

# HERRAMIENTAS GENERALES PARA EL DESARROLLO DE JUEGOS Y SOFTWARE EDUCATIVO

**Fernando Galindo Soria**

IPN  
UPIICSA  
Dpto. de Computación  
México, D.F

DGIT  
CENIDET  
Cuernavaca, Mor.

Septiembre de 1989

## RESUMEN

En este trabajo se presenta un conjunto de herramientas para desarrollar juegos y Software Educativo. Como primer punto se presenta el concepto de Guión Computacional y sus diferencias con el Guión de Cine y Televisión, enfatizando las características de interacción con la computadora y el manejo de personajes y su apropiación por parte del usuario del software.

A partir de esto se ve que en el guión se pueden encontrar las componentes básicas de un sistema de información (Estructura de datos, acciones y estructura del sistema) y en particular se observa que a partir del guión se pueden obtener en forma natural:

- a) Un conjunto de imágenes (fondos y objetos) y sonidos que conforman la salida del sistema.
- b) Un conjunto de acciones que se tienen que realizar para validar los resultados del usuario y en su caso, plantear nuevos problemas a éste.
- c) Un autómatas o gramática que represente la estructura del juego Software Educativo.

A partir de lo anterior, se ve cómo desarrollar los elementos computacionales del sistema y en particular Editores de Imágenes, Generadores de Programas y un Mecanismo de control, los cuales integrados nos proporcionan la arquitectura general de múltiples programas de Juegos Educativos.

## INTRODUCCIÓN

En este documento se presenta un conjunto de resultados de una investigación realizada por profesores y alumnos del Instituto Tecnológico de Zacatepec y del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET-DGIT) de la Ciudad de Cuernavaca, Mor., sobre los métodos y herramientas orientadas al desarrollo de Software Educativo y generalizado posteriormente para desarrollar cualquier tipo de software altamente interactivo (aquél en el que el usuario lleva prácticamente el control del proceso y espera una respuesta a sus acciones en tiempo real) como por ejemplo Simuladores y Juegos por Computadora.

## 1. ANÁLISIS

Cuando se comienza el proceso de análisis para desarrollar un software educativo lo primero que se detecta es que se tienen una serie de requerimientos que lo hacen muy diferente al software tradicional incluyendo entre otros, los siguientes:

- a) El usuario es normalmente no-experto en Informática y no está interesado en aprender a programar para usar estas herramientas.
- b) Espera tener un control completo sobre el sistema y una respuesta en tiempo real sobre sus acciones.
- c) Espera que el sistema no se caiga independientemente de las acciones que ejecute.

Es el segundo requerimiento el que nos ha hecho replantear completamente el método de desarrollo de tal manera que no permita desde la etapa de ANÁLISIS reflejar al máximo la necesidad de controlar en tiempo real las acciones del sistema.

### 1.1 El guión computacional

Es dentro de este contexto donde después de que los estudiantes de la Especialidad en Computación Educativa del CENIDET investigaron sobre los diferentes medios que se utilizan para apoyar el proceso de Enseñanza-Aprendizaje, incluyendo medios estáticos (proyecciones, radio, TV), y medios dinámicos (Teatro y enseñanza participativa) se llegó al concepto de GUIÓN COMPUTACIONAL, el cual absorbe la idea de un Guión de Cine o Televisión pero integrando como eje de control del guión una columna donde se presentan las acciones realizables por parte del usuario.

Lo anterior surgió como resultado de integrar la capacidad interactiva de una computadora con los estudios realizados sobre los mecanismos de aprendizaje, ya que se ha detectado que el ser humano:

- Recuerda alrededor del 10% de lo que lee.
- 20% de lo que escucha.
- 30% de lo que ve.
- 50% de lo que ve y escucha.
- 70% de lo que escucha y analiza
- 90% de lo que ve y hace.

De donde se ve, que un mecanismo que propicia la acción por parte del participante, apoya en gran medida el proceso de aprendizaje.

Ahora bien, ni la radio ni la televisión permiten la interacción por parte del receptor y no es hasta el surgimiento de la computadora, donde ya se cuenta con un mecanismo de este tipo, es por eso que, importante enfatizar que la ventaja de la computadora se encuentra en esa capacidad de hacer que brinda al usuario sobre los otros medios, por lo que, un guión computacional es un documento formado básicamente por las siguientes columnas:

Acciones del Usuario	Respuesta del Sistema Imágenes Sonidos
Señales, datos ordenes, preguntas etc., que realiza el usuario sobre el sistema.	Principales imágenes y sonidos con los que el sistema refleja ante el usuario los resultados de sus acciones.

Como se podrá observar el guión consta de 2 columnas principales en las cuales se representan las acciones del usuario y las respuestas del sistema.

En el caso de las respuestas es común tener dos columnas, en la primera se reflejan las transformaciones que se llevan a cabo en las imágenes y en la segunda los sonidos involucrados.

En el guión se pueden reflejar en una forma relativamente sencilla para un no-experto en Informática, los requerimientos de entrada y salida del sistema y el orden en que se deben ejecutar las acciones de acuerdo al estado del sistema y a las nuevas señales dadas por el usuario.

## 1.2 Consideraciones para el desarrollo de guiones computacionales

El guión computacional es por sí solo únicamente una herramienta de análisis y lo mismo puede servir para hacer un tutor o un simulador, dependiendo de la habilidad del desarrollador y de su capacidad para reflejar sus ideas en forma creativa. Sin embargo, ya existen algunos trabajos previos que pueden facilitar el desarrollo de un buen guión. Entre septiembre y noviembre de 1988 un conjunto de alumnos y profesores del Instituto Tecnológico de Zacatepec (ITZ), desarrollaron una propuesta propia para hacer software educativo, en particular sobre Física y que más adelante ha mostrado su bondad en otras áreas.

Tradicionalmente en la enseñanza de la Física se transmite al alumno un conjunto de fórmulas y se le pide que aprenda a aplicarlas para resolver un problema dado, o sea que se enseña al alumno a razonar en forma deductiva; el problema se presenta porque prácticamente nunca se le explica o enseña a obtener la fórmula, o sea a razonar en forma inductiva. El motivo básico que se aduce para no enseñar a encontrar la fórmula es la falta de tiempo, ya que para llegar a la fórmula, un físico requiere de una gran cantidad de experimentos, sin embargo, y con el uso de la computadora esto ya no es necesariamente cierto, ya que en la misma computadora se puede simular un laboratorio donde el alumno pueda repetir una y mil veces los experimentos (Como se puede observar en los trabajos de la Fundación Rosenblueth) y llegar a encontrar sus propias reglas que expliquen el comportamiento de un fenómeno. A partir de lo anterior, en el ITZ propusieron una estructura de guión con las siguientes etapas:

### a) Búsqueda de Variables.

Se pretende que el estudiante detecte cuáles son las variables relevantes de un fenómeno.

Para lograr lo anterior se presenta un fenómeno (Por ejemplo caída libre), y se grafica su comportamiento de acuerdo a un conjunto de variables no necesariamente relevantes.

A partir de ahí y después de varios experimentos se pide al participante que proponga las variables relevantes y se le presenta nuevamente el fenómeno, la gráfica que se obtiene con las variables propuestas y la grafica que describe el fenómeno, con el fin de que determine el grado de certeza de su propuesta, repitiéndose el ciclo hasta contar con las variables relevantes.

b) Búsqueda de relaciones.

En esta etapa se busca que el alumno proponga una fórmula que describa el comportamiento del problema.

Para lo cual se presenta el fenómeno y la gráfica obtenida con las variables relevantes y se le pide que proponga una fórmula que explique el fenómeno.

Nuevamente se presenta el fenómeno y su gráfica, pero, ahora se incluye la gráfica obtenida en la ecuación propuesta, con el fin de que el alumno detecte su grado de certeza y así se continúa hasta que la fórmula corresponda al problema.

c) Explicación.

Se busca que el participante reafirme su propuesta explicando el fenómeno y su comportamiento.

d) Etapa Deductiva

Hasta aquí termina la etapa inductiva y se continúa con una etapa deductiva donde el alumno aplica la fórmula (descubierta por él) a problemas diferentes.

e) Generalización

En esta etapa se busca que el alumno realice, experimente y descubra lo que pasaría si ...

i) Cambian los parámetros iniciales. Gravedad, velocidad inicial, ángulo, etc.

ii) Cambia el comportamiento o las características del espacio. Caída de objetos en el agua, juego de básquetbol en la Luna, etc.

iii) Cambia el Universo. En esta última etapa el alumno puede reproponer las ecuaciones y observar lo que pasaría en un universo regido por las nuevas ecuaciones.

Lo anterior es una idea muy general pero enfatiza la búsqueda de guiones que fomenten el Razonamiento Inductivo, Deductivo y de Generalización, como en este caso.

Otra de las ideas nucleares del software basado en guión surge al recuperar algunos de los conceptos del guión tradicional como son la escenografía, los actores y el personaje, en el caso de la escenografía y los actores se traducen en los conceptos de fondo y objeto, donde un fondo es una imagen fija que se mantiene durante toda una etapa del juego o proceso de enseñanza y un objeto es todo aquello que se mueve o cambia tomando como telón el fondo.

El concepto de Personaje es tal vez uno de los conceptos nucleares del desarrollo del software interactivo, ya que se ha detectado que es común que el participante en un juego de Software Educativo se apropie de las acciones de la figurita que está en la pantalla y las tome como propias, oyéndose comúnmente la frase "me quedan pocas vidas" este hecho que puede ser altamente negativo si se desconoce o es mal usado, es tal vez el núcleo del proceso interactivo, ya que el usuario tiene la vista en el personaje y espera que actúa de acuerdo a su requerimientos, lo cual (se puede aprovechar para enfatizar cierto tipo de acciones o resultados, por lo que, al desarrollar el guión es importante detectar quién es el personaje). Por ejemplo, en caída libre puede ser la pelotita que cae; en el ciclo del agua puede ser una gota de lluvia.

Ya que se tiene el personaje, es conveniente marcarlo (ponerle patitas o caritas) y enfatizar sus acciones, con el fin de que resalten los fenómenos que interesa que capte el participante.

## 2. DISEÑO

Ya que se tiene un guión computacional, una de las cosas que sorprenden más, en lo fácil que se puede obtener el diseño del sistema a partir de éste.

Tradicionalmente se considera que un sistema de infamación está integrado por una estructura de Datos, un conjunto de Acciones y una Estructura u orden en el que se ejecutan las acciones sobre el sistema.

### 2.1 Datos, Imágenes y Sonidos

En el caso del Software Interactivo este modelo se conserva pero incluye algunas consideraciones propias, ya que en este caso surgen las preguntas: Dónde están los Datos? Cuáles acciones?, sin embargo, al observar un mirar un guión se observa que todos estos componentes se encuentran presentes pero con otros nombres, ya que aparte de los datos tradicionales (como velocidad, tiempo, gravedad, en el caso de caída libre) se encuentran involucrados "datos" como:

- Fondos
- Objetos
- Personajes
- Sonidos

Incluyendo sus características y posiciones, de donde al diseñar software educativo o de juegos se tienen que considerar los archivos de Imágenes, Sonido y Personajes, y obtener a partir del Guión estos datos

### 2.2 Acciones, Modelos y Control del Fenómeno

Las acciones involucradas son más difíciles de detectar, ya que, por lo común en el guión aparecen los "monitos" y lo que hacen muy claramente descrito, pero los procesos que modelan internamente las acciones de esos "monitos" no se describen nunca.

Dentro de un software interactivo se tienen básicamente dos tipos de acciones:

- a) Las primeras, son aquellas acciones que se ven, como calcular el valor de una suma y compararla con el resultado del usuario.
- b) Las segundas, son las acciones que no se ven y que normalmente están incluidas en el modelo donde se refleja el comportamiento del sistema, de tal manera que para cada acción del usuario ocurra una transformación en el modelo y ésta a su vez se refleja en un conjunto de imágenes y sonidos, que permiten validar sus actos al usuario.

Normalmente este modelo se construye a partir de algún modelo (Cuantitativo o cualitativo) del fenómeno que se está presentando al usuario, por ejemplo: puede estar formado por las ecuaciones que rigen la caída libre y un conjunto de reglas que permiten transformar los datos del usuario (movimiento del cursor, de una palanca, una letra) en valores aceptados por el modelo y otro conjuntos de reglas que permiten transformar los resultados obtenidos, en posiciones u órdenes que permitan ver los resultados (posición de un objeto en la pantalla, número de un mensaje, número para cambiar de estado, aumento del brillo, etc.)

### 2.3 Estructura del Sistema, Gramáticas y Autómatas

La tercera componente del sistema nos refleja su estructura, o sea el orden en que se aplican las acciones sobre datos a partir de las entradas y del estado actual del sistema.

En el caso de software interactivo, la estructura se encuentra directamente en el guión, ya que en éste aparece explícitamente el orden de ejecución de las acciones, sin embargo, se presenta el problema de reflejar este guión dentro de un sistema computacional.

La solución trivial consiste en reescribir el guión en pseudocódigo o en algún lenguaje de programación y de ahí pasarlo a la computadora, sin embargo, al aplicar esta solución se detecta que el software generado es por lo común muy extenso, lleno de código de apoyo (mensajes, llamados archivos de imágenes, validación de datos) y altamente repetitivo. Al analizar el problema con más detalle, se ve prácticamente todo el programa se repiten las mismas acciones (ecuaciones del modelo) y llamados a imágenes, sólo que en diferente orden dependiendo de lo que realice el usuario, por lo que una propuesta más avanzada y relativamente fácil de instalar consiste en hacer un programa general que a partir de los actos del usuario y de acuerdo al estado en que se encuentra el sistema (que está haciendo) decida cuál es la siguiente acción y la mande a ejecutar.

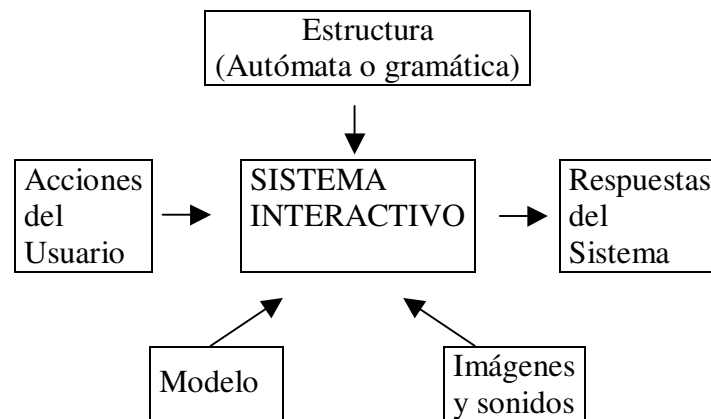
Este tipo de mecanismo es muy fácil de construir y lo único que requiere es de una tablita donde se indique a la computadora qué hacer.

	Señales (Señales de Usuario)
Estado (Qué está haciendo la computadora)	Nuevas acciones y cambios de estado

A esta tabla se le conoce como Autómata y nos permite representar al guión en una forma muy condensada.

Por ejemplo: Si lo que se tiene es el guión de un juego de aventura o un laberinto, cada uno de los cuartos que forman el laberinto equivale a un estado y cada una de las acciones del usuario que involucra pasar de un cuarto a otro, equivale a una señal para cambiar de estado.

Por lo que si integramos todos los componentes anteriores, la arquitectura base de un software interactivo es:



Donde a partir de las Acciones del Usuario (Entradas) y del estado en que se encuentra el sistema, se puede encontrar en la tabla de la estructura qué partes del modelo (ecuaciones, validaciones) se deben aplicar para obtener los valores que permitan integrar las imágenes y sonidos en una respuesta del sistema.

### 3. HERRAMIENTAS COMPUTACIONALES

Como se puede observar, la arquitectura de un software interactivo involucra la necesidad de mecanismos que permita manejar la estructura del sistema, el modelo y los datos principalmente Imágenes y Sonidos.

Nuevamente el enfoque tradicional nos orienta a desarrollar un gran programa que refleje en Pseudocódigo cada parte del sistema y que nos permita pasarlo directamente a algún lenguaje de computación y de ahí a la máquina.

Sin embargo, al estudiar el problema se ha detectado que existen un conjunto de herramientas de propósito general y que en su momento facilitan la programación del software educativo.

Estas herramientas consisten básicamente en Editores de Imágenes, Controladores de Sistemas y Generadores de Programas.

El principal argumento para no usar este tipo de herramientas se presenta por "la dificultad" que involucra su desarrollo, sin embargo, éste no es un argumento válido, ya que una herramienta automatizada se desarrolla una sola vez y se puede utilizar para múltiples problemas, y por otro lado, como se verá a continuación ya se cuenta con métodos para desarrollar este tipo de herramientas en forma relativamente sencilla, de tal forma que desarrollar un sistema de acuerdo al enfoque de herramientas automatizadas, muchas veces es más fácil y rápido que la programación tradicional.

### 3.1 Controlador del sistema

El controlador del sistema recibe como entrada la acción que está ejecutando el usuario, el estado actual y la tabla de estructura, y su función consiste en indicar qué acciones se deben ejecutar, en qué orden y en su momento qué imágenes y sonidos y con qué característica se deben de presentar al usuario. Por facilidad consideraremos que la Tabla de Estructura tiene la siguiente forma:

		señal		
		s1	s2	s3
estado	e1	e3 ...	...	.....
	e2	.....	...	.....
	e3	.....	...	.....
	e4	.....	...	.....

En la tabla se muestra como ejemplo que: dado que se esta en el estado **e1** y el usuario manda la señal **s1** el sistema pasa al estado **e3**.

Como se puede ver, el controlador es un programa que únicamente consulta una tabla y manda ejecutar acciones de acuerdo a lo que indica la tabla. De donde si por ejemplo, se está desarrollando un correspondiente a cada uno de los cuartos o pasillos del laberinto, las señales corresponderían a los movimientos del cursor y las acciones a los problemas o trampas involucradas en cada parte del laberinto, en su momento mientras el usuario no traspase una frontera, el sistema lo mantiene en la misma parte y cuando traspasa la frontera involucra un cambio de señal y por tal motivo un cambio del estado (que se refleja cambiando la imagen y los objetos) y de las acciones involucradas.

Como toda esa información se encuentra en una tabla, cambiar de un juego a otro, únicamente involucrar el cambio de una tabla por otra, pero sin modificar la programación del controlador.