Experimento con Lámina de Oro. Reactivo. Modelo de Rutherford. Reactivo. Modelo de Rutherford. Modelo de Rutherford.
S20	PILOT	Reactivo al Modelo de Rutherford: 1a parte.
S21	ST 01002	Reforzamiento a S20 en caso de error
S22	PILOT	Reactivo al Modelo de Rutherford: 2a parte.
S23	ST 01003R2	Reforzamiento a S22 en caso de error
S24	ST 01100 ST 01101	Orbitas en el modelo de Rutherford. Protón.

Tabla 4. (Continuación) ESTADOS DEL AUTOMATA, CONTENIDO Y ACCION. Cuando se activa un estado, se ejecutan una secuencia de acciones que despliegan un contenido. Todos los reactivos están construidos en Pilot. Cuando aparecen las siglas ST en la columna ACCION, se hace referencia a la ejecución de un programa en Story Board.

La estructura final del Sistema Tutorial **ATOMO** está constituida por :
 195 archivos gráficos, 25 programas en PILOT, 27 archivos de historia, 1 programa ejecutable en PASCAL, que ocupan un espacio total en disco de 2.5 MB.

5. DISCUSION.

Hasta este momento hemos presentado el Modelo para el Manejo de Flujos de Información y Control en un Sistema Tutorial y su implementación física en el sistema **ATOMO**.

Consideramos importante abrir una breve discusión en torno a cuatro aspectos relevantes: la utilización de la computadora frente a otros medios educativos; la incorporación de IAC al proceso educativo; la estructura del modelo y sus variantes de acuerdo a la implementación física y; la factibilidad de aplicar este modelo a otros tipos de software educativo.

5.1 La computadora en relación a otros medios educativos.

A medida que la ciencia pone a disposición de la sociedad nuevos elementos de comunicación, se generan opciones de aplicación al proceso instruccional. Éste es el caso de la fotografía, que permite captar imágenes del universo para mostrarlas y recorrer los detalles con propósitos educativos.

El cine, desde su nacimiento es científico. De esta manera su inclusión en los procesos instruccionales es un paso natural. En este momento la sociedad cuenta con una profusa colección de filmes que tienen como propósito instruir al espectador sobre las características y propiedades de su ambiente.

Otro caso en particular son las diapositivas que, asociadas a los registros magnéticos, se utilizan ampliamente en la construcción de audiovisuales.

La computadora ocupa un lugar privilegiado entre los medios susceptibles de uso para instrucción. En el caso particular de la sociedad mexicana, la computadora tiene un fuerte ingrediente de novedad. Como cualquier otro elemento ambiental desconocido, la computadora despierta interés y, por lo tanto, promueve una actitud exploratoria en el usuario potencial.

Mientras que las proyecciones de filmes científicos demandan una actitud contemplativa de los educandos, la computadora promueve la participación de los estudiantes para mostrar imágenes en movimiento y/o avanzar en la exposición del material.

Los libros impresos ofrecen una amplia variedad de ejercicios sobre un objeto de conocimiento. Pero nunca podría, un libro, negarse a "voltar la hoja" cuando el alumno ha cometido un error. Esta es la herramienta mínima que ofrece la computadora, como señal de alerta, ante una respuesta errónea. Pero también ofrece la posibilidad de gratificar de inmediato al educando por sus aciertos.

No abundaremos sobre la importancia de la computadora como laboratorio multidisciplinario. Pero debe destacarse que es el único medio auxiliar para la instrucción que puede estructurar trayectorias personalizadas (de acuerdo al conocimiento previo en el estudiante y a su velocidad de aprendizaje) durante la exposición de los contenidos.

En cuanto a la carencia de recursos poco podemos hacer. Es casi imposible que en una escuela, cada alumno cuente con una computadora y menos aún en las escuelas rurales. No obstante, esperamos que en un corto plazo se dediquen más recursos a la educación y se aproveche, con equipo suficiente y software educativo eficientemente desarrollado, el potencial informativo de la Instrucción Asistida por Computadora.

5.2 La incorporación de la Instrucción Asistida por Computadora al proceso educativo.

El proceso educativo se desarrolla en dos facetas complementarias pero no necesariamente ligadas: instrucción y formación.

Conceptuamos a la instrucción como la transferencia de conocimiento específico desde el educador al educando. Es una experiencia generalizada que esta faceta consume la mayoría de los recursos dedicados al proceso educativo. Por

ejemplo: un profesor invierte la mayor parte del tiempo de un curso tratando de que los educandos aprendan los mecanismos algebraicos para resolver ecuaciones. Los resultados de estos esfuerzos son siempre parciales; los educandos deben complementar su formación con ejercicios y consultas a libros que no siempre son lo suficientemente explícitos para lograr su objetivo.

Como resultado de esta práctica, el tiempo que debería dedicarse a la formación de los estudiantes se torna prácticamente inexistente.

La faceta formativa del proceso educativo la referimos al desarrollo de criterios y actitudes, de aplicación del conocimiento, que el educando necesitará para resolver los problemas que se le presenten durante su ejercicio profesional. Es evidente la importancia que esta faceta tiene para la estructuración académica de nuestros educandos. Y destaca la urgente necesidad de incorporar la Instrucción Asistida por Computadora a nuestro proceso educativo.

Los sistemas para IAC robustecen el ejercicio instruccional pero no consumen los recursos en clase. Como consecuencia, el profesor queda libre para dirigir sus esfuerzos hacia la integración del conocimiento en estrategias de razonamiento y de toma de decisiones que configuran la formación de los estudiantes.

Inicialmente la educación se separa en instrucción, que debe ser asistida por computadora, y en formación, que se mantiene bajo la responsabilidad del profesor. El ejercicio instruccional modifica el acervo de conocimiento, declarativo y procedural, del sujeto, mientras que la formación modifica su estado intelectual, esto es, el uso del conocimiento para definir estrategias de búsqueda y de resolución de problemas.

Tanto el conocimiento específico, como sus habilidades intelectuales modifican su percepción del universo y, por lo tanto, su comportamiento bajo condiciones ambientales específicas.

5.3 Los flujos de información y control en diversas implementaciones físicas.

En el Modelo para el Manejo de Flujos de Información y Control en un Sistema Tutorial, el Mecanismo de Presentación tiene la posibilidad de recibir flujos de control tanto del Mecanismo de Control como del Mecanismo de Evaluación. La forma en que se establecen dichos flujos depende de la implementación física del modelo.

Para ciertas implementaciones, el Mecanismo de Evaluación estará habilitado, junto con el Mecanismo de Control, para establecer flujos de Control sobre el Mecanismo de Presentación (figura 15).

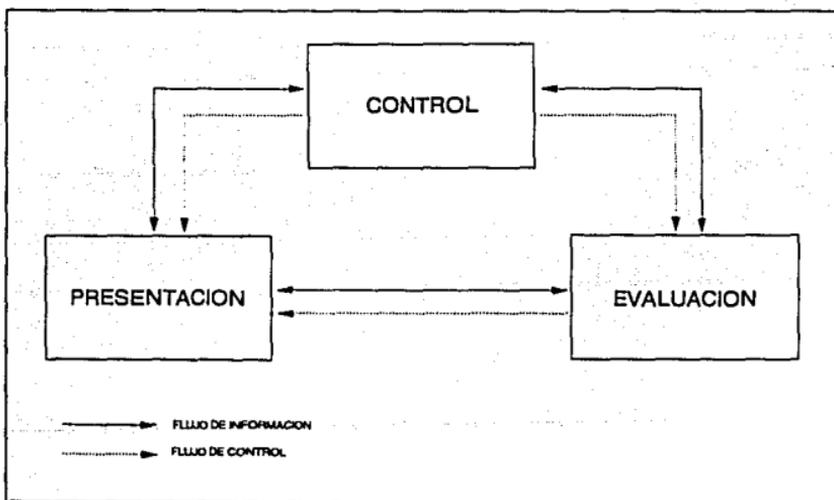


Figura 15. Flujos de Información y Control entre el Mecanismo de Presentación y los de Evaluación y Control. (Descripción en el texto).

Para otras implementaciones, los flujos que establece el Mecanismo de Control sobre el Mecanismo de Presentación, solo se realizarán a través del Mecanismo de Evaluación (Figura 16).

Una última posibilidad de implementación sugiere que el controlador sea el único que establezca los flujos de Control e Información con el Mecanismo de Evaluación y el Mecanismo de Presentación (figura 17).

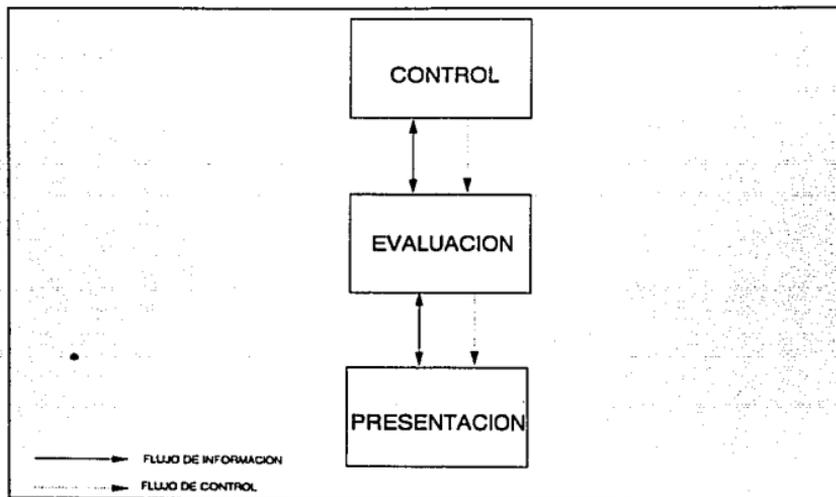


Figura 16. Flujos de Información y Control del Mecanismo de Control al Mecanismo de Presentación a través del Mecanismo de Evaluación. (Descripción en el texto).

De este modo, los modelos de las figuras 16 y 17 quedan comprendidos en el modelo general de la figura 15. Las variantes al modelo general se caracterizan por la ausencia de ciertos flujos. El establecimiento de estos flujos depende de las posibilidades que una herramienta (o el conjunto de ellas) proporcionen.

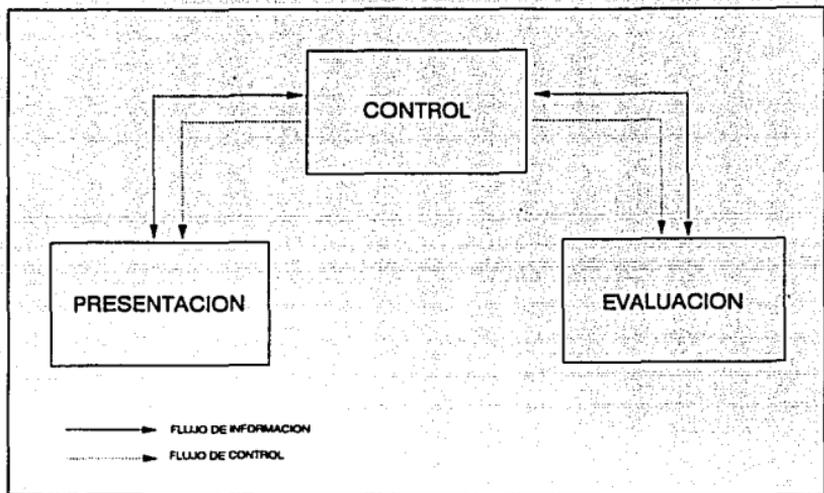


Figura 17. Flujos de Información y Control entre el Mecanismo de Control y los de Evaluación y Presentación. (Descripción en el texto).

5.4 Aplicación del modelo a la descripción de flujos en Juegos Educativos, Simuladores y Leccionarios.

El objetivo fundamental del software educativo interactivo, es transferir conocimiento específico del instructor-máquina al educando; por tanto debe realizar tres funciones básicas:

- a) presentación de información,
- b) evaluación de las interacciones alumno-máquina, y
- c) control del sistema.

Estas funciones han sido descritas en el presente trabajo como parte de los mecanismos que conforman nuestro modelo. En las secciones siguientes disertamos

en torno a la aplicación del modelo a la descripción de los flujos de información y control en los diferentes tipos de software educativo.

Aplicación del Modelo a Juegos Educativos.

Las funciones de control y presentación son las mismas que se describen para el sistema tutorial. Cabe mencionar que el registro de aciertos y errores del estudiante se implementa normalmente en términos de puntuación.

La función de evaluación depende del objetivo instruccional del juego. Se pueden evaluar respuestas, secuencias de teclas presionadas, e incluso la posición del cursor en la pantalla (figura 18).

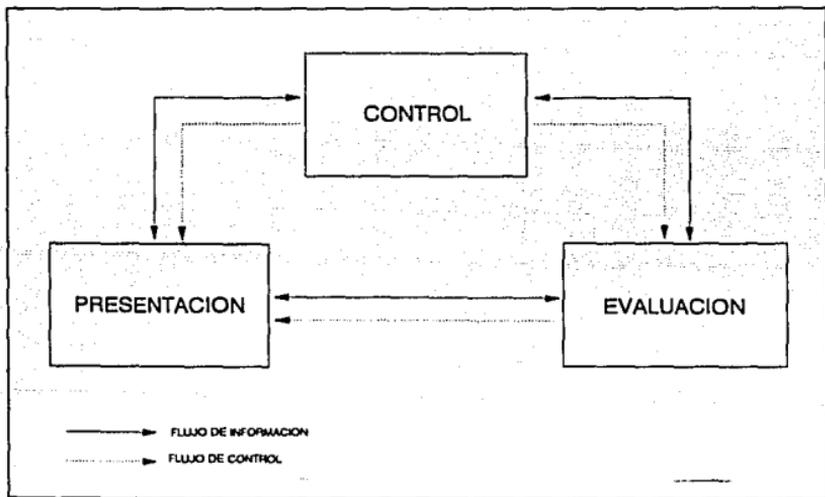


Figura 18. Aplicación del Modelo para el Manejo de Flujos de Información y Control en Juegos Educativos y Simuladores.

Aplicación del Modelo a Simuladores.

El Mecanismo de Control mantiene el estado interno de la simulación (es decir, el conjunto de variables y valores de las mismas a un tiempo determinado,

así como las relaciones existentes entre ellas), y de acuerdo a dicho estado interno, ejecuta acciones precisas (desplegar nuevas imágenes, solicitar acciones al alumno, etc.).

El Mecanismo de Evaluación tiene dos tipos de entradas: internas y externas. Las variables externas se reciben como respuesta del alumno, mientras que las variables internas se obtienen de medios tales como reloj interno, generación aleatoria, bases de datos del simulador, etc.

El Mecanismo de Evaluación toma las variables de entrada y, en base a un algoritmo, asigna valores a las variables que intervienen en la simulación (figura 18).

Aplicación del Modelo a un Leccionario.

El Mecanismo de Control contiene una secuencia fija de presentación, de manera tal que sólo existen flujos de información y control de este mecanismo al Mecanismo de Presentación.

No se utiliza ninguna de las funciones del mecanismo Evaluador (figura 19).

De esta manera, los criterios que determinan las variantes del modelo en los distintos tipos de SE son:

- a) el establecimiento de flujos de información y control entre los mecanismos del modelo y;
- b) la utilización de las funciones que cada mecanismo realiza.

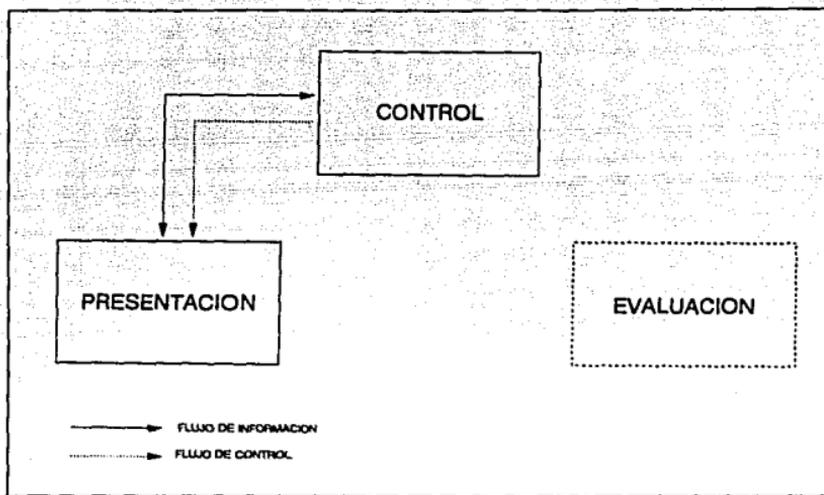


Figure 18. Aplicación del Modelo para el Manejo de Flujos de Información y Control en un Leccionario. (Descripción en el texto).

6. CONCLUSIONES.

El Modelo para el Manejo de Flujos de Información y Control en un Sistema Tutorial que se presenta en esta tesis es de aplicación general para los diversos tipos de software educativo.

Las funciones de cada mecanismo son esencialmente las mismas para cada tipo de software educativo salvo algunas variaciones dependientes del objetivo instruccional en particular.

La aplicación de un Autómata de Estados Finitos para la implementación de la función de control en el Software Educativo representa una opción viable y conveniente.

Esta conclusión se apoya en la simplicidad de programación del autómata, en la facilidad que ofrece para modularizar el código fuente, la posibilidad de generar estructuras lineales o ramificadas y, la facilidad para aplicar mantenimiento al sistema.

El licenciado en Informática debe promover la utilización de la computadora como herramienta de apoyo al proceso instruccional, de modo que coadyuve al replanteamiento de las políticas educativa e informática nacionales.

La participación del Licenciado en Informática en la Instrucción Asistida Por Computadora se realiza a través de las actividades específicas:

- 1) Seleccionar y utilizar los recursos - software y hardware- adecuados.
- 2) Aplicar la Ingeniería de Software en la implementación de software educativo de manera tal que se asegure la calidad del mismo, con actividades que van desde la planeación de proyecto, la utilización de técnicas depuradas de programación, hasta el establecimiento de un plan de pruebas y mantenimiento.
- 3) Aplicar técnicas y facilidades de integración de herramientas de construcción de SE; esto se traduce en la utilización de interfases físicas - entre dispositivos de

video, sonido, y otras computadoras- y en la construcción de interfases lógicas - entre lenguajes de programación, paquetería y aplicaciones-.

La Instrucción debe ser asistida por computadora.

En el caso de México y otros países latinoamericanos, la aplicación de las computadoras hacia el fortalecimiento del proceso de enseñanza-aprendizaje es una opción urgente en la modernización del ejercicio educativo. Consideramos que la investigación y el desarrollo sobre esta alternativa tecnológica nos permitirá resolver los problemas educativos que nos plantea el incremento en la demanda de estos servicios y el decremento, en cantidad y calidad, de los profesionales en docencia.

7. PROPUESTAS SOBRE INVESTIGACIONES FUTURAS.

En el ambiente educativo de nuestro país se ha despertado un profundo interés por desarrollar e integrar software educativo al proceso de enseñanza aprendizaje. Sin embargo, los desarrollos existentes resultan insuficientes para cubrir los requerimientos. En México es imperativo intensificar la generación y validación de software para IAC.

En el caso de Atomo, como desarrollo concreto, nos pronemos generar y aplicar un diseño experimental para la evaluación de su efecto instruccional. Por otra parte, es necesario implementar otras estructuras teóricas (árboles de decisión, inteligencia artificial, probabilidad condicionada) para el manejo de flujos de información y control en el software educativo, y comparar su eficiencia contra el modelo presentado en esta tesis. Asimismo, aplicar el modelo de control propuesto a otro sistema para IAC.

Apéndice A: Breve historia de la Educación en México.

1. **Epoca prehispánica.** La Educación era privilegio de los sacerdotes y guerreros; la transmisión de conocimientos era de maestros a discípulos seleccionados de antemano. Esto impedía al grueso de la población tener acceso al conocimiento.
2. **La Colonia.** Se presenta un conflicto cultural al llegar un nuevo idioma, costumbres, conocimientos frente a la pluralidad cultural de la región mesoamericana. La Evangelización fracasa en su intento por homogeneizar los elementos formativos de los pueblos sometidos ya que se trata de un elemento de dominación mas que de educación.
3. **México Independiente.** Nace la convicción de que se deben crear vínculos que promuevan la integración de la nueva nación mexicana; la educación toma su carácter esencial e integrador.
4. **Constitución de Apatzingán.** Se otorga al congreso federal la facultad de promover la enseñanza como origen de todo bienestar individual y social.
5. **Constitución de 1857.** Eleva a rango constitucional la libertad de enseñanza.
6. **La Ley Orgánica de Instrucción Pública de 1867-1888.** Establece una educación sin subordinación a de principios religiosos. Declara a la educación como gratuita, obligatoria y laica.
7. **1905.** Justo Sierra crea la Secretaría de Instrucción Pública y Bellas Artes.
8. **La Revolución Mexicana.** "Para entendernos, para establecer una verdadera comunicación nacional, necesitábamos, por lo menos, hablar, leer y escribir un mismo idioma" (González Pedredo, 1982).
9. **1921.** José Vasconcelos inicia la "época de oro" de la educación en México, funda la Secretaría Educación Pública (SEP) y, lo más importante : propone una identidad nacional que parta del mestizaje y concibe la educación más allá de la tarea de la escuela.
10. **Guerra Cristera.** La facción revolucionaria, reforma el artículo 3º sustituyendo la palabra *laica* por *socialista*.
11. **1945.** El artículo 3º Constitucional vuelve a reformarse (Jaime Torres Bodet) con el propósito de precisar una doctrina educativa en armonía con la esencia del proyecto nacional a forjar; esta reforma está orientada a un planteamiento ético y político. "La educación que imparta el Estado (...), tenderá a desarrollar armónicamente todas las facultades del ser humano y fomentará en él, a la vez, el amor a la patria y la conciencia de la solidaridad internacional en la independencia y en la justicia"
12. **Epoca Contemporánea.** Por nombrar algunos esfuerzos relevantes en las últimas cuatro décadas, citamos :

Gobierno de Adolfo López M. El Plan Nacional de Mejoramiento para la expansión y Mejoramiento de la enseñanza Primaria (plan de once años), la elaboración de los primeros libros de texto gratuitos.

Gobierno de Gustavo Díaz O. Se organizó el Servicio Nacional de Orientación Vocacional y se emplearon en el aula los métodos "Aprender Haciendo" y "Enseñar Produciendo".

Gobierno de Luis Echeverría A. Se expide la Ley Nacional de Educación para Adultos y se sustituye la Ley Federal de Educación; se crea el Colegio de Ciencias y Humanidades y el Colegio de Bachilleres.

Gobierno de José López Portillo. Se impulsa el sistema de enseñanza abierta.

Gobierno de Carlos Salinas de Gortari. Se modifica el artículo 3º constitucional para otorgar privilegios a la Iglesia sobre la impartición de la Educación, así como para reconocer su personalidad jurídica. Se reeditan los nuevos Libros de Texto para la Educación Básica.

Apéndice B: Listados Fuente de ATOMO

PROGRAMA FUENTE AUTOMAT.PAS

Programa en lenguaje Pascal 5.0 que implementa el autómata de estados finitos que controla a ATOMO. La entrada se lee de un archivo en disco (INPUT.DAT), la relación entrada/estado-siguiente se representa en un arreglo bidimensional. El ciclo toma la entrada y estado actual, hasta que se llegue al estado "fin".

uses Dos;

type

 nombre_modulo = string[8];

var

 comando : string[8];

 linea : string[1];

 automata : array[1..24,1..2] of integer;

 edo_actual, entrada, codigo : integer;

 modulo : array[1..25] of nombre_modulo;

 f:TEXT;

Procedure init_automata;

{ Inicialización del arreglo. Se establece la relación estado-actual (i), entrada (j) que determina el estado siguiente (k), de modo que:

 automata[i,j] := k }

begin

 automata[1,1] := 2;

 automata[1,2] := 3;

 automata[2,1] := 4;

 automata[2,2] := 2;

 automata[3,1] := 4;

 automata[3,2] := 3;

 automata[4,1] := 5;

 automata[4,2] := 6;

 automata[5,1] := 7;

 automata[5,2] := 5;

 automata[6,1] := 7;

 automata[6,2] := 6;

 automata[7,1] := 8;

 automata[7,2] := 7;

 automata[8,1] := 9;

 automata[8,2] := 10;

 automata[9,1] := 10;

 automata[9,2] := 8;

 automata[10,1] := 11;

 automata[10,2] := 10;

 automata[11,1] := 12;

 automata[11,2] := 11;

 automata[12,1] := 14;

 automata[12,2] := 13;

 automata[13,1] := 14;

 automata[13,2] := 13;

 automata[14,1] := 15;

 automata[14,2] := 14;

```
automata[15,1] := 16;  
automata[15,2] := 15;  
automata[16,1] := 17;  
automata[16,2] := 16;  
automata[17,1] := 18;  
automata[17,2] := 17;  
automata[18,1] := 19;  
automata[18,2] := 18;  
automata[19,1] := 20;  
automata[19,2] := 19;  
automata[20,1] := 22;  
automata[20,2] := 21;  
automata[21,1] := 20;  
automata[21,2] := 21;  
automata[22,1] := 24;  
automata[22,2] := 23;  
automata[23,1] := 22;  
automata[23,2] := 23;  
automata[24,1] := 25;  
automata[24,2] := 24;
```

{ Tabla de relación número del estado - nombre del programa en PILOT a ejecutar }

```
modulo[1] := 'S01';  
modulo[2] := 'S02';  
modulo[3] := 'S03';  
modulo[4] := 'S04';  
modulo[5] := 'S05';  
modulo[6] := 'S08';  
modulo[7] := 'S07';  
modulo[8] := 'S08';  
modulo[9] := 'S09';  
modulo[10] := 'S10';  
modulo[11] := 'S11';  
modulo[12] := 'S12';  
modulo[13] := 'S13';  
modulo[14] := 'S14';  
modulo[15] := 'S15';  
modulo[16] := 'S16';  
modulo[17] := 'S17';  
modulo[18] := 'S18';  
modulo[19] := 'S19';  
modulo[20] := 'S20';  
modulo[21] := 'S21';  
modulo[22] := 'S22';  
modulo[23] := 'S23';  
modulo[24] := 'S24';  
modulo[25] := 'fin';  
comando := '';  
end;
```

{ RUTINA PRINCIPAL }

begin

```

init_automata;
assign(f, 'INPUT.DAT');
Edo_Actual := 1;           { Se asigna estado inicial }
entrada := 0;
comando := modulo[1];
repeat
  Swapvectors;
  exec(getenv('TESIS'), comando);   { Se ejecuta estado }
  Swapvectors;
  if doserror < > 0 then
    writeln('dos error ', doserror);
  reset(f);
  readln(f, linea);               { Se lee entrada }
  val(linea, entrada, codigo);
  close(f);
  edo_actual := automata[edo_actual, entrada];

                                     { Se determina estado
                                     siguiente}

  comando := modulo[edo_actual];

until comando = 'fin';
end.

```

PROGRAMA FUENTE S10.PIL

Programa en lenguaje PILOT que implementa el estado 10 para el autómata de estados finitos que controla a ATOMO.

Hace llamada a programa en StoryBoard para experimento con cargas eléctrica. Despliega la primera parte de explicación para rayos catódicos en StoryBoard (00601), continúa explicación con animación en PILOT, y completa con tres historias más (00604,00604,00605). Aplica reactivo en PILOT, y finaliza con explicación en StoryBoard.

La sección "MENUS" contiene las opciones de control CONTINUAR, SALIR Y MARCAR AVANCE, SALIR SIN MARCAR AVANCE y REGRESAR LECCION ANTERIOR.

```
R: ***** ATOMO: INTRODUCCION *****
R: Estado.....: S10
R: Descripcion.....: Experimento de Cargas Eléctricas
R: Rayos catódicos y experimento con Tubo con Gas
R: Primavera 1992
R: P (control del programa): F (activa ESCAPE) U (Mayúsculas)
P:FU
DX:NIVEL$(10),ANT$(10)
```

```
r: Carga rutina (EXEC.BIN) para hacer llamada a programa externo
dx:x$(226)
fx:exec.bin
fi:0,x$
!(x$=chr(255)):FILE EXEC.BIN NOT FOUND
E(X$=CHR(255))
```

*INICIO

```
R: ***** EXPERIMENTO CON CARGAS ELECTRICAS *****
R: -Llamada a prog. externo
C: NIVEL$ = "J:INICIO"
C: ANT$ = "J:INICIO"
U:VENT
TS:M4;G0,15;F1
T: Un momento por favor...
C:X$(9,80) = "ST CARGAS"
V:X$
```

*NIV2

```
R: Llamada a prog externo
C: NIVEL$ = "J:NIV2"
C: ANT$ = "J:INICIO"
U:VENT
TS:M4;G0,15;F2
T: Un momento por favor...
C:X$(9,80) = "ST 00601"
V:X$
```

```
r: ***** PRESION BAJA DENTRO DEL TUBO... *****
```

R: A\$ pantalla completa, B\$,C\$ SPRITES
TS:G0,3;C0,F3
DX:A\$(3287)
D:B\$(2218),C\$(2218)
FX:00603.SOZ
FI:0,A\$
R: TEXTO 1
FX:00603AA.SPR
FI:0,B\$
R: TEXTO 2
FX:00603AC.SPR
FI:0,C\$

V:A\$
r:P:A1
A:
r: llamadas a sprites...

G:SB\$;G15,35;S17
r:P:A1
AX:
G:SC\$;G160,30;S17
r:P:A1
AX:
TS:G3,23;A Presion del Gas
R: Dibuja cuadro
G:H90;C1,1;g90,100;f100;r90;f300;r90;f100;r90;f300
G:G91,101;P2
TS:G7,12;A 100
TS:G9,21;A 0
R: Marca nuevo nivel, "limpia" anterior...
C: ren = 100 ; pos = 0

*CICLO
C: son = ren * 1.5
S:*10(son;1500)
C: pos = ren + 1; ren = ren + 10
G:H90;C1;G90,ren;F100
G:G91,pos;P0
J!(ren < 170);CICLO
S:*20(son;1800)
AX:

*NIV3
R: Llamada a prog externo
C:NIVEL\$ = "J:NIV3"
C:ANT\$ = "J:NIV2"
U:VENT
TS:M4;G0,15
T: Un momento, estoy pensando
: lo que sigue...
C:X\$(9,80) = "ST 00604"
V:X\$

*NIV4

R: Llamada a prog externo
C:NIVEL# = "J:NIV4"
C:ANT# = "J:NIV3"
U:VENT
TS:M4;G0,15
T: Un momento, estoy dibujando
: lo que sigue...
C:X#(9,80) = "ST 00605"
V:X#

*NIV5

R: Llamada a prog externo
C:NIVEL# = "J:NIV5"
C:ANT# = "J:NIV4"
U:VENT
TS:M4;G0,15;F2
T: Solo me toma unos momentos
: continuar...
C:X#(9,80) = "ST 00606"
V:X#

R: *** REACTIVO CARAC. RAYOS CATODICOS ***

R: Declaracion de pantallas...
DX:A#(7101)
FX:00600R.SQZ
Fi:O,A#

TS:M5;E0;F3;c0
V:A#

R: C R

TS:G27,13;F3;C1
P:A1
AH:
TS:G1,14;F2
M:IF
TN:ES FALSO! los rayos
: catodicos Si tienen masa!
TY:Correcto!

TS:G27,17;F3;C1

P:A1
AH:
TS:G1,18;F2
M:Vlv
TN:ES VERDADERO! los rayos
: catodicos viajan en linea recta!
TY:Excelental

TS:G27,21;F3;C1

P:A1
AH:

```

TS:G1,22;F2;CO
M:FIN
TN:ES FALSO! Se emiten desde
: el polo negativo (catodo)
TY:Muy bien!
w:150
*NIV6
R: Llamada a prog externo
C:NIVEL# = "J:NIV6"
C:ANT# = "J:NIV5"
U:VENT
TS:M4;GO,15;F1
T: Un momento, estoy preparando
: lo que sigue...
C:X$(9,80) = "ST 00607"
V:X#
C:RES = 1
FX:INPUT.DAT
FO:0,RES
J:FIN
*VENT
R: ..... MENU .....
DX:M$(10),M1$(4)
TS:M2;GO,15
T:
:
: #3 -----
: | PRESIONA #12 F1 #3 PARA SALIR AL #12MENU DE LECCION #3 |:::
: | |:::
: | O #14 [ENTER] PARA CONTINUAR #3 |:::
: |-----
: :::
: :::
: #15
AX:
E:
R: .....
R: Procedimiento para secuencias de escape
R: C (compute); ASC(%B(L)) calcula el valor ascii del ultimo caracter en buffer
*SYSX C:L = LEN(%B); K = ASC(%B(L))
R:Evalua teclas de funcion
R: F1 menu de leccion
J(K = 187):MENU
R: F3 MARCA/SALIR
J(K = 189):MSALIR
R: F5 SALIR
J(K = 191):SALIR
R: F9 ANTERIOR
J(K = 195):ANT
R: F10 SEGUIR
J(K = 196):SIGUE
E:SIGUE
*MENU
TS:M2
T:

```


INDICE DE FIGURAS Y TABLAS

Pag.

Figura 1. Clasificación de los tipos de Software Educativo según Hong (1987).	12
Figura 2. Gráfico del Automata de Estados Finitos que reconoce palabras con un número impar de unos y al menos un cero.	27
Figura 3. Esquema General sobre el Flujo y Transformación de la Información para la Construcción de Software Educativo.	30
Figura 4. Refinamiento del Proceso de Aplicación Experimental del Esquema General sobre el Flujo y Transformación de la Información para la Construcción de Software Educativo.	31
Figura 5. Diagrama con los flujos de información y control entre los tres mecanismos que constituyen a un sistema tutorial.	33
Figura 6. Pantalla del Sistema Tutorial ATOMO perteneciente a la Lección 1 ANTECEDENTES HISTORICOS, Tema 1.1 EL ATOMO SEGUN LOS GRIEGOS. Se describe la idea que tenían los antiguos griegos acerca de la composición de la materia.	38
Figura 7. Pantalla perteneciente a la lección 2 ESTRUCTURA ATOMICA, Tema 2.1 EL ELECTRON, del Sistema Tutorial ATOMO. Se ilustra la trayectoria de los rayos catódicos dentro de un tubo de Crookes.	38
Figura 8. Pantalla perteneciente a la lección 2 ESTRUCTURA ATOMICA, Tema 2.1 EL ELECTRON, del Sistema Tutorial ATOMO. Se presenta el experimento que demuestra que los rayos catódicos se desplazan en línea recta.	39
Figura 9. Pantalla perteneciente a la lección 2 ESTRUCTURA ATOMICA, Tema 2.3 LA RADIOACTIVIDAD, del Sistema Tutorial ATOMO. Se ilustra la propiedad del uranio de emitir radiaciones.	39
Figura 10. Pantalla perteneciente a la lección 2 ESTRUCTURA ATOMICA, Tema 2.3.3. PARTICULAS ALFA, BETA Y GAMMA. Se muestra el comportamiento de la radiación que emite el radio en un campo eléctrico.	40
Figura 11. Pantalla perteneciente a la lección 2 ESTRUCTURA ATOMICA, Tema 2.4 EL MODELO ATOMICO DE RUTHERFORD. Se muestra el experimento que ayudó a determinar la existencia de espacios dentro del átomo.	40

Figura 12. Implementación del Sistema Tutorial.	41
Figura 13. Estructura del Contenido Temático en un Sistema Tutorial.	42
Figura 14. Diagrama del Autómata de Estados Finitos que controla a ATOMO.	44
Figura 15. Flujos de Información y Control entre el Mecanismo de Presentación y los de Evaluación y Control. (Descripción en el texto).	50
Figura 16. Flujos de Información y Control del Mecanismo de Control al Mecanismo de Presentación a través del Mecanismo de Evaluación.	51
Figura 17. Flujos de Información y Control entre el Mecanismo de Control y los de Evaluación y Presentación.	52
Figura 18. Aplicación del Modelo para el Manejo de Flujos de Información y Control en Juegos Educativos y Simuladores.	53
Figura 19. Aplicación del Modelo para el Manejo de Flujos de Información y Control en Leccionarios.	55
Tabla 1. Función F_s (del estado siguiente) en el Autómata de Estados Finitos del ejemplo.	26
Tabla 2. Lecciones y Temas del Sistema Tutorial ATOMO.	36
Tabla 3. Conjunto de funciones de transferencia al estado siguiente (F_s) en el Autómata de Estados Finitos que controla a ATOMO.	43
Tabla 4. Estados del Autómata, Contenido y Acción.	45

INDICE DE AUTORES.**Pags.**

ARAOZ-Castillo, M.	3
BARKER, Philip.	15,16
BURKE, L. Robert.	10
CAMPOS-Cázares R.	10
CUELI, J.	1
GUEVARA-Pozas, A.D.	3,5,8,32
HONG, Jon-Chao.	8,11,12
HUDSON, Keith.	12,16,18,23,28
NAFINSA.	1
LAWRENCE, S.	23
MENDEZ, J.S.	1
PRICE, Robert V.	8,12,15,16,24
RICH, Elaine.	23
SANDERS, Donald H.	2
SLEEMAN, D. y Brown J.S.	23,24
SMITH, S.	8,10,24
ZEA, Leopoldo.	1

BIBLIOGRAFIA.

ARAOZ-Castillo, M. y Guevara-Pozas, A.D. (1991).
PRINCIPIOS PARA EL DISEÑO DE SISTEMAS TUTORIALES A TRAVÉS DE UN LIBRO DISKETTE.

(Ed). En prensa.

BARKER, Philip. (1987).
AUTHOR LANGUAGES FOR CAL.
Mc. Millan Education LTD (Ed). London.

BURKE, L. Robert. (1986).
COMPUTER ASSISTED. SOURCE BOOK.
Prentice-Hall (Ed). U.S.A.

CAMPOS-Cázares R. y Guevara-Pozas, R. (1991).
SEGI : UN SISTEMA DE ENSEÑANZA SOBRE SEGURIDAD INFORMATICA.
(Ed.) En prensa

CRUZ-GARRITZ, D. et al. (1987).
ESTRUCTURA ATOMICA: UN ENFOQUE QUIMICO.
Addison-Wesley Iberoamericana (Ed). México.

CUELI, J., et al. (1990).
VALORES Y METAS DE LA EDUCACION EN MEXICO.
Papeles de Educación No. 1.
SEP (Ed). México. La Jornada, Dic. de 1990.

GERSTING, Judith L. (1987).
MATHEMATICAL STRUCTURES FOR COMPUTER SCIENCE.
W. H. Freeman and Company (Ed). U.S.A. 2a. ed.

GUEVARA-Pozas, A.D. (1992).
LA FIGURA COMPUTARIZADA EN EL PROCESO INSTRUCCIONAL C.I.S.E.-
U.N.A.M. (Ed.) En prensa.

HONG, Jon-Chao. (1987).
CAI SYSTEM CONSIDERATION : A MODEL OF COURSEWARE DESIGN
STRATEGY. Computer Education Review.
Vol. 6 Junio, 1987. U.S.A.

HUDSON, Keith. (1986).
INTRODUCING CAL : A PRACTICAL GUIDE TO WRITING COMPUTER-ASSISTED
LEARNIG PROGRAMS.
Chapman and Hall (Ed). London.

IBARROLA, M. (1985).
LAS DIMENSIONES SOCIALES DE LA EDUCACION.
SEP Cultura (Ed). México.

KOLMAN, Bernard y Busby, Robert C. (1986).
ESTRUCTURAS DE MATEMATICAS DISCRETAS PARA LA COMPUTACION.
Prentice Hall (Ed). México.

LA ECONOMIA MEXICANA EN CIFRAS.
NAFINSA (Ed). 1990

LAWRENCE, S. (1987).
SOFTWARE ENGINEERING: THE PRODUCTION OF QUALITY SOFTWARE.
Macmillan Publishing Company (Ed). London.

MENDEZ, J.S. (1991).
PROBLEMAS ECONOMICOS DE MEXICO.
McGraw Hill (Ed). México.

PRICE, Robert V. (1991).
COMPUTER AIDED INSTRUCTION. A GUIDE FOR AUTHORS.
Brooks/Cole Publishing Company (Ed). U.S.A.

RICH, Elaine. (1988).
ARTIFICIAL INTELLIGENCE.
Mc.Graw-Hill International Editions (Ed). U.S.A.

RINCON A. (1991).
ABC DE QUIMICA.
Herrero (Ed). México.

SANDERS, Donald H. (1985).
INFORMATICA: PRESENTE Y FUTURO.
Mc. Graw Hill (Ed). México, 1a. Edición.

SLEEMAN, D. y Brown J.S. (1982).
INTELLIGENT TUTORING SYSTEMS.
Academic Press (Ed). London.

SMITH, S. (1988).
COMPUTER-BASED TUTORIALS TAKE OFF.
Nature Review Vol. 5, mayo, 1988. U.S.A.

ZEA, Leopoldo. (1983).
INTRODUCCION A LA FILOSOFIA.
U.N.A.M. (Ed). México.