

17

**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES "ACATLÁN"**

288938

**"DISEÑO DE SOFTWARE EDUCATIVO PARA EL APOYO EN EL
APRENDIZAJE DE LOS NIÑOS DE EDUCACIÓN PRIMARIA. CASO
PRÁCTICO EN UNA INSTITUCIÓN EDUCATIVA DEL SECTOR PRIVADO"**

MEMORIA DE DESEMPEÑO PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA

FERNANDO GÁMEZ VÁZQUEZ

ASESOR: OCEAN. ANTONIO GAMA CAMPILLO



DICIEMBRE DEL 2007



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedico este trabajo especialmente a mi

MADRE, con un enorme
agradecimiento, ya que gracias a su
esfuerzo y sacrificio por sacarnos
adelante he podido llegar a ver
realizados estos momentos.

Te quiero mucho.

*A la Universidad Nacional Autónoma de
México y en especial a la ENEP
"Acatlán", por haberme permitido el
estudiar en sus aulas.*

*A todos y cada uno de los profesores que
me impartieron su conocimiento a lo largo
de mi vida estudiantil*

¡Muchas gracias!

*A mis hermanos Alberto, Luz Maria,
Martín, Leticia, Luis Manuel y Gerardo
como un estímulo de superación
constante en cada una de las actividades
que realicen.*

*A C. Miriam Castillo Ruiz, porque este
trabajo representa la culminación de una
etapa muy importante en mi vida y quiero
compartirla contigo.*

Te quiero mucho.

*A mis sobrinos Jacqueline, Javier,
Mauricio, Luis Eduardo, Ana Laura,
Daniela, Tania y José Fernando,
esperando que todos ellos, algún día, vean
realizados sus objetivos.*

*A mis amigos de la ENEP "Acatlán",
gracias por su apoyo y compañerismo en
las diferentes etapas dentro de la
carrera.*

Pero sobre todo Gracias a Dios

*Fernando Gámez Vázquez.
Diciembre del 2000.*

ÍNDICE

	<i>Página</i>
INTRODUCCIÓN.....	1
CAPITULO I. EL SOFTWARE EDUCATIVO COMO UN APOYO DIDÁCTICO EN UNA INSTITUCIÓN EDUCATIVA DEL SECTOR PRIVADO	
1.1 Marco teórico.....	4
1.2 Introducción al problema.....	5
1.2.1 Necesidades académicas observadas.....	7
1.3 El software educativo y su clasificación.....	8
1.3.1 Programas de introducción de conceptos.....	8
1.3.2 Programas de repaso.....	8
1.3.3 Programas de ejercicio y práctica	10
1.4 Ventajas y errores más comunes al implantar el software educativo.....	11
1.5 Investigación de mercado en diferentes tiendas y almacenes acerca del software educativo.....	12
1.5.1 Análisis de la investigación de mercado.....	19
CAPITULO II. EL USO DE HERRAMIENTAS DE DISEÑO Y PROGRAMACIÓN VISUALES PARA EL DESARROLLO DE SOFTWARE EDUCATIVO	
2.1 Antecedentes.....	21
2.1.1 Técnicas de programación o desarrollo de aplicaciones.....	21
2.1.1.1 La programación en línea.....	21
2.1.1.2 La programación estructurada.....	22
2.1.1.3 La programación orientada a objetos.....	22
2.1.1.4 La programación orientada a eventos o sucesos.....	23
2.2 <i>Delphi</i> y <i>Visual Basic</i> , opciones para el desarrollo de software educativo.....	24
2.2.1 Características principales de estos lenguajes.....	25
2.3 Manejo de GUI (<i>Interfaz gráfica de usuario</i>).....	26

2.4 Descripción de los controles visuales en <i>Visual Basic</i> y <i>Delphi</i>	27
2.5 Descripción de los sucesos básicos en <i>Visual Basic</i> y <i>Delphi</i>	32

CAPITULO III. APLICACIÓN 1, "EL RELOJ".

3.1 Objetivo del programa.....	33
3.2 Descripción del programa.....	33
3.3 Diagrama de bloques del programa.....	34
3.4 Descripción de cada uno de los bloques del programa.....	35
3.4.1 Pantalla principal.....	35
3.4.1.1 Matemáticas aplicadas para el movimiento de las manecillas del reloj.....	37
3.4.1.2 Descripción de las variables utilizadas en el programa.....	39
3.4.1.3 Descripción de las subrutinas para la pantalla principal.....	40
3.4.2 Pantalla de la opción <i>APRENDE</i>	49
3.4.2.1 Descripción general de la pantalla de esta sección.....	50
3.4.3 Opción <i>BUSCAR LA HORA</i>	55
3.4.4 Sección <i>ESCRIBE LA HORA</i>	56
3.5 Aplicación y explotación del software resultante.....	59
3.5.1 Análisis y discusión de resultados.....	60

CAPITULO IV. APLICACIÓN 2, "EL MAPA DE LA REPÚBLICA MEXICANA"

4.1 Objetivo del programa.....	61
4.2 Descripción del programa.....	62
4.3 Diagrama de bloques del programa.....	63
4.4 Descripción de cada uno de los bloques del programa.....	63
4.4.1 Pantalla principal.....	63
4.4.1.1 Descripción de las variables utilizadas en el programa.....	65
4.4.1.2 Descripción de subrutinas y procedimientos de la pantalla principal.....	66

4.4.2	Pantalla de la opción <i>REPASO</i>	70
4.4.3	Opción <i>LOCALIZAR</i>	75
4.4.4	Opción <i>ESCRIBE LA CAPITAL</i>	80
4.5	Análisis de la implantación del software.....	83
4.5.1	Análisis y discusión de resultados.....	84
CONCLUSIONES DEL PROYECTO		85
ANEXO		
I.	Códigos del programa " <i>EL RELOJ</i> ".....	88
II.	Códigos del programa " <i>EL MAPA DE LA REPÚBLICA MEXICANA</i> ".....	99
III.	Establecimientos visitados para el estudio de mercado.....	107
BIBLIOGRAFÍA		108

INTRODUCCIÓN

Actualmente la computadora cumple con un papel muy importante en el desempeño diario de las actividades en cualquier ámbito, se ha aplicado a ramos tan diversos que prácticamente la podemos ver en cualquier parte. Su uso desde la primera generación, enfocado a la investigación y estrategias militares, se ha extendido gracias al desarrollo que ha alcanzado y a la evolución que día a día va en incremento, desde aquellos grandes monstruos que utilizaban válvulas al vacío hasta las pequeñas microcomputadoras capaces de realizar un gran número de actividades y a una velocidad cada vez mayor, enfocándose incluso a cualquier área como la ciencia, medicina, educación, arte, administración, etc..

Es por eso, incluso, que en la actualidad, muchos estudiantes ven en la computadora una herramienta más para realizar sus actividades escolares, reforzar sus conocimientos adquiridos en clase de una manera más entretenida e inconscientemente, al abrir una enciclopedia, un juego interactivo, un software con esquemas del cuerpo humano, etc. aprenden como en un juego e incluso se alejan cada vez más del mundo tan abstracto que generan otros ambientes de aprendizaje.

Cabe mencionar que en el mercado nacional se maneja una gran cantidad de programas o software educativo, la mayoría de las veces importados del extranjero o hechos en México y que podrían, en determinado momento, cumplir o no con los requisitos necesarios para su implantación en centros educativos.

Algunos de los inconvenientes, son muchas veces: el idioma en el que fueron creados, el precio que, en la mayoría de los casos, es elevado y no esta al alcance de muchos bolsillos, el tipo de actividades de los módulos que no se acoplan a los requerimientos de los educadores, los temas a tratar, los requerimientos de hardware del sistema o software a manejar, etc.

El motivo de estas memorias de desempeño profesional es el demostrar como aplicando algunas herramientas de Matemáticas aplicadas y computación, que son materias interdisciplinarias, se puede resolver este tipo de problemáticas.

Aprovechando el cargo desempeñado en el despacho de *Asesoría Profesional en Programación A. C.*, encargado del desarrollo de software educativo, presento en estas memorias de desempeño profesional las principales causas que propician el uso del desarrollo del software educativo, tomando en cuenta mi experiencia como colaborador directo de un grupo de educadores y realizando un estudio de mercado para generar un panorama más amplio de la situación actual del software educativo en México, aplicando estadística descriptiva para su representación gráfica y así comprender cuáles son los problemas o dificultades a los que se enfrentan los educadores al seleccionar un software determinado que pueda, en determinado momento, servir para llevar a cabo sus actividades docentes y el porqué muchas veces es preferible el diseñar y desarrollar el software a utilizar, además ver cuales son las posibles ventajas o desventajas que acarrea el utilizar este tipo de programas y dar posibles soluciones.

Posteriormente analicé las características de las herramientas de diseño y programación **Visual Basic** de *Microsoft*¹ y **Delphi** de *Borland*² que generan ambientes gráficos de programación con objetos o controles que facilitan el desarrollo del software educativo.

Así mismo, desarrollé dos aplicaciones, con base en el planteamiento propuesto por profesores del colegio La Salle de Seglares, para poder utilizarlos ahí mismo.

Los programas propuestos para su desarrollo son los siguientes:

1. El reloj

Es un programa en el cuál el niño aprende a leer las manecillas del reloj realizando diferentes actividades como: explicaciones de la lectura de las manecillas del reloj, ejercicios de reforzamiento y autoevaluación.

2. El mapa de la República Mexicana

Contiene diferentes secciones para recorrer los estados, repasar sus capitales por medio de ejercicios, y jugar con la división política para localizar estados.

Finalmente analicé los resultados obtenidos al utilizar el software en dicho colegio y elaboré una serie de conclusiones con base en lo anterior.

¹ Compañía Estadounidense dedicada al desarrollo de software de aplicación y sistemas operativos.

² Compañía Estadounidense dedicada al desarrollo de software de aplicación y sistemas operativos.

CAPITULO I. EL SOFTWARE EDUCATIVO COMO UN APOYO DIDÁCTICO EN UNA INSTITUCIÓN EDUCATIVA DEL SECTOR PRIVADO

1.1 Marco teórico

Una de las principales ventajas al utilizar las computadoras como una herramienta de apoyo a la educación, es el aprovechamiento de los grandes adelantos que poco a poco se han venido desarrollando en esta, como el uso de multimedia (*gráficos, audio, animación*), robótica (*uso de pequeñas piezas como motores, poleas, lamparas, etc., que se ensamblan para formar prototipos*) hoy en día es más común ver un gran número de programas educativos y juegos que ayudan al estudiante a sus labores escolares, desafortunadamente, la mayoría de los programas educativos disponibles en nuestro idioma, son software desarrollado originalmente en otras lenguas; inglés principalmente, por lo tanto se trata de programas concebidos para un público muy diferente al hispano, enfocados incluso, a una cultura totalmente diferente a la nuestra.

En nuestro país son muy pocas las compañías que realmente desarrollan este tipo de programas, generalmente el software que se maneja es de origen Estadounidense y simplemente se traduce al idioma español para su distribución en los diferentes almacenes de elementos de computación¹.

VERMIC, S.A. de C.V. es una compañía mexicana que desarrolla y comercializa software educativo en español desde 1990. Sus productos ofrecen una productiva combinación de aprendizaje con entretenimiento, y actualmente son utilizados en miles de hogares y más de 200 instituciones educativas de México y otros países con comunidades hispanas.²

¹ Información obtenida en el estudio de mercado realizado para los temas siguientes.

² Información proporcionada por VERMIC en la dirección electrónica <http://www.vermic.com/>

Algunos de los ejemplos del software que desarrolla esta compañía son los siguientes:

Laberinto de Arnoldo (contiene un juego interactivo que contempla todas las materias de 1° a 6° grados), el cuerpo humano (con esquemas y animaciones), geografía de México (con un recorrido en mapas de la república mexicana), etc.

Otra de las compañías que ofrecen software educativo en México es **ZETA Multimedia** la cual se ha consolidado, según información obtenida en la revista española PCMEDIA, como una de las empresas españolas punteras a la hora de editar software educativo en castellano, publicando programas con temas de interés para el usuario de una forma amena y sencilla³.

Los ejemplos del software realizado por esta compañía, disponibles en México, son los siguientes: El cuerpo humano, El gran atlas del pequeño aventurero, Como funcionan las cosas, El esqueleto 3D, etc.

Más compañías que desarrollan este tipo de software educativo son Educational Software Institute, SuperKids™, Microsoft, etc.

Por lo anterior se deduce que hace falta un impulso para que las pequeñas compañías mexicanas que desarrollan este tipo de programas puedan desarrollarse y de esta forma entrar al mercado nacional sustituyendo el software de origen extranjero para adecuarse a las necesidades que nuestro alumnado demanda.

1.2 Introducción al problema

Al ingresar al despacho de *Asesoría Profesional en Programación A. C.*, estuve a cargo del diseño de bases de datos en *FOXPRO*, el giro de este despacho es diseñar y dar soporte técnico en sistemas de inscripciones, nóminas, control de alumnos en general, así como también creación de materiales de trabajo como libros de texto, blocks de gráficos, etc., para alumnos de las diferentes escuelas que dicho despacho maneja.

³ <http://www.canaldinamic.es/PCMANIA/PC055/RP/ZMETIME/pc055rpzmtme0000.html>

Estas escuelas se mencionan a continuación:

- Colegio Indoamericano
- Colegio Simón Bolívar niñas
- Preescolar y Primaria La Salle Francia
- Preparatoria La Salle del Pedregal
- Universidad del Pedregal

En dichas escuelas, el despacho maneja una planta de profesores asignados para impartir la clase de programación y herramientas, teniendo como objetivo fundamental contribuir en el desarrollo del niño en su razonamiento lógico-matemático para la solución de problemas, por medio de algoritmos de actividades y programación con lenguajes de computadora y además fomentar el manejo de la computadora como una herramienta de trabajo para la elaboración de trabajos de diferentes materias con procesadores de palabras, hojas de cálculo, graficadores, etc., en escuelas secundarias y preparatorias y actualmente la nueva materia de Informática en preparatorias.

Al estar relacionados con las actividades académicas de los diferentes colegios mencionados, se observaron algunas necesidades, por parte de los profesores, *para utilizar la computadora como un apoyo didáctico* y así lograr que los alumnos refuercen la información proporcionada ó crear ambientes de aprendizaje más agradables, principalmente en la *sección primaria de la escuela La Salle de Francia* con alumnos de 2do y 3er grado.

Otro de los objetivos de la escuela era el de *elevar el nivel académico entre los alumnos* implementando la computadora con nuevas técnicas de aprendizaje o mecanización de conceptos, según sea el caso.

1.2.1 Necesidades académicas observadas

- Dificultad de aprendizaje
 - *Memoria: Generalmente en las materias descriptivas como ciencias naturales, ciencias sociales, etc., que manejan una gran cantidad de información y en muchos casos es difícil ó tedioso recordar y que requieren estrategias de enseñanza para facilitarla.*
 - *Ritmo: En el aprendizaje de ciertos temas, en los cuales al ser entendidos por algunos alumnos más rápido que por otros, provocan el defasamiento en ciertos sectores del grupo.*
- Falta de materiales específicos

Se utilizaban materiales concretos como rompecabezas de diferentes mapas, relojes de cartón con manecillas movibles y otros, sin embargo, se pretendía generar actividades en las cuales el alumno interactue con la información aprendida utilizando principios como el de Estimulo - Respuesta.

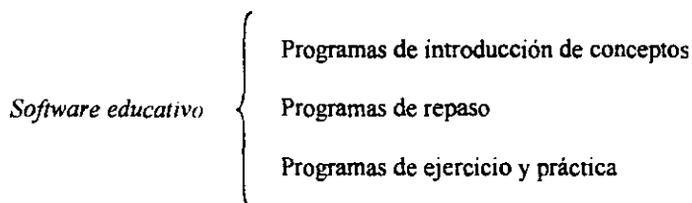
La idea era implementar la computadora como una herramienta de trabajo, de tal forma que el alumno se apoye en ésta de una manera sencilla y dinámica y así mejorar su desempeño académico. Pero para ello era necesario el analizar las diversas formas de explotarla y adecuarla a las necesidades académicas que los profesores requerían en ese momento.

Una de las propuestas, ante estas expectativas de trabajo fue el investigar sobre *software educativo*, aprovechando el recurso del laboratorio con el que cuenta la escuela y que hasta este momento no se había utilizado para este tipo de actividades.

1.3 El software educativo y su clasificación

Software educativo son los programas que desarrollan tareas específicas para el proceso de enseñanza- aprendizaje.

La clasificación sugerida a continuación es planteada de acuerdo al proceso de enseñanza-aprendizaje en el niño el cual consta generalmente en tres etapas *Introducción de conceptos, Repaso y Ejercitación.*⁴



1.3.1 Programas de introducción de conceptos.

Son aquellos programas en los que se supone que el alumno no ha tenido contacto alguno con el tema a tratar, por lo tanto es importante que proporcionen la información de tal forma que pueda comprender el tema en cuestión. Dentro de esta clasificación están los tutoriales y simuladores.

1.3.2 Programas de Repaso

En este tipo de programas queda entendido que el alumno ya conoce el tema, pero aún puede tener algunas dudas sobre ciertos conceptos por lo tanto proporcionan explicaciones y actividades de ejemplificación.

⁴ David P. Ausubel, Psicología educativa. Editorial Trillas, p.66.

Tanto los *programas de introducción de conceptos* como los *programas de repaso* pueden dividirse a su vez en diferentes etapas:

- **Motivación**
Consiste en interesar al alumno con propuestas significativas a su realidad vinculadas al tema en cuestión. En este caso la computadora por sí sola es un medio altamente atrayente para el alumno, ya que como se observó, el niño se interesa al saber que va a usar una computadora y se interesa aún más cuando comienza a ver las imágenes llenas de colores y las actividades que se proponen en esta.
- **Explicación del tema**
Efectuar la explicación del tema considerando los conocimientos previos de los alumnos, para el caso de programas de introducción de conceptos. en el caso de los programas de repaso esta se plantea con el antecedente de la evaluación de la introducción pudiendo corresponder a la retroalimentación del tema.
- **Ejemplificación**
En este paso se aconseja plantear los ejemplos con los errores más frecuentes que por experiencia se sabe que cometen los alumnos.
- **Evaluación**
Se evalúa la comprensión del alumno con una muestra de ejercicios que no pretenden calificar al alumno sino la claridad de la explicación del maestro. Por lo tanto conviene que los ejercicios sean limitados en cantidad.
- **Retroalimentación de conceptos**
Una respuesta por parte del software para aquellos conceptos que no fueron muy bien comprendidos por el alumno y que requieran de un nuevo repaso. Aquí el programa debe indicarle al usuario cuáles fueron sus errores y debe haber la posibilidad de ejercitarlos para una evaluación posterior.

1.3.3 Programas de ejercicio y práctica

Aquí es importante la revisión de conceptos y una serie de actividades para que el alumno repase la información adquirida, mecanice o adquiera una cierta habilidad en el tema.

Las etapas para este tipo de programas son las siguientes:

- **Motivación**
Consiste en interesar al alumno con propuestas significativas a su realidad vinculadas al tema en cuestión.
- **Revisión del concepto**
Revisión del concepto teórico que se va a ejercitar o del algoritmo⁵ que se espera que aplicará el alumno para la solución de los ejercicios.
- **Meta**
Propuesta de metas a lograr por los alumnos, tanto en extensión como en eficiencia esperada para que el ejercicio sea significativo, motivante y trascendente para el mismo alumno. Aquí se propone utilizar marcadores de número de aciertos ó una serie de mensajes o animaciones estimulantes por acierto.
- **Realización del ejercicio**
Al tiempo que el maestro supervisa y detecta las dificultades del promedio del grupo.
- **Evaluación**
Registro de los resultados del ejercicio para confrontarlos con las metas propuestas.

⁵ *Algoritmo* es un conjunto de pasos ordenados para resolver un problema.

También se puede encontrar con que un programa educativo puede contener elementos de las diferentes clasificaciones, es decir, puede ser predominante de ejercicio y práctica, pero contener algunos elementos de introducción de conceptos.

1.4 Ventajas y errores más comunes al implantar el software educativo

Ventajas:

- Suponen una aplicación del principio Estimulo - Respuesta y promueven la ejercitación mediante la repetición.
- La mayoría de estos programas presentan ejercicios aleatorios, es decir, siempre son diferentes por lo que aun que los alumnos tengan la misma actividad, a cada uno se le presentan problemas distintos con el mismo grado de dificultad.
- La presentación de estos programas generalmente es muy atractiva y frecuentemente en forma de juego (Principio lúdico).
- Al profesor se le facilita la tarea de calificar e identificar inmediatamente el tipo de dificultad de la mayoría del grupo y de este modo retroalimentar la actividad con otro tipo de explicaciones.
- El profesor puede plantear las actividades para todos los alumnos promoviendo el trabajo de cada uno a su propio ritmo, respetando de este modo su individualidad y Permitiendo que el alumno o usuario trabajen a su ritmo, proporcionando inmediatamente corrección a sus respuestas.
- Resultan muchas veces atractivos para los alumnos, especialmente para quienes tardan en incorporar ciertos conocimientos o destrezas.
- Descargar al profesor de trabajos mecánicos.

Errores más comunes:

- Abusar de su utilización, mecanizando mentalmente al niño.
- Ignorar la necesidad de una metodología que abarque desde la motivación hasta la retroalimentación, delegado a la computadora labores que corresponden únicamente al maestro.
- No definir a los alumnos metas a alcanzar trayendo como consecuencia su saturación o desmotivación.
- Suponer que la intervención del profesor ya no es necesaria para dar la intencionalidad y retroalimentación a la clase.
- No relacionar su contenido con las actividades en el salón de clases.

1.5 Investigación de mercado en diferentes tiendas y almacenes acerca del software educativo

Una vez planteado el problema y sabiendo cuales son las posibles ventajas y desventajas que acarrea el implantar el software educativo, es necesario conocer la situación actual de este tipo de programas en el mercado, de tal forma que sepamos que software se vende, de que origen es, a quienes esta dirigido, que temas trata, etc., para así poder decidir que es lo que conviene y realizar la toma de decisiones y dar propuestas de solución al problema.

Una definición acerca de la Investigación de Mercados es: *“El procedimiento sistemático de recopilar, registrar y analizar todos los datos relacionados con los problemas de la comercialización de bienes y servicios”*⁶

⁶ Committe on definitions of the American Marketing Association, Marketing Definitions (Chicago: American Marketing Association, 1960).

En esencia el propósito de la investigación de mercados es proporcionar información a los tomadores de decisiones para facilitarles la identificación de una oportunidad o de una situación problemática a fin de que puedan tomar las mejores decisiones posibles cuando enfrenten tales situaciones.⁷

Por tal motivo se investigó en diferentes almacenes y tiendas de autoservicio del Distrito Federal el software que manejan, para sustentar de esta forma las afirmaciones mencionadas anteriormente acerca de este tipo de programas, en los siguientes cuadros se analiza el material encontrado de acuerdo a sus características:

TABLA 1-1. Resultado del software analizado.

Producto	Descripción	Edad	Idioma	País de origen
Atlas del mundo	Sólo Consulta de los países de todo el mundo	9 a 12	Español	España, Argentina, Chile, Méx, USA.
Alge-Blaster	Ayuda al desarrollo del entendimiento de procesos básicos de álgebra.	12 a 18	Inglés	USA
Carlitos vuelta al colegio	Vocabulario, ortografía y lectura, gracias al diccionario interactivo con 1,000 palabras	4 en adelante	Español	ESPAÑA
El cuerpo humano	Consulta de esquemas e imágenes en tercera dimensión	12 a adultos	Español	USA

⁷ Boyd Westfall Staach, Investigación de Mercados, Editorial UTEHA, 1986, p.5.

Producto	Descripción	Edad	Idioma	País de origen
El gran atlas del pequeño aventurero	Consulta de mapas animados para viajar por todo el mundo.	8 a 12	Español	USA
Grammar Gremlins	Ofrece práctica efectiva en las reglas básicas de la gramática	8 a 12	Inglés	USA
Laberinto de Arnoldo	Es un juego que incluye Matemáticas, Español, Historia, Geografía, Civismo, Ciencias Naturales, Educación Artística y Educación Física.	8 a 12	Español	MEXICO
Mapa escolar de México	Sólo consulta con división política	8 a adulto	Español	MEXICO
Math and me	Introduce al niño a la lectura de conceptos de matemáticas	3 a 6	Inglés	USA
Math Blaster	Algebra con operaciones básicas, enunciados, ecuaciones con juegos de destreza.	4 a 16	Inglés, Español	USA
Math Blaster Jr.	Matemáticas desde preescolar hasta el primer grado	3 a 6	Inglés, Español	USA
Math Blaster Plus	Ejercita la destreza de matemáticas básica	6 a 12	Inglés	USA
Read 'n Roll	Contiene historias originales para envolver al alumno en comprensión de lectura	8 a 12	Inglés	USA

Producto	Descripción	Edad	Idioma	País de origen
Reading and me	Prepara a los niños para la lectura, por medio de diferentes juegos.	4 a 7	Inglés	USA
Speed Reader II	Ayuda al lector a incrementar su rapidez de lectura	12 a adultos	Inglés	USA
Spell it!	Ayuda a los estudiantes a manejar más de 1000 palabras que generalmente causan confusión en su manejo.	10 a adultos	Inglés	USA
Super genios ¡Numeruditos!	Matemáticas y resolución de problemas, para desarrollar el razonamiento matemático deductivo.	8 a 15	Inglés, Español	USA
Word attack Plus!	Enseña nuevas palabras y definiciones e incrementa su destreza en lectura y escritura	8 a 12	Inglés	USA
Word Attack!	Perfecciona su destreza de comunicación así como su vocabulario	8 a adultos	Inglés	USA

En este primer cuadro se anotaron las características del software analizado, en más de treinta y tres almacenes y tiendas de software del Distrito Federal y Ciudad Satélite, las direcciones y nombres de los establecimientos visitados, se muestran en el anexo 3.

Tabla 1-2. Clasificación del software de acuerdo a la materia que aborda.

Programa	Area de estudio					
	L	E	M	T	G	A
Atlas del mundo					✓	
Alge-Blaster			✓			
Carlitos vuelta al colegio	✓	✓				
El cuerpo humano						✓
El gran atlas del pequeño aventurero					✓	
Grammar Gremlins	✓	✓				
Laberinto de Arnoldo			✓		✓	✓
Mapa escolar de México					✓	
Math and me	✓			✓		
Math Blaster			✓			
Math Blaster Jr.			✓	✓		
Math Blaster Plus			✓			
Read 'n Roll	✓	✓				
Reading and me	✓	✓		✓		
Speed Reader II	✓					
Spell it!	✓	✓				
Super genios ¡Numeruditos!			✓			
Word attack Plus!	✓	✓				
Word Attack!	✓	✓				

L Lectura
E Escritura
M Matemáticas

T Temprano aprendizaje
G Geografía
A Anatomía

Al observar los resultados es notable que la mayor parte de los programas son enfocados a la lectura y escritura, en segundo término están las matemáticas y por último geografía, anatomía y otros temas.

Tabla 1-3. Clasificación del software de acuerdo a la materia que aborda.

Area de estudio	Tratan el tema	No tratan el tema	Total
Lectura	9	10	19
Escritura	7	12	19
Matemáticas	6	13	19
Temprano aprendizaje	3	16	19
Goografía	4	15	19
Anatomía	2	17	19

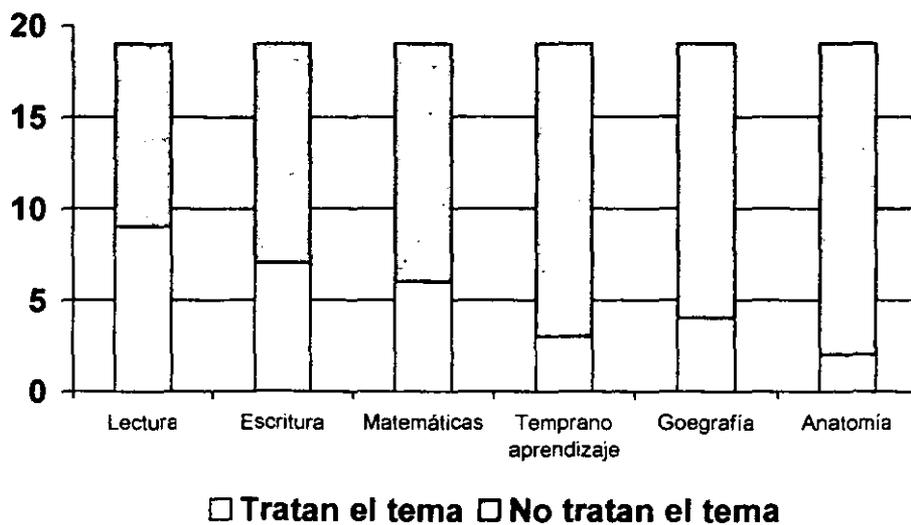


Tabla 1-4. Porcentajes del software analizado de acuerdo al idioma que maneja.

Idiomas	Programas evaluados	Porcentaje
<i>Español</i>	6	31.6
<i>Inglés</i>	10	52.6
<i>Ambos</i>	3	15.8
Total	19	100

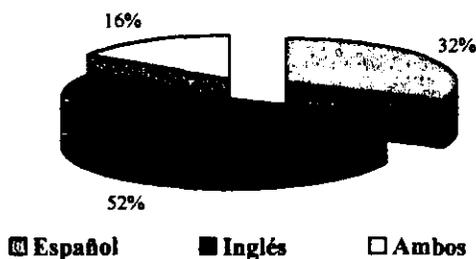
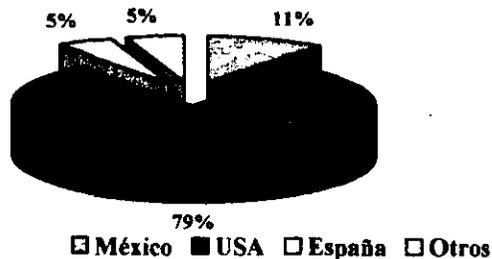


Tabla 1-5. Porcentajes de acuerdo al país de origen

País de origen	Programas evaluados	Porcentaje
<i>México</i>	2	10.5
<i>USA</i>	15	78.9
<i>España</i>	1	5.3
<i>Otros</i>	1	5.3
Total	19	100



1.5.1 Análisis de la investigación de mercado

De acuerdo a la información obtenida por los diferentes almacenes visitados se llegó a las siguientes conclusiones:

- Los programas en su mayoría tratan temas de Lectura, escritura y matemáticas, aunque también hay software que trata temas diferentes.
- La mayoría de los programas son de consulta
- La mayor parte de los programas son enfocados a los niveles de primaria de primero a sexto grados, en segundo lugar temas de secundaria y preparatoria.
- El idioma que predomina es el inglés y después el español.
- El país de origen predominante es Estados Unidos de Norteamérica.

Con los resultados anteriores se llegó a las siguientes conclusiones:

- a) Se requiere de mayor número de programas que permitan ejercitar al alumno todos los conceptos aprendidos en forma de juegos, ejercicios de autoevaluación, etc. para evitar un poco solamente consultar la información.
- b) Se requiere adecuar programas a las necesidades académicas planteadas por los profesores de los diferentes niveles ya que muchos de estos programas se concretan a actividades generales, ignorando aquellas que en determinado momento pueden ser las que el profesor necesita.

- c) Por su procedencia muchos programas se distribuyen en otros idiomas, dificultando aún más el aprovechamiento en los alumnos y muchas veces enfocados a otras culturas totalmente diferentes a las que esta viviendo el alumno.

Por lo anterior, se llegó a la conclusión de que es importante el diseñar el software que el alumno utilizará, basándose en las necesidades académicas del profesorado y diseñando una serie de actividades, propuestas por los profesores para conducir a los alumnos hacia los objetivos planteados.

CAPITULO II. EL USO DE HERRAMIENTAS DE DISEÑO Y PROGRAMACIÓN VISUALES PARA EL DESARROLLO DE SOFTWARE EDUCATIVO

2.1 Antecedentes

Una vez que fue decidido el diseñar software educativo para satisfacer las necesidades académicas dentro de la institución mencionada, se llevó a cabo la investigación de herramientas de diseño y programación para seleccionar entre las diferentes propuestas, aquella que mejor se adapte a las características necesarias para desarrollar el software pensado. De acuerdo a la investigación realizada se llegó a lo siguiente:

2.1.1 Técnicas de programación o desarrollo de aplicaciones.

2.1.1.1 La programación en Línea

Es la primer técnica implementada para el desarrollo de programas, consiste en escribir un código en forma de lista general de tal forma que las instrucciones se van ejecutando renglón por renglón y llevan un orden secuencial, esto quiere decir que hasta que no se lleve a cabo ó ejecute la primer instrucción, no se realizara la del siguiente renglón, etc..

Ver fig. 2-1

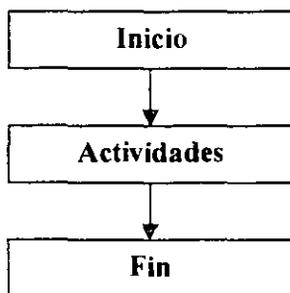


Fig. 2-1 Representación de la programación en línea.

2.1.1.2 La programación estructurada

Surge como una solución para resolver las dificultades que ocasiona un código en lista, esta consiste en dividir el programa general en bloques o módulos, los cuáles son independientes uno del otro y se ejecutan desde cualquier parte del programa en la que sean llamados de tal forma que al generarse un error es más sencillo localizarlo dentro de un bloque específico y depurarlo, sin necesidad de revisar la lista general de código. En este tipo de programación también se puede llamar un bloque desde otro bloque independiente, hay una comunicación desde cualquier parte del programa. Ver fig. 2-2.

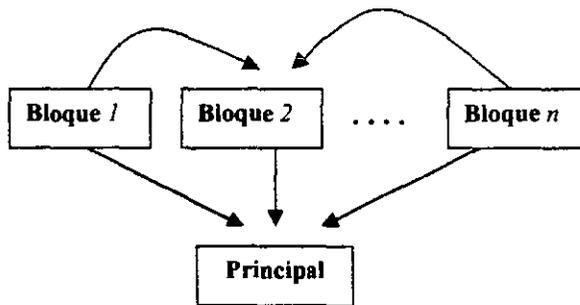


Fig. 2-2 Representación de la programación estructurada

2.1.1.3 La programación orientada a objetos

Es una de las técnicas más modernas de desarrollo de aplicaciones, consiste en desarrollar entidades de software llamados objetos. Estos incluyen *campos* que ayudan a definir el objeto, *procedimientos* que definen la manera en que se pueden modificar los objetos y una *identidad* que los distinguen de algún otro objeto. La ventaja es que al usarlos en diferentes jerarquías o niveles se pueden heredar desde un nivel inferior las características de un objeto que este en niveles superiores y así el programador no tendrá que escribir nuevamente todo el código para cada objeto si no simplemente maneja las diferencias. Este tipo de técnicas suele ser un poco abstracta debido a los términos que emplea.

Ver fig. 2-3.

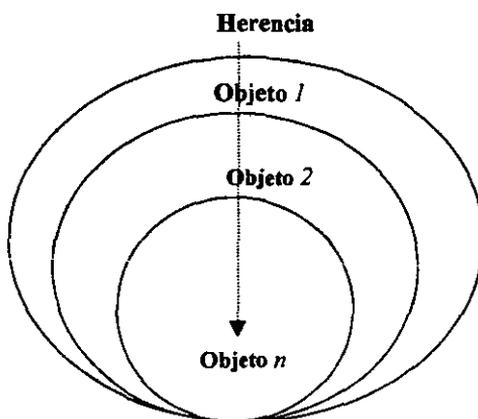


Fig. 2-3 Representación de la programación orientada a objetos.

2.1.1.4 La programación orientada a eventos ó sucesos

Es otra técnica moderna de desarrollo enfocada a ambientes ó sistemas operativos gráficos como Windows 95, consiste en asignar códigos a componentes visuales, entendiendo por componente visual todo aquello que tiene forma y características propias como color, tamaño, grosor, etc., que lo distinguen de los demás componentes visuales. Un componente visual se puede manipular o manejar en forma aislada e independiente sin afectar a los demás componentes visuales además se emplean para formar el GUI (*Interfaz gráfica de usuario*) que permite la comunicación entre el usuario y la computadora. El GUI es todo aquello que el usuario ve a través de la pantalla con lo que se comunica, un componente visual puede o no tener asociado un código y este código sólo se ejecutara o llevara a cabo hasta que el usuario realice algún evento o actividad que altere al componente visual, por ejemplo hacer un click sobre un dibujo para que cambie de color, presionar la tecla ENTER para que se guarde un archivo, hacer doble click sobre un icono para que se abra una aplicación, etcétera. Todas estas actividades que realiza el usuario, click, teclear, dobleclick para manipular a los componentes visuales son llamadas *eventos o sucesos*. Ver fig. 2-4.

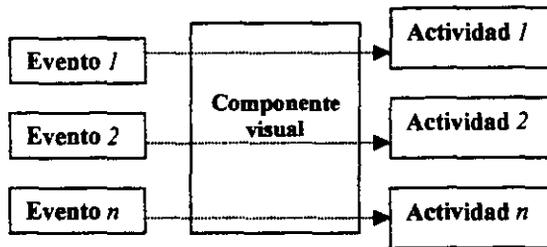


Fig. 2-4 Representación de la programación orientada a eventos o sucesos

2.2 Delphi y Visual Basic, opciones para el desarrollo de software educativo.

Por lo descrito anteriormente se llegó a la conclusión de utilizar la *programación orientada a eventos o sucesos* debido a que ofrece grandes ventajas como el diseño de GUI y características que exigen los programas de software educativo, para la creación de sus pantallas y presentaciones de gráficos y animaciones, esto no quiere decir que la programación orientada a objetos, programación estructurada o programación lineal sean menos eficaces o no reúnan las características necesarias para la creación de software educativo, todas son opciones de programación e incluso se pueden complementar para la obtención de mejores resultados.

Al hablar de programación orientada a eventos o sucesos podemos mencionar a Visual Basic y Delphi como herramientas de diseño, ya que desde hace algunos años han venido a revolucionar la forma de programación para el desarrollo de aplicaciones, aprovechando todas las características de Windows, por su facilidad de manejo, proporcionando un conjunto de herramientas que facilitan el diseño de programas, empleando los objetos y controles manejados en Windows:

- Cuadros de texto
- Etiquetas
- cajas de lista
- ventanas, imágenes

CAPITULO II. EL USO DE HERRAMIENTAS DE DISEÑO Y PROGRAMACIÓN VISUALES PARA EL DESARROLLO DE SOFTWARE EDUCATIVO

Reduciendo el código que hubiera sido utilizado para crear el tipo de pantallas, que en determinado momento, se hubieran realizado con otro tipo de lenguaje o sistema.

*Visual Basic es un sistema de desarrollo diseñado especialmente para crear aplicaciones gráficas de una forma rápida y sencilla*¹

*Visual Basic es una herramienta sencilla pero potente para desarrollar aplicaciones Windows en Basic.*²

2.2.1 Características principales de estos lenguajes

Este tipo de lenguajes han implementado en los programadores, una serie de técnicas de uso del lenguaje de programación que reduce cada vez más el porcentaje de error en la lógica del programa, una de ellas es *la modulación* que consiste en dividir cada vez más los programas en secciones más sencillas e independientes una de otra, por lo tanto el manejo de módulos³ en las diferentes secciones del código permite un mejor control dentro de la estructura general de un programa, ya que a la hora de depurar un programa resulta más sencillo identificar cada uno de los diferentes procesos organizados en su propio módulo a revisar una lista general de códigos.

¹ Ceballos Fco. Manuel, *Microsoft Visual Basic aplicaciones para windows*, Addison-Wesley Iberoamericana, 1993, p 3.

² Cornell Gary, *Manual de Visual Basic 3 para windows*, Mc-Graw-Hill, 1994, p xxi.

³ *Un módulo es una parte relativamente pequeña y manejable de código de programación*
Cornell Gary, *Manual de Visual Basic 3 para Windows*, Mc-Graw-Hill, 1994, p 5.

2.3 Manejo de GUI (Interfaz gráfica de usuario)

Como se mencionó anteriormente el GUI o Interfaz gráfica de usuario es todo aquello que se ve a través de la pantalla y por medio del cual el usuario se comunica con la computadora para que esta realice las actividades asignadas.

Como ejemplo de Interfaz gráfica de usuario podemos mencionar The farm (La granja), realizado por *Humongous Entertainment*, un juego en inglés que ayuda al niño a familiarizarse con las herramientas de trabajo relacionadas con las actividades de los granjeros en Estados Unidos, aquí el niño simplemente se comunica con la computadora señalando las imágenes que reconoce para que la computadora decida si es correcta o no la opción seleccionada. La comunicación entre el niño y la computadora se lleva a cabo simplemente utilizando los elementos gráficos dentro de la pantalla que se muestra a continuación.

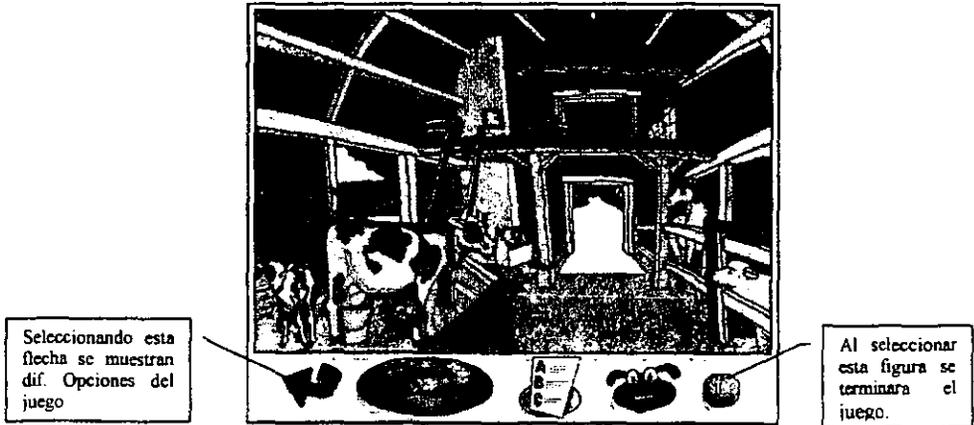


Figura 2-5 Ejemplo de Interfaz gráfica de usuario.

Utilizando Visual Basic y Delphi es posible realizar este tipo de GUI, ya que estos sistemas implementan una serie de componentes visuales para el desarrollo de aplicaciones en Windows.

Los componentes visuales son aquellos elementos con los cuales se forma el GUI y pueden ser manipulados en forma aislada e independiente⁴

2.4 Descripción de controles visuales en Visual Basic y Delphi.

Básicamente manejan dos tipos de componentes *formas* y *controles*.

- **Las formas** son el fondo de los programas, es el área de trabajo sobre la cual se insertan o colocan los diferentes controles con los cuales se creará el Interfaz gráfico de usuario (GUI).

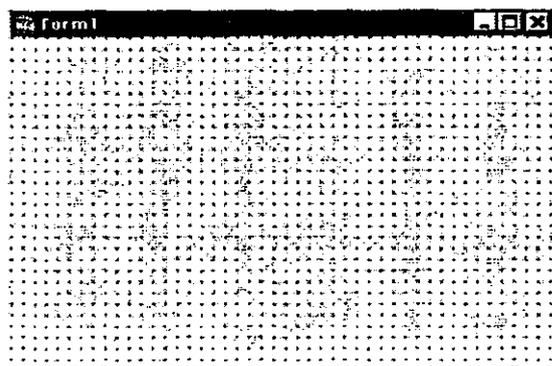


Figura 2-6 Forma o ventana utilizada en Delphi y Visual Basic.

Esta forma tiene *características o propiedades* que la distinguen de los demás objetos utilizados para el diseño del GUI algunas de ellas son las siguientes:

Propiedad	Utilidad
Name	Asigna un nombre a la forma para diferenciarla de los demás controles y formas y para que pueda ser programada ya que de lo contrario no se podría.

⁴ Maya Martinez Joel, *Delphi I*, Editorial Libra, 1998, p50

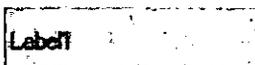
**CAPITULO II. EL USO DE HERRAMIENTAS DE DISEÑO Y PROGRAMACIÓN VISUALES PARA EL
DESARROLLO DE SOFTWARE EDUCATIVO**

Propiedad	Utilidad
Caption	<i>Asigna un letrero o mensaje a la forma sobre su barra de titulos.</i>
BorderStyle	<i>Tipo de borde de la forma</i>
BackColor en V.B. Color en Delphi	<i>Color de fondo para la forma.</i>

- **Los controles** son componentes visuales que se colocan sobre la ventana o forma y, de acuerdo al que se trate, realizan diferentes actividades .

A continuación se mencionaran solamente los más importantes ya que Visual Basic Y Delphi poseen un gran número de estos controles visuales.

- **Label (Etiqueta)**



Permite colocar letreros sobre la forma pero no pueden ser modificados mientras el programa se esté ejecutando.

Propiedades importantes:

Propiedad	Utilidad
Caption	<i>Para asignar el letrero</i>
Name	<i>Asigna un nombre al control</i>

- **Text Box (Caja de texto)**



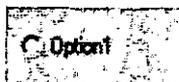
Es una caja sobre la cual se puede introducir texto, borrar y modificar el existente, mientras el programa se está ejecutando.

**CAPITULO II. EL USO DE HERRAMIENTAS DE DISEÑO Y PROGRAMACIÓN VISUALES PARA EL
DESARROLLO DE SOFTWARE EDUCATIVO**

Propiedades importantes:

Propiedad	Utilidad
Text	<i>Para asignar el letrero</i>
Name	<i>Asigna un nombre al control</i>

• **Option Box (Cuadro de opciones)**



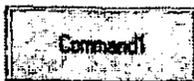
Son controles que se pueden manejar en grupo y permiten seleccionar una opción entre varias opciones. Cuando el control es seleccionado se ilumina el centro de color negro y los demás cuadros de opciones del mismo grupo se desactivaran. Una de las principales características de este componente es que al seleccionar uno de un grupo, los restantes se desactivaran.

Propiedades importantes:

Propiedad	Utilidad
Caption	<i>Para asignar el letrero</i>
Name	<i>Asigna un nombre al control</i>
Value	<i>Determina si el control esta seleccionado o no con 2 posibles valores True (cierto) o false (falso).</i>

**CAPTULO II. EL USO DE HERRAMIENTAS DE DISEÑO Y PROGRAMACIÓN VISUALES PARA EL
DESARROLLO DE SOFTWARE EDUCATIVO**

- **Command (Botón de comando)**



Es un control al cual se le asignan códigos para que se ejecuten las actividades deseadas dentro de la aplicación.

Propiedades importantes:

Propiedad	Utilidad
Caption	<i>Para asignar el letrero sobre el botón</i>
Name	<i>Asigna un nombre al control</i>

- **Picture Box (Caja de pintura)**



Es un objeto muy importante que posee características que permiten que sea utilizado como un área de trabajo con escalas, para desarrollar aplicaciones gráficas utilizando métodos como Line (trazar líneas), circle(trazar círculos), etc.

Propiedades importantes:

Propiedad	Utilidad
ScaleMode	<i>Define la escala para trabajar en esta área y pueden ser centímetros, milímetros, pulgadas, etc.</i>
Name	<i>Asigna un nombre al control</i>

**CAPITULO II. EL USO DE HERRAMIENTAS DE DISEÑO Y PROGRAMACIÓN VISUALES PARA EL
DESARROLLO DE SOFTWARE EDUCATIVO**

- **Frame (Cuadro de marco)**



El *frame* es un control que permite agrupar controles dentro de él para un manejo más sencillo.

Propiedades importantes:

Propiedad	Utilidad
Caption	<i>Para asignar el letrero</i>
Name	<i>Asigna un nombre al control</i>

- **Timer (Temporizador)**



Este control se encarga de ejecutar un código asignado empleando intervalos de tiempo, de tal forma que al estar ejecutándose un programa se podrá mandar actividades independientes cada intervalo de tiempo establecido.

Propiedades importantes:

Propiedad	Utilidad
Interval	<i>Determina el intervalo de tiempo para ejecutar los códigos. Se maneja en milisegundos.</i>
Name	<i>Asigna un nombre al control</i>

2.5 Descripción de los sucesos básicos en Visual Basic y Delphi

Al utilizar este tipo de herramientas de programación, es importante mencionar que cada componente que forma parte del GUI (Interfaz gráfico de usuario) puede contribuir al programa con una serie de instrucciones o actividades asignadas al ser manipulado por el usuario, esto quiere decir que cada componente permanecerá estático o sin responder hasta que el usuario haga un click con el apuntador del mouse sobre alguno de ellos, haga doble click, presione alguna tecla, escriba información, etc. cualquiera de estas actividades realizadas por el usuario provocaran que los controles respondan a los estímulos mostrando una pantalla nueva, cambiando el color de alguna área del gráfico, mandando un mensaje, provocando una animación o algún sonido, etc. en fin se pueden asignar un gran numero de actividades a realizar dependiendo las necesidades del programa, *con lo anterior se puede decir que un suceso o evento es la actividad provocada por el usuario al manipular los componentes del GUI*, en visual Basic y Delphi existen un gran número de sucesos o eventos que se pueden emplear para el manejo del programa los más utilizados o comunes son los siguientes:

Evento o suceso	Descripción
Click	Hacer un click con el apuntador del mouse sobre algún componente.
DblClick	Hacer un doble click con el apuntador del mouse sobre algún componente
Mouse Move	Mover el apuntador del mouse sobre algún componente
Keypress	Presionar alguna tecla
Create (Delphi)	Se lleva a cabo cuando se crea la forma en el momento de ejecutar el programa.
Activate (Visual Basic)	Se lleva a cabo cuando aparece la forma en la pantalla al ejecutarse el programa.
Load (Visual Basic)	Ocurre antes del activate, mientras se carga el formulario en la memoria de la computadora.
GotFocus	Cuando se selecciona algún objeto del GUI.

CAPITULO III. APLICACIÓN 1, "EL RELOJ"

Aprovechando las características de las computadoras que pueden crear entornos interactivos de comunicación entre un programa y el usuario, logrando que exista una comunicación entre el usuario y el programa a través de juegos y actividades diferentes y debido al problema observado en la primaria La Salle Francia, que acarrea en el alumno de 2do. grado la lectura de las manecillas del reloj y la falta de material didáctico para este tipo de actividades, se pensó en diseñar este programa para que por medio de diversas actividades como el manejo de las manecillas, la escritura de la hora leída en la pantalla del reloj, lectura acerca de los principios básicos de la lectura del reloj, el alumno ejercite en forma individual las actividades que se le dificultan para aprender a leer la hora.

3.1 Objetivo del programa

Apoyar al niño en la lectura de las manecillas del reloj.

3.2 Descripción del programa

Por la forma en que trabaja este software, lo clasificamos como un programa de ejercicio y práctica, ya que proporciona explicaciones y actividades de ejemplificación, orientados a alumnos que ya conocen el tema y además incluye prácticas para que el alumno refuerce la información adquirida y adquiera una cierta habilidad en la lectura de las manecillas del reloj.

El programa del reloj se divide en cuatro diferentes secciones:

1) Aprender: Es la sección, por medio de la cual, el alumno puede reafirmar el tema y revisar la dudas que pudiera tener para aprender a leer las manecillas del reloj atendiendo a la explicación proporcionada y realizando las actividades de autoevaluación sugeridas.

2) **Buscar la hora:** En esta sección, el usuario puede practicar moviendo las manecillas del reloj a la hora indicada por el programa, como estímulo para el usuario se lleva un conteo de aciertos obtenidos y errores cometidos.

3) **Escribir la hora:** Aquí el usuario tendrá que escribir la hora marcada por el reloj, también se estimula al usuario contando aciertos y errores.

4) **Salida:** Al seleccionar esta opción el alumno puede abandonar el programa y terminar con sus actividades.

3.3 Diagrama de bloques del programa

Para una mejor comprensión de las secciones en que esta dividido el programa del reloj se muestra el siguiente diagrama de bloques.

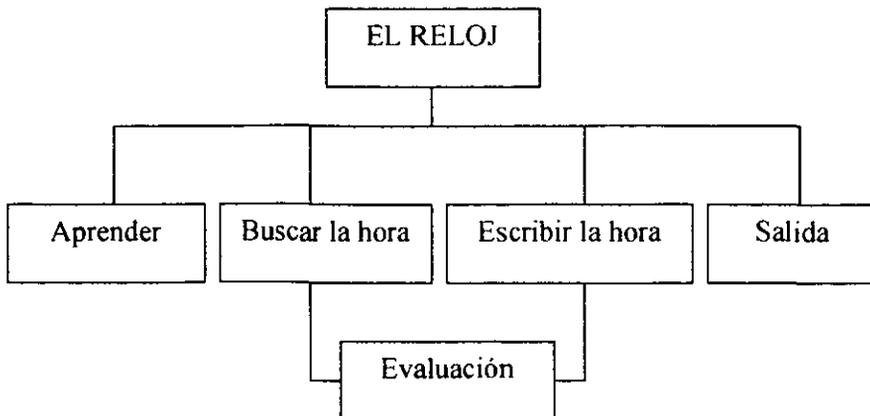


Figura 3-1 Diagrama de bloques

Las secciones de *Buscar la hora* y *Escribir la hora*, nos muestran al final de la ejercitación un cuadro de resultados totales.

3.4 Descripción de cada uno de los bloques del programa

Debido a la extensión del programa se consideró el analizar y describir en forma aislada cada uno de sus bloques, mostrando GUI (Interfaz gráfico de usuario), tabla de componentes visuales mencionando sus características principales, y códigos asignados a los sucesos o eventos especificados en la tabla.

3.4.1 Pantalla Principal

Es mostrada al inicio del programa y por medio de esta, se puede tener acceso a cada uno de los bloques del programa, seleccionando con el indicador del Mouse la opción deseada y haciendo un Click sobre ella.

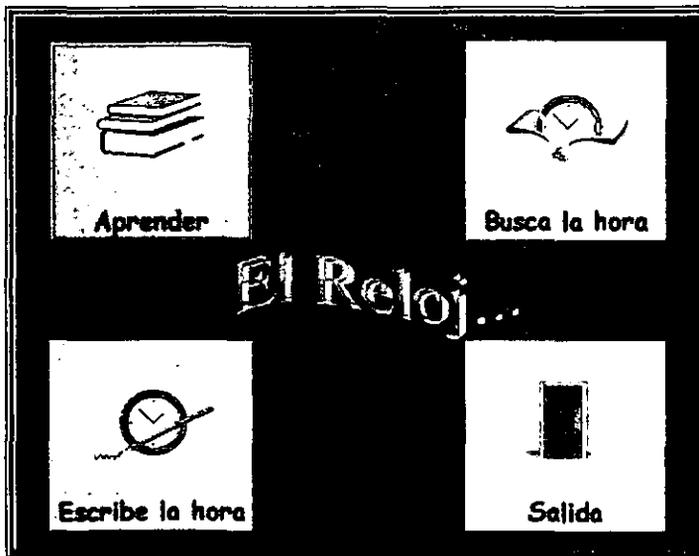


Figura 3-2. Pantalla principal

Descripción general de la pantalla principal

- a) *Requerimiento funcional:* Pantalla para elegir las opciones Aprender, Escribir la hora, Buscar la hora y Salida.
- b) *Entrada:* Se requiere para que el usuario seleccione a donde desea ingresar.
- c) *Proceso:* El usuario, al seleccionar la opción, lo llevara a la actividad elegida.
- d) *Salida:* Pantalla para elegir opción.

Tabla de Objetos y sucesos utilizados para esta sección

Objeto	Nombre	Letrero ó Mensaje	Suceso
Sspanel	Panel2	NINGUNO	NINGUNO
Image1	Imagen1	NINGUNO	NINGUNO
Command1	BOTON(1)	Aprender	Click
Command2	BOTON(2)	Escribe la hora	Click
Command3	BOTON(3)	Busca la hora	Click
Command4	BOTON(4)	Salida	Click

- El objeto llamado Panel2 es el recuadro sobre el que se colocaron los objetos con los que se maneja esta opción.
- La imagen1 es la que se encuentra en el centro del recuadro y contiene el letrero del reloj.
- Para programar los botones de comando se utilizó la técnica de *arreglos de control* que consiste en agrupar los objetos asignadoles el mismo nombre para que compartan

el mismo código y su programación sea más ágil, ya que con un solo código se manejan todos los botones y se diferencian uno del otro utilizando un número de índice entre paréntesis.

3.4.1.1 Matemáticas aplicadas para el movimiento de las manecillas del reloj

Para poder representar el funcionamiento de un reloj analógico verdadero en la computadora fue necesario emplear las matemáticas ya que se requería de dos manecillas que al girar constantemente sobre la carátula del reloj marcaran las horas y los minutos y además que tuvieran la posibilidad de manipularse al azar o en forma dirigida para realizar las actividades que requiere el apoyo en la lectura de las manecillas del reloj.

Por tal motivo se buscó la forma de representar en forma gráfica este tipo de componentes y se pensó en utilizar vectores en el plano cartesiano ya que algunas cantidades en las matemáticas y otras ciencias, tales como el área, el volumen, la longitud del arco, la temperatura y el tiempo, solo tienen magnitud y se pueden caracterizar completamente con un solo número real (*con una unidad de medida apropiada como cm^2 , cm^3 , $^{\circ}C$, etc.*).

Una cantidad de este tipo es una **cantidad escalar** y el número real correspondiente se llama **escalar**.

Conceptos como el de velocidad o fuerza poseen tanto magnitud como dirección y a menudo se representan por flechas o segmentos dirigidos, es decir, segmentos en los que se señala un sentido y representan una dirección. A un segmento dirigido también se le llama **Vector**¹.

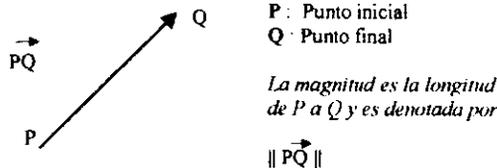


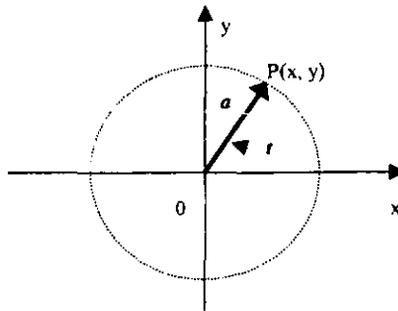
Figura 3-3 Representación de un vector.

¹ W.Swokowsky Earl, *Cálculo con geometría analítica*. Grupo editorial Iberoamérica, p.678

El movimiento de las manecillas del reloj se puede representar en el plano cartesiano como un cuerpo que describe una trayectoria circular en donde un vector esta girando a partir del origen del plano su trayectoria $P(x, y)$ al tiempo t esta dada por:

Magnitud : Longitud del vector, representada por a .

Dirección : Angulo de inclinación del vector, con respecto al eje x , representado en grados por t .



Componente X

$$X = a \cos t$$

Componente Y

$$Y = a \sin t$$

Figura 3-4 Representacion del movimiento de una manecilla del reloj, por medio de un vector.

Para encontrar cada componente del vector se emplearon las fórmulas especificadas en la figura 3-4 y se hace la conversión de grados a radianes ya que esta es la escala en que trabaja la computadora.

Considerando que 180° equivalen a π radianes

Tenemos entonces que $1 \text{ radian} = \frac{180^\circ}{3.1416} \cong 57.29^\circ$

Es por eso necesario dividir la cantidad en grados sobre 57.3 para su conversión a radianes.

Se utilizaron varios vectores para trazar las manecillas del reloj, ya que éstas tienen una forma especial y no sólo se representan con una línea sencilla.

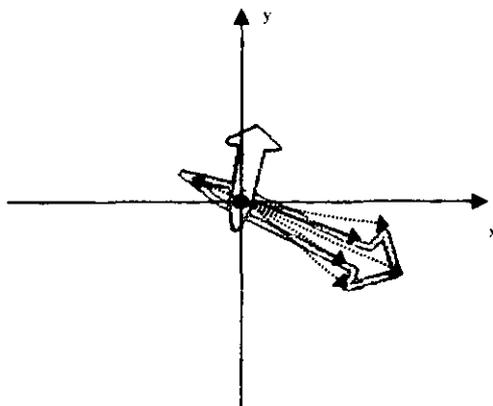


Figura 3-5 Representación de una manecilla del reloj, por medio de varios vectores.

Para trazar las manecillas, solo se dibujaron líneas uniendo cada uno de los puntos finales de los vectores, como se muestra en la figura anterior (*Figura 3-5*).

3.4.1.2 Descripción de las variables utilizadas en el programa

AnguloCh, AnguloG: Variables numéricas de tipo entero que controlan el ángulo de inclinación en la pantalla de las manecillas chica y grande respectivamente.

Minuto, Hora: Variables numéricas de tipo entero que se encargan de almacenar el minuto y la hora marcados por el reloj.

MinutoA, HoraA: Variables numéricas de tipo entero que almacenan el minuto y la hora seleccionados al azar por el reloj para la sección de preguntas.

Respuesta: Variable numérica de tipo entero que permite comparar el valor que responde el usuario con el que asigna la máquina al azar para poder calificar el resultado.

Aciertos: Variable numérica de tipo entero que almacena el número de aciertos en las actividades realizadas.

OPCIÓN: Variable numérica de tipo entero que determina que actividad se va a realizar de acuerdo a la selección del usuario.

Contador, k: Variables numéricas de tipo entero que controlan las iteraciones en los circuitos o ciclos utilizados en el programa.

CONTINUA, DESACTIVAR Variables de tipo booleano que determinan si se continua con el programa o se desactiva la opción cuando su valor se encuentre en TRUE (Verdadero).

TEORIA(1 To 12) Es un arreglo de variables que permite almacenar la información que explica al alumno como leer las manecillas del reloj.

3.4.1.3 Descripción de subrutinas para la pantalla principal

Se utilizaron subrutinas ² auxiliares para agilizar los códigos y manejar las algunas actividades en diferentes partes del proyecto.

A continuación se explican cada una de ellas:

Subrutina Comp_Coloca_Manecillas

Para crear esta subrutina, era necesario el representar las manecillas del reloj en una forma gráfica y la más idónea para esto fue el utilizar vectores desplazándolos en el plano cartesiano

La subrutina se encarga de seleccionar una hora y un minuto al azar y posteriormente lo marca con las manecillas y lo indica en las cajas de texto correspondientes.

Para marcar la hora con las manecillas se auxilia de la subrutina *Manecillas*.

Nota :

Considerando que en el plano cartesiano los grados máximos a representar son 360° se hizo la conversión a minutos tomando en cuenta que el minuto máximo es 60 utilizando regla de 3.

² Subrutinas Subprogramas que ejecutan un proceso específico
Luis Joyanes Aguilar, **Fundamentos de programación**, Mc-Graw-Hill, p. 173

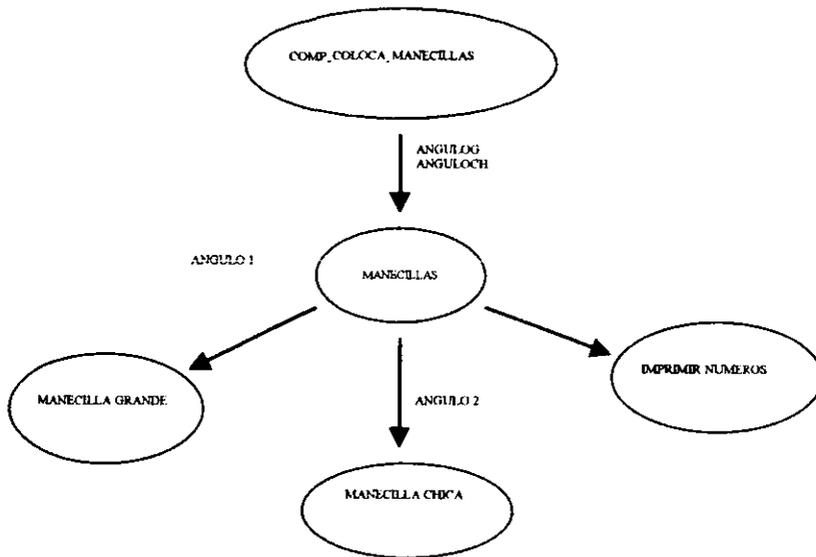


Figura 3-5 Diagrama de flujo de datos de la subrutina Comp_Coloca_Manecillas.

Subrutina Manecillas

Se encarga de llamar las subrutinas que dibujan cada una de las manecillas del reloj, indicando a que grados se inclinara cada una de ellas.

Subrutina Imprime_ NÚMEROS

Esta subrutina, cada que es llamada, se encarga de imprimir sobre la carátula del reloj los números marcando las horas y los minutos, para la precisión de los números impresos se empleó la técnica de los vectores mencionada en los códigos anteriores.

Subrutina Mostrar_Objetos

Se encarga de seleccionar los objetos que aparecerán de acuerdo a la opción seleccionada, la propiedad que se utilizó en esta subrutina es *VISIBLE*, la cual al tener el valor de *True* muestra el objeto y cuando esta en *False* lo oculta.

Subrutina Manecilla_Chica

Dibuja la manecilla chica del reloj (*Horario*), por medio de vectores inclinándola sobre la carátula a los grados que indica la variable *AnguloCh*.

Subrutina Manecilla_Grande

Dibuja la manecilla chica del reloj (*Minutero*), por medio de vectores inclinándola sobre la carátula a los grados que indica la variable *AnguloG*.

Esta pantalla se maneja como el menú principal ya que al hacer click en cualquiera de los cuatro botones de comando se evalúa inmediatamente el índice correspondiente para saber que actividad realizará el programa:

- a) *Cuando el índice sea 1*, se activará la música de bienvenida, y se habilitará el temporizador que se encargará de mover las manecillas del reloj mientras se manda el texto asignado.
- b) *Cuando el índice sea 2*, se colocarán las manecillas en la hora seleccionada al azar y se indicará al usuario que escriba la hora señalada por el reloj.
- c) *Cuando el índice sea 3*, se colocarán las manecillas en la hora seleccionada al azar y se indicará al usuario que coloque las manecillas en la hora especificada.

Dicho código se muestra a continuación en la **figura 3-6**.

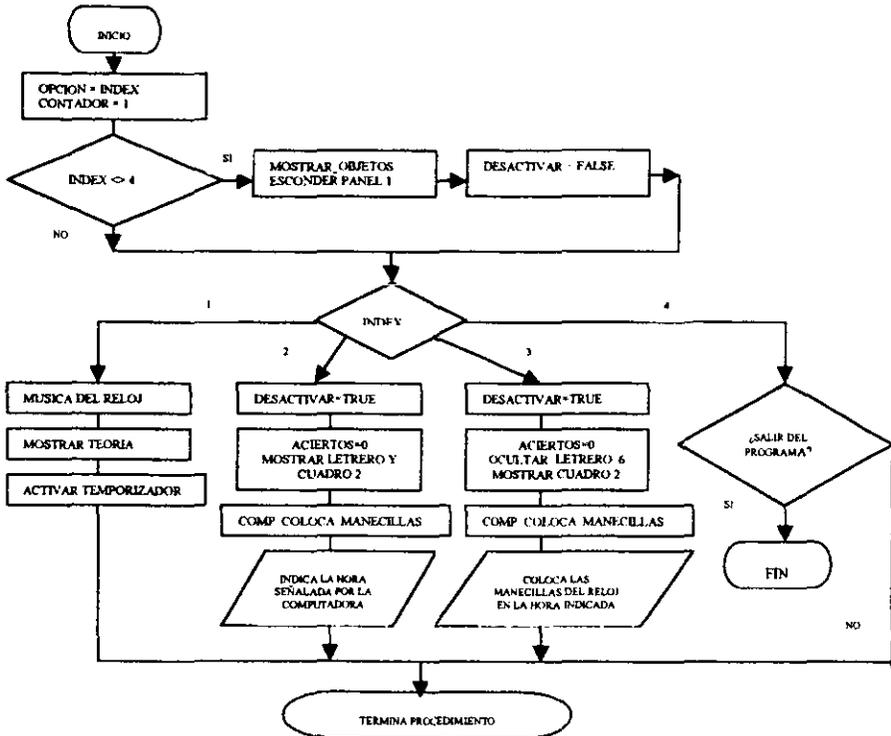


Figura 3-6 Diagrama de flujo del procedimiento asignado al evento CLICK del arreglo de botones (Referencia tabla de la pág. 36).

Subrutina Música

Se encarga de asignar al reproductor de multimedia el tipo de archivo que manejará, el cuál variará de acuerdo a la opción que lo necesite, el nombre del archivo, e indicara que debe abrir y ejecutar la música, si es que la computadora en la que se ejecute cuenta tarjeta digitalizadora de audio.

Hablando ahora del Timer o temporizador, como se explicó en el capítulo anterior, éste es un objeto que permite ejecutar códigos a un intervalo de tiempo determinado, medido en milisegundos, sin importar que otras actividades se estén llevando a cabo, además aprovechando esta característica, el usuario puede estar escribiendo la hora, leyendo la

información contenida en el programa, etc., en fin realizando cualquier otra actividad, mientras el reloj esté caminando sus manecillas como si fuera verdadero.

Nota: El temporizador se activa al asignar a su propiedad ENABLED el valor de TRUE (Verdadero) y se desactiva al asignarle FALSE (Falso) a dicha propiedad.

A continuación muestra en el siguiente procedimiento (**Timer1**), que cada 200 milisegundos llama a la subrutina que hace que el minuterero se mueva.

El tiempo de reacción del Timer es asignado en tiempo de diseño, es decir mientras se diseña el GUI del proyecto directamente se asigna utilizando la propiedad **INTERVAL** del temporizador o *Timer*.

El procedimiento *Timer2* se encarga de mover el MINUTER hasta un tiempo determinado al azar por la computadora y cuando llega al tiempo especificado hace que se desactive para que deje de mover el minuterero, esta actividad se lleva a cabo para la opción de escribir la hora ya que la máquina decide qué hora debe escribir correctamente el usuario.

El procedimiento *Timer3* se encarga de mover el HORARIO, de la misma forma que el Timer 2, determina al azar el tiempo y cuando este se cumple desactiva el Timer 3 para que deje de mover el horario.

Subrutina CaminaHora

Se encarga de avanzar la manecilla del HORARIO cada vez que es requerido por el usuario, el movimiento se realiza al azar ya que se detendrá hasta llegar al punto asignado por la máquina para que posteriormente el usuario conteste correctamente la hora que apunta. (*Sección escribir la hora*)

Ver Figura 3-7 Diagrama de flujo de la subrutina CaminaHora.

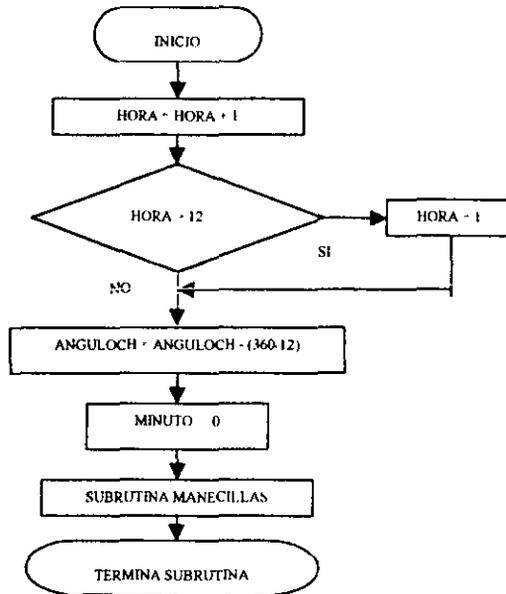


Figura 3-7 Diagrama de flujo de la subrutina CaminaHora.

Subrutina CaminaMinuto

Como se mostró en los códigos de los temporizadores 1 y 2, éstos llaman o ejecutan la subrutina siguiente que se encarga de avanzar o regresar la manecilla en el reloj de acuerdo al valor asignado a la variable Dirección, cada movimiento en la manecilla está determinado por un intervalo en grados equivalente a $\frac{1}{60}$ de la circunferencia total ya que debe recorrer la circunferencia total en 60 movimientos para que transcurra una hora, además cada movimiento de la manecilla grande (MINUTERO), provoca un pequeño movimiento de la manecilla pequeña (HORARIO), de tal forma que cada vez que la manecilla grande da una vuelta completa a la circunferencia, la pequeña sólo recorrerá la distancia entre un número y otro marcados en la carátula del reloj.

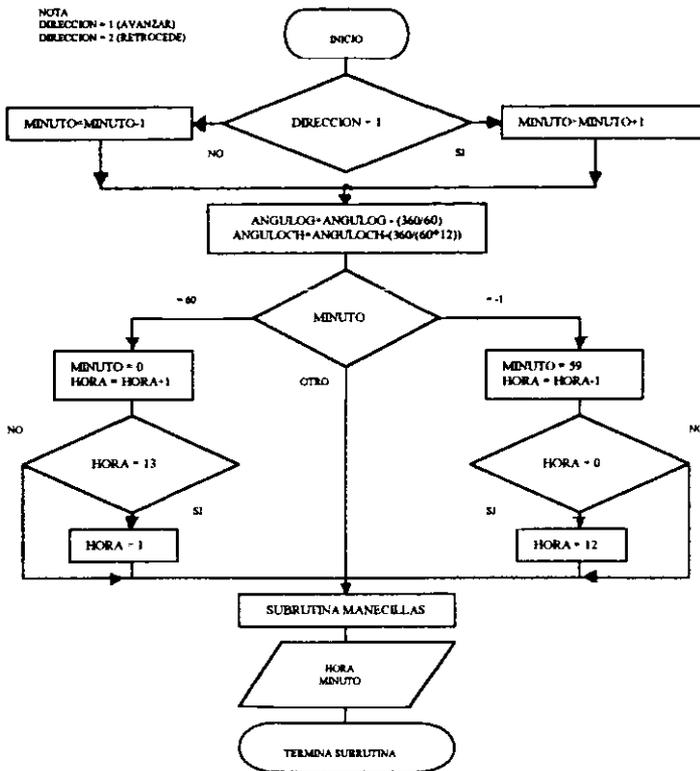


Figura 3-8 Diagrama de flujo de la subrutina CaminaMinuto.

Un aspecto importante para el desarrollo de cada una de las actividades en el programa es el establecer o asignar los valores iniciales a cada uno de los componentes y variables que están involucrados en los diferentes módulos del programa, así se especifica a continuación el estado inicial del controlador de audio (tomando en cuenta que en algunas máquinas carecen de tarjeta de audio se implementó una sentencia para evitar los errores y que la máquina se bloquee cuando detecte está falta, y se asigna la información a los arreglos de variables para mostrarla posteriormente en la opción de CONOCER.

Todo esto se lleva a cabo al cargarse la forma (*evento LOAD*) en memoria, mientras se manda a ejecutar el programa y éste está siendo leído por la máquina.

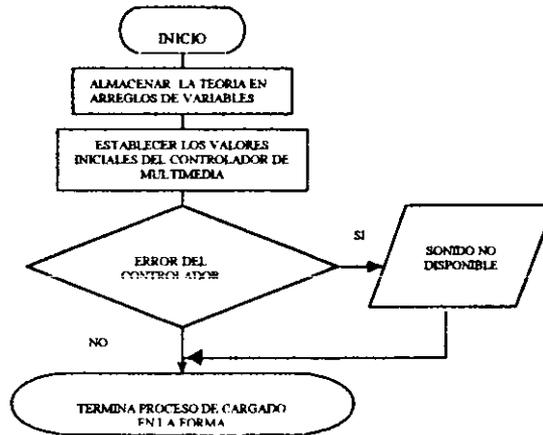


Figura 3-9 Diagrama de flujo del procedimiento asignado al evento LOAD de la forma
(Referencia tabla de la pág. 36).

Al activarse la forma, una vez que terminó de cargarse en memoria y se muestra sobre la pantalla de la computadora, se establece la escala del área de trabajo de la caja de pintura que tendrá 12 unidades de ancho y 12 unidades de alto, el tamaño de cada una de estas unidades, depende del tamaño del ancho y el alto que se le dio a la caja de pintura al dibujarse, se asignan los ángulos iniciales de la inclinación de las manecillas, que para cada una de ellas es de 90° , esto quiere decir que ambas estarán apuntando hacia el número 12.

Por último se ocultan todos los controles que no deben de mostrarse al inicio, ya que cada uno de ellos irán apareciendo de acuerdo a las opciones seleccionadas y al desarrollo de las actividades que desempeñe el alumno.

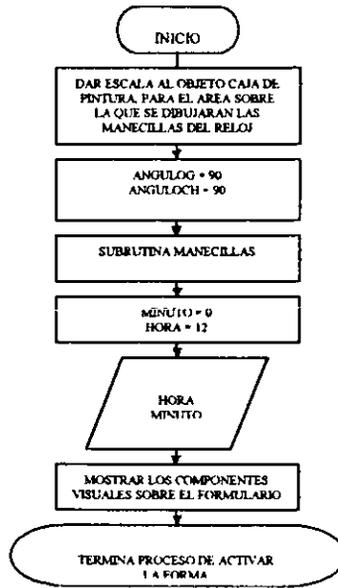


Figura 3-10 Diagrama de flujo del procedimiento asignado al evento ACTIVATE de la forma
(Referencia tabla de la pág. 36).

3.4.2 Pantalla de la opción APRENDE

Al entrar a esta opción, el usuario puede ir leyendo acerca de la forma en que funciona el reloj y como es que trabajan sus manecillas, la teoría se va escribiendo en una etiqueta (Label 3) llamada LetreroG, al mismo tiempo en que giran las manecillas simulando un reloj verdadero, la hora y el minuto, que marcan las manecillas, son escritos en las cajas de texto correspondientes.



Figura 3-10 Pantalla de la opción APRENDE.

Tabla de objetos de esta opción

Objeto	Nombre	Letrero ó Mensaje	Suceso
Panel1	Panel1	NINGUNO	NINGUNO
Picture1	Cuadro1	NINGUNO	NINGUNO
Picture2	Pintura	NINGUNO	NINGUNO
Picture3	Cuadro	NINGUNO	NINGUNO
Label1	ET1	Hora	NINGUNO
Text1	TextoH1	12	GotFocus, KeyPress
Command1	ControlHora(2)	Flecha hacia la Izquierda	Click

Objeto	Nombre	Letrero ó Mensaje	Suceso
Command2	ControlHora(1)	Flecha hacia la Derecha	Click
Label2	ET2	Minuto	NINGUNO
Text2	TextoM1	25	GotFocus, KeyPress
Command3	ControlMin(2)	Flecha hacia la Izquierda	Click
Command4	ControlMin(1)	Flecha hacia la Derecha	Click
Label3	LetreroG	NINGUNO	NINGUNO
Command5	Flecha(1)	Flecha hacia la Izquierda	Click
Command6	Flecha(2)	Flecha hacia la Derecha	Click
Command7	BOTONFin	Tache	Click

3.4.2.1 Descripción general de la pantalla de esta sección

- a) *Requerimiento funcional:* Pantalla que muestra los conceptos básicos para la lectura del reloj, incluyendo algunas actividades de autoevaluación.
- b) *Entrada:* Se requiere que el usuario seleccione las flechas de avanza y retrocede para revisar los conceptos básicos de la lectura del reloj, además de tener la posibilidad de mover las manecillas del reloj y escribir la hora correcta para las actividades de autoevaluación que vienen incluidas.
- c) *Proceso:* La máquina muestra la información cada que se seleccionan las flechas de avanza y retrocede al llegar a cierto número de explicaciones manda algunas actividades de autoevaluación para la revisión de los conceptos.
- d) *Salida:* Caja de texto con los conceptos y cuadros de mensaje para indicar los resultados de las actividades.

Sección de códigos de la pantalla de la opción Aprende

Al colocar el apuntador sobre la caja de textoH1, se seleccionará la hora escrita anteriormente para que se borre de forma automática al escribir la nueva hora.

Al terminar de escribir la nueva hora el usuario y presionar la tecla ENTER, se evaluará inmediatamente por la computadora, comparándola con la que apunta la manecilla del reloj e indicando el resultado al usuario por medio de cajas de mensaje.

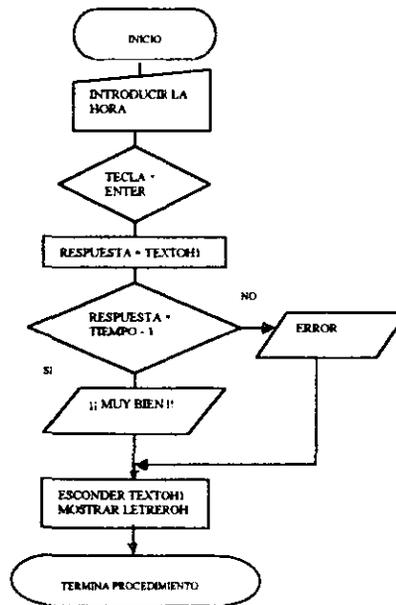


Figura 3-11 Diagrama de flujo del procedimiento asignado al evento KEYPRESS de la caja de TextH1.

(Referencia en la tabla de la pág. 49)

Subrutina Mensaje

Se encarga de llamar el cuadro que indica los resultados al usuario, los parámetros que requiere esta función son el *mensaje m* y el *título* del cuadro *l*.

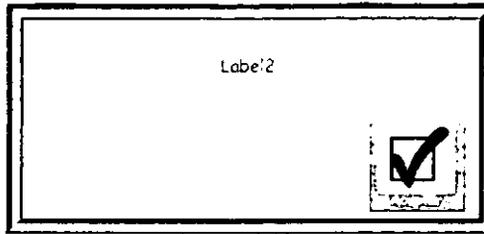


Figura 3-12 Figura de la ventana Mensaje

Tabla de objetos de la Ventana de Mensaje

Objeto	Nombre	Letrero ó Mensaje	Suceso
Panel	VMensaje	NINGUNO	NINGUNO
Label1	Label1	NINGUNO	NINGUNO
Label2	Label2	NINGUNO	NINGUNO
Command	Aceptar	Aceptar	Click

El procedimiento ConTrolHora, se encarga de mover la manecilla del reloj cada vez que se hace click sobre el botón de y de acuerdo al índice seleccionado se avanza o retrocede seleccionando cualquiera de los 2 casos.

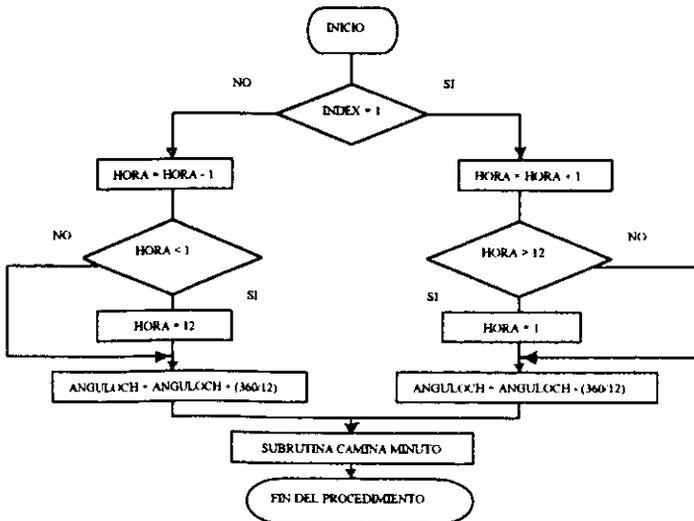


Figura 3-13 Procedimiento asignado al evento CLICK del arreglo de botones CONTROLHORA

(Referencia en la tabla de la pág. 49)

El procedimiento ControlMin se emplea para hacer que avance o retroceda el minuto seleccionado en las manecillas del reloj.

El código asignado al arreglo de controles llamado Flecha, se ejecuta cada vez que se selecciona alguna de las flechas de avance o retroceso, tomando en cuenta que de acuerdo al índice y el número de opción será la actividad que realizara el código.

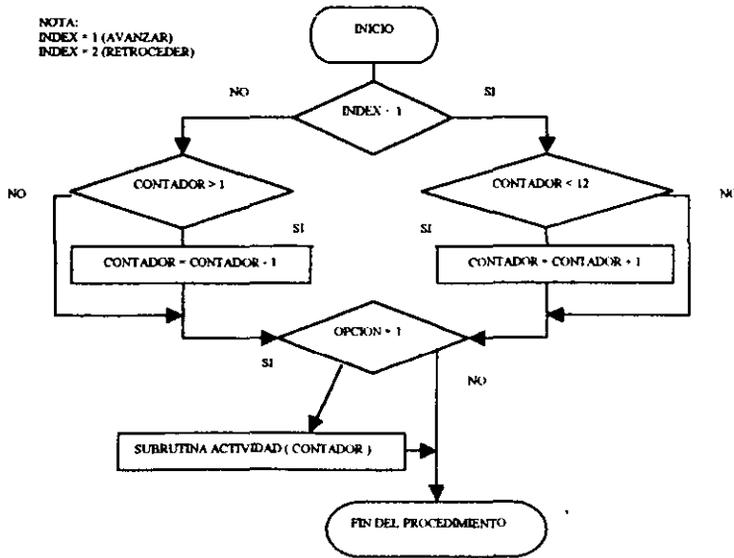


Figura 3-13 Procedimiento asignado al evento CLICK del arreglo de botones Flecha
(Referencia en la tabla de la pág. 49)

Subrutina Actividad

La actividad que se llevará a cabo será asignada por la siguiente subrutina, que se encargará de mostrar la teoría de la lectura del reloj para las tres primeras pantallas, mientras se mueven sus manecillas, en la pantalla 4 se moverá al azar el MINUTERO para que el alumno indique el nuevo minuto y el programa lo evalúe, las pantallas 5,6,7,9,10,11 muestran más teoría de la lectura de las manecillas del reloj, la pantalla 8 realiza una actividad con la manecilla HORARIO para reafirmar los conocimientos adquiridos y la pantalla 12 indica el fin de la actividad.

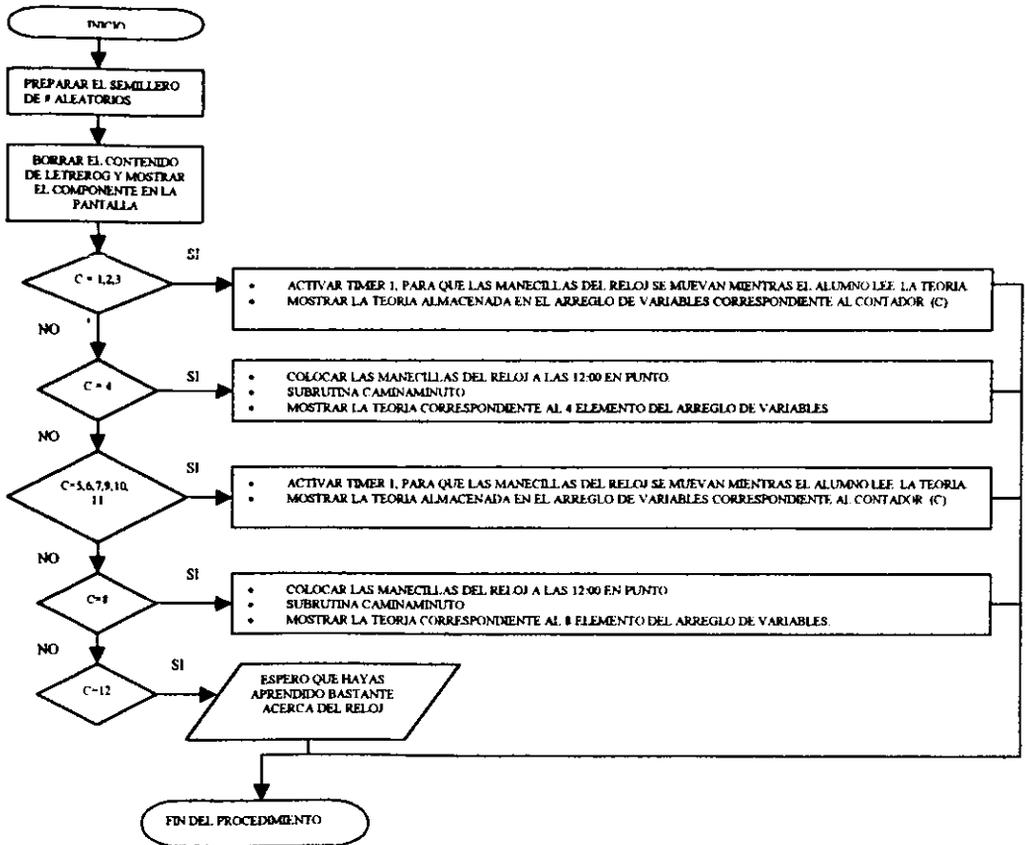


Figura 3-14 Diagrama de flujo de la subrutina Actividad

3.4.3 Opción BUSCAR LA HORA

En esta sección el usuario puede manipular las manecillas del reloj, empleando las flechas que indican la hora y el minuto para localizar la hora indicada, en forma aleatoria, por la computadora.

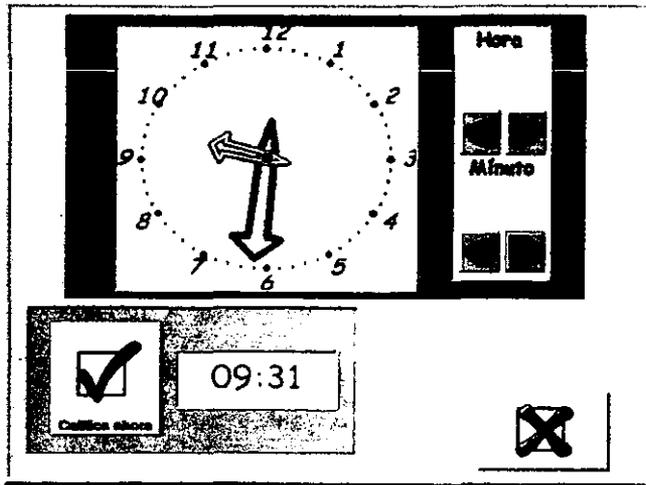


Figura 3-15 Pantalla de la opción Buscar la hora

Tabla de objetos y sección de códigos

Objeto	Nombre	Letrero ó mensaje	Suceso
PictureBox1	Cuadro2	NINGUNO	NINGUNO
Command1	BOTONCalif	Califica ahora	Click
Label1	LetreLeer	09:31	NINGUNO

Descripción general de la pantalla

- a) *Requerimiento funcional*: Pantalla que permite manejar las manecillas del reloj para apuntar a la hora marcada por la máquina.

- b) *Entrada*: Se requiere que el usuario seleccione las flechas de la hora y el minuto para poder mover las manecillas del reloj a la hora indicada, posteriormente pueda verificar si la actividad es correcta o no.
- c) *Proceso*: La máquina muestra en una etiqueta la hora al azar para que el alumno la localice con las flechas correspondientes, posteriormente puede hacer click en el botón de "Califica ahora" para ver su resultado.
- d) *Salida*: Etiqueta para mostrar la hora al azar y cuadros de mensaje para indicar los resultados de las actividades.

3.4.4 Sección ESCRIBE LA HORA

Es la última sección del software y se encarga de preguntar aleatoriamente la hora que indican las manecillas del reloj, el usuario tendrá que escribir en las cajas correspondientes la hora y el minuto para ser evaluado por la máquina.

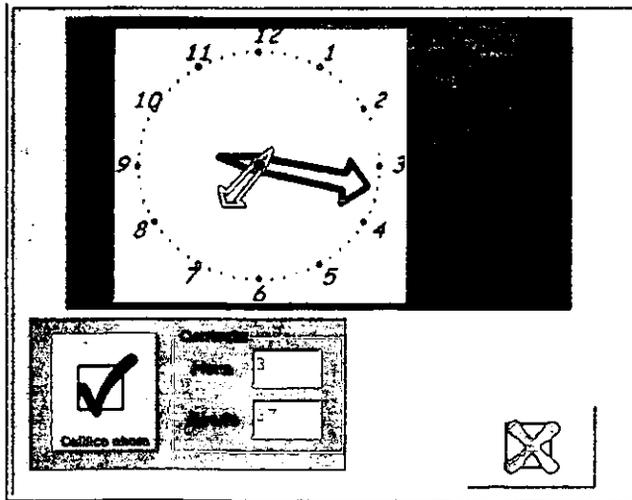


Figura 3-16 Pantalla de la opción Escribe la hora.

Tabla de objetos para esta sección

Para esta sección se utilizaron los mismos objetos a excepción del *frame* con el título *Contesta* que se encarga de mostrar los controles necesarios para escribir la hora y el minuto correcto indicado por las manecillas del reloj.

Objeto	Nombre	Letrero ó mensaje	Suceso
Frame	Fcontesta	Contesta	NINGUNO
Label	ET3	Hora	NINGUNO
Label	ET4	Minuto	NINGUNO
TextBox	TextoH	3	GotFocus, Keypress
TextBox	TextoM2	37	GotFocus

Descripción general de la pantalla de esta sección

- e) *Requerimiento funcional*: Pantalla que califica al usuario la escritura correcta de la hora indicada por el reloj.
- f) *Entrada*: Se requiere que el usuario escriba la hora correcta en la caja de textos llamada TextoH y el minuto correcto en la caja de textos llamada TextoM2, posteriormente hacer click en el botón de la leyenda "Califica ahora".
- g) *Proceso*: La máquina selecciona al azar 10 horas diferentes e indica al alumno que debe de escribir correctamente la hora indicada.
- h) *Salida*: Cuadro de mensaje indicando si hay acierto o error.

Procedimiento BOTONCalif ()

(La siguiente sección de códigos se ejecuta al hacer clic sobre el botón mencionado)

Para la sección de *Escribir la hora* como para *Buscar la hora*, este botón se encarga de calificar los resultados, haciendo una comparación de lo que contesta el usuario con lo que muestra la computadora, para poder asignar un resultado.

La variable que indica si esta buscando o escribiendo la hora se llama OPCION, al tomar el valor de 3 evalúa la posición de las manecillas (BUSCAR LA HORA), cuando es 2 simplemente verifica que el valor de la hora y minuto escritos en las cajas de texto correspondientes concuerden con la seleccionada por la computadora(ESCRIBIR LA HORA).

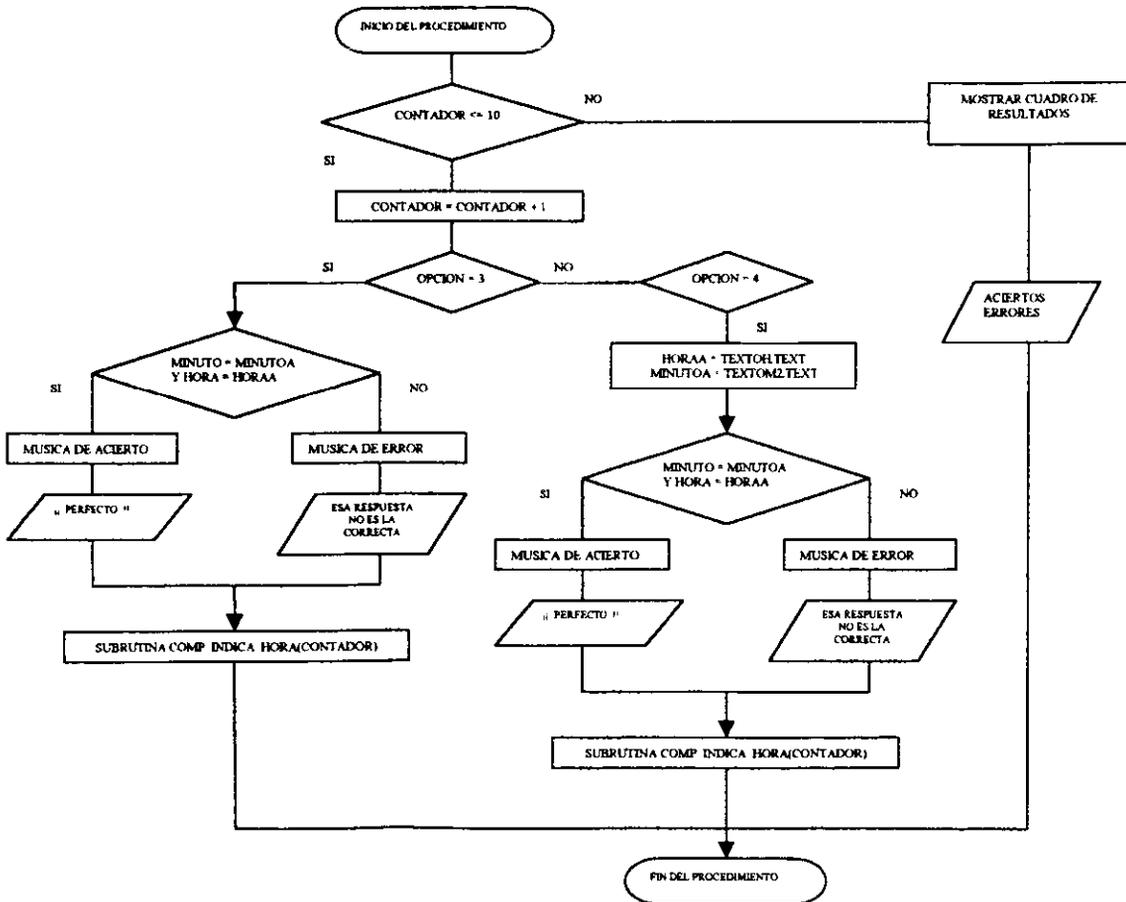


Figura 3-17 Procedimiento asignado al evento CLICK del BOTONCALIF.

3.5 Aplicación y explotación del software resultante

Este software está destinado para alumnos de 2do. Grado de primaria ya que es en este momento en el que se le enseña al alumno a leer las manecillas del reloj, en la primaria La Salle Francia.

Diseño de validación

En dicha escuela hay 4 grupos de 2do. Grado con 46 alumnos en promedio cada uno de ellos, y se decidió aplicar el software solo a dos de los grupos mencionados para así determinar los resultados obtenidos al evaluar los dos grupos que estudiaron los conceptos de la lectura del reloj siguiendo las técnicas tradicionales del salón de teoría y dos grupos que estudiaron utilizando el software en la computadora como un apoyo en sus actividades.

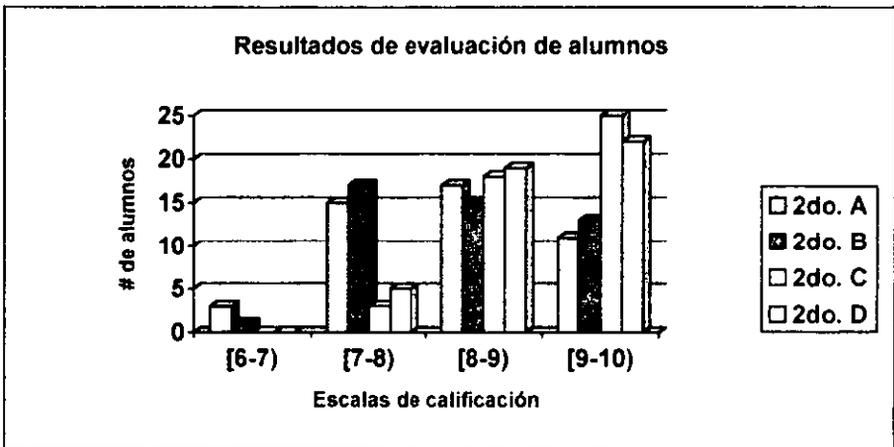
La evaluación efectuada a los alumnos de los 4 grupos anteriores consistió en dos etapas diferentes:

1. Se les proporcionó una hoja de papel que tenía dibujadas 10 carátulas de relojes marcando horas diferentes y se les pidió a los alumnos escribir debajo de cada una de ellas la hora que indicaban sus manecillas.
2. Una vez terminada la primera etapa se les proporcionó otra hoja de papel con 10 dibujos de carátulas de relojes sin manecillas pero debajo de cada una de ellas se les escribió una hora diferente y se les pidió a los alumnos que dibujaran las manecillas sobre dichas carátulas indicando la hora escrita.

Al terminó de las actividades se observaron los siguientes resultados:

Grupo	Escala de calificaciones resultantes con escalas de 5 a 10				
	[6-7)	[7-8)	[8-9)	[9-10)	Total
2do. A	3	15	17	11	46
2do. B	1	17	15	13	46
2do. C	0	3	18	25	46
2do. D	0	5	19	22	46

Figura 3-18. Representación gráfica de resultados de evaluación de alumnos.



3.5.1 Análisis y discusión de resultados

Se asignó el software a los grupos 2do. C y 2do. D, y por los resultados obtenidos en la tabla, se ve claramente como aumentó considerablemente el número de alumnos que aprendió el tema ya que fueron evaluados con el mismo examen en el salón de teoría y sin embargo los grupos que no usaron el software tienen un grado de aceptación bueno en sus calificaciones pero les falta un poco de precisión.

CAPITULO IV APLICACION 2

"EL MAPA DE LA REPUBLICA MEXICANA"

El programa de la República Mexicana se implementó en La Salle de Francia, ya que se buscó la manera en la que el niño aprenda en una forma más amena e incluso en forma de juego a memorizar una gran cantidad de información como lo es el manejo de la situación geográfica de cualquier país e incluso conocer y manejar los nombres de los estados y de la capital de cada uno de ellos, se buscó otra alternativa ya que constantemente el profesor observó la necesidad de material didáctico a parte de los grandes mapas que constantemente lleva al salón de clase, por tal motivo se pensó en utilizar la computadora para atraer al niño en este tipo de actividades e interesar al alumno de tal forma que por el mismo explore y conozca la división política de México.

Se pensó en la computadora ya que por si sola es un medio altamente atrayente para los niños que cursan el 3er grado de primaria e incluso se armaron actividades para que ellos mismos se tracen metas a seguir, como el alcanzar una calificación alta en las diferentes actividades realizadas dentro del software mencionado.

Se tuvo la necesidad de realizar este software, ya que, aunque existen programas en el mercado que manejan mapas y enciclopedias, la mayor parte de los programas de geografía solamente son de consulta, esto quiere decir que no traen ejercicios de aplicación para que el alumno refuerce los conocimientos aprendidos y ésta era una de las necesidades que el alumno requería.

4.1 Objetivo del programa

Apoyar al niño en la memorización de los diferentes estados y capitales de la República Mexicana, así como su localización geográfica.

4.2 Descripción del programa

Por la forma en que trabaja este software, lo clasificamos como un programa de ejercicio y práctica, ya que proporciona prácticas y ejercicios para que el alumno refuerce la información adquirida y adquiera una cierta habilidad en el tema.

El programa del Mapa de la República Mexicana se divide en cuatro diferentes secciones:

1) *Repaso:* Es la sección, por medio de la cual, el alumno puede conocer la localización política de la República Mexicana y cada uno de sus estados y su respectiva capital directamente con un mapa, por medio de una breve presentación, que el alumno controla, y en donde se muestra el estado y su capital.

2) *Localizar:* En esta sección, el usuario puede jugar a localizar los estados indicados al azar por la computadora seleccionando en el mapa cada uno de ellos y directamente la máquina lo califica indicando la capital si es que el resultado es satisfactorio e incrementando puntos que se muestran como estímulos para el alumno.

3) *Escribir la capital:* Aquí el usuario tendrá que escribir la capital del estado indicado en el mapa, también se estimula al usuario contando aciertos y errores.

4) *Salida:* Al seleccionar esta opción el alumno puede abandonar el programa y terminar con sus actividades.

4.3 Diagrama de bloques del programa

Para una mejor comprensión de las secciones en que esta dividido el programa del mapa de la República Mexicana se muestra el siguiente diagrama de bloques.

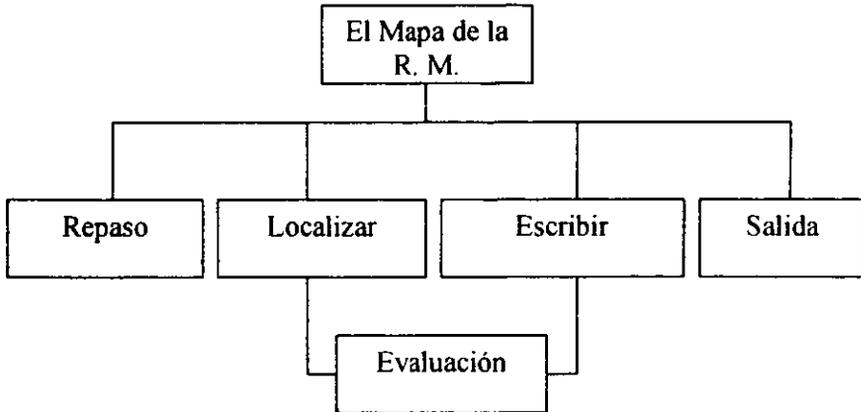


Figura 4-1 Diagrama de bloques del programa

Las secciones de *Localizar el Estado* y *Escribir la Capital*, nos muestran al final de la ejercitación un cuadro de resultados totales.

4.4 Descripción de cada uno de los bloques del programa

Debido a la extensión del programa se consideró el analizar y describir en forma aislada cada uno de sus bloques, mostrando GUI (Interfaz gráfico de usuario), diagrama de inferencia y código para cada uno de ellos.

4.4.1 Pantalla Principal

Es mostrada al inicio del programa y por medio de esta, se puede tener acceso a cada uno de los bloques del programa, seleccionando con el indicador del Mouse la opción deseada y haciendo un Click sobre ella.



Figura 4-2 Pantalla principal

Tabla de Objetos y sucesos utilizados para esta sección

Objeto	Nombre	Letrero ó Mensaje	Suceso
Formulario	Ventana	República Mexicana	Create ¹
Panel1	Panel1	NINGUNO	NINGUNO
Image1	Mapa	NINGUNO	MouseDown ²
Button1	BOTON1	Repaso	Click
Button2	BOTON2	Localizar	Click
Button3	BOTON3	Contestar	Click
Button4	BOTON4	Salida	Click

¹ Los códigos asignados al evento CREATE, se ejecutan automáticamente en el momento en que se corre el programa y aparece la pantalla principal en el monitor

² Los códigos asignados al evento MouseDown, se ejecutan en el momento en que se hace Click con algún botón del mouse sobre el mapa

- a) *Requerimiento funcional:* Pantalla para elegir las opciones Aprender, Escribir la hora, Buscar la hora y salida.
- b) *Entrada:* Se requiere para que el usuario seleccione a donde desea ingresar.
- c) *Proceso:* El usuario al seleccionar la opción lo llevara a la actividad elegida.
- d) *Salida:* Pantalla para elegir opción.

4.4.1.1 Descripción de variables utilizadas para este programa

En primer lugar se declaró un registro formado por 4 campos: Edo, Cap, PosX, PosY que son el Estado, la Capital y el par de coordenadas sobre las cuales se localiza el estado sobre la imagen del mapa de la República Mexicana, se declaró de esta forma para almacenar en una forma más eficiente toda la información de los 31 estados para llevar a cabo el juego.

Además se declararon otras variables para el manejo del programa

Mex : RecordMap;

Es el registro que permitirá tener acceso a cada uno de los campos de información.

NumEdo, OPCION, Turno, Aciertos

Son variables que controlan el número de estado seleccionado, la opción o actividad a realizar, el turno y número de aciertos, respectivamente. (Almacenan números enteros)

REdo : array[1..31] of Integer;

Nos permite ir almacenando las respuestas para verificarlas con las del usuario.

Aviso : Boolean;

Permite saber si alguna opción está disponible o no.

3.4.1.2 Descripción de subrutinas y procedimientos de la pantalla principal

En primera instancia, en el momento en que se ejecuta el programa y aparece el mapa en la pantalla, automáticamente se almacena la información necesaria en los registros correspondientes, para manejarla posteriormente en el transcurso del programa y se iluminan los estados del mapa

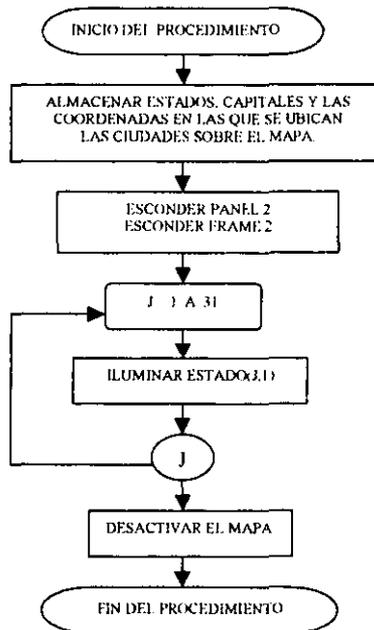


Figura 4-3 Códigos asignados al evento *CREATE* de la forma.

Como se puede observar al final del código del procedimiento anterior, se hace referencia a las subrutinas siguientes:

Subrutina Iluminar_Estado

Esta subrutina se encarga de iluminar cada uno de los estados de la República Mexicana, utilizando el color asignado en el argumento del procedimiento.

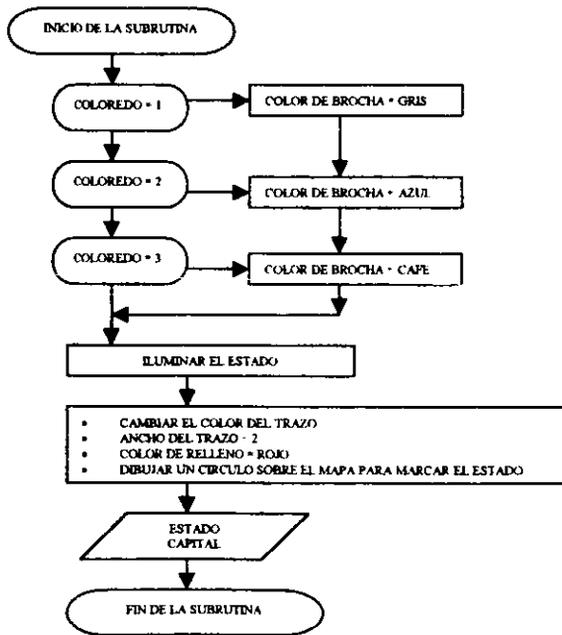


Figura 4-4 Diagrama de flujo de la subrutina Iluminar_Estado.

Los siguientes cuatro botones Botón 1, Botón 2, Botón 3 y Botón 4, se encargan de ejecutar las actividades Conocer, escribir, localizar y terminar respectivamente, el diagrama de flujo se muestra a continuación, las actividades que se llevan a cabo al hacer click con el apuntador del mouse sobre el botón 1 son las siguientes:

Al hacer click sobre el botón 1, entraremos a la opción CONOCER, inmediatamente se desactivaran los botones que no se pueden usar en esta opción, se esconderá el cuadro de resultados y evaluación y sólo quedará el mapa con las flechas activadas para que el usuario pueda conocer cada uno de los estados que se mostrarán, por medio de la *subrutina CONOCE*.

Ver la Figura 4-5 Diagrama de flujo del botón 1.

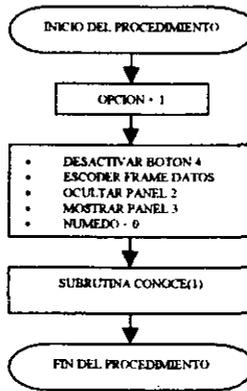


Figura 4-5 Diagrama de flujo del botón 1.

Al hacer click sobre el botón 2, aparece un cuadro en la pantalla que nos indicará los resultados obtenidos al termino de la evaluación, se desactivan los botones que no se pueden utilizar en esta opción y se ejecuta la *subrutina LOCALIZA* que hace las actividades correspondientes.

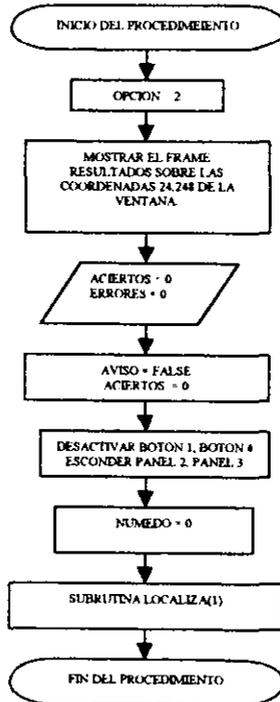


Figura 4-6 Diagrama de flujo del botón 2

Al hacer click sobre el botón 2, aparece un cuadro en la pantalla que nos indicará los resultados obtenidos al termino de la evaluación, un cuadro que muestra los estados a localizar, se desactivan los botones que no se pueden utilizar en esta opción y se ejecuta la *subrutina LOCALIZA* que hace las actividades correspondientes.

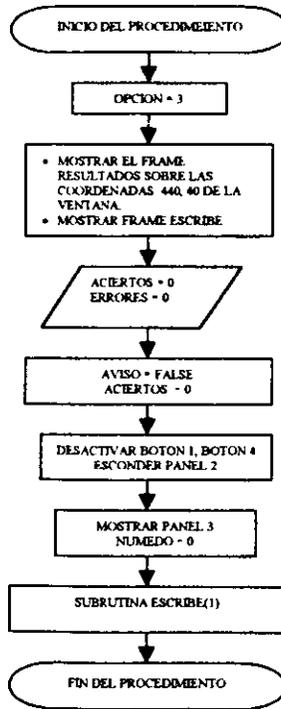


Figura 4-7 Diagrama de flujo del botón 3.

Al hacer click sobre el botón 4 se pregunta si se desea salir o no y de acuerdo a la respuesta seleccionada por el usuario, se termina o sigue el programa.

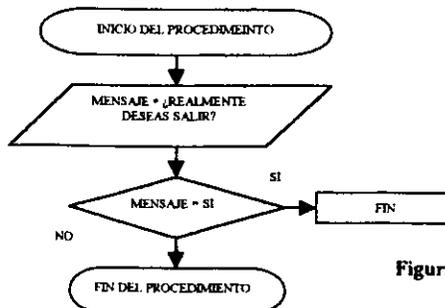


Figura 4-8 Diagrama de flujo del botón 4.

4.4.2 Pantalla de la opción REPASO

Es de gran importancia esta sección, ya que es aquí donde el alumno se familiariza con el mapa de la República Mexicana, conoce cada uno de los estados, sus nombres y sus capitales.



Figura 4-9 Pantalla de la opción REPASO

Tabla de objetos de esta opción

Objeto	Nombre	Letrero ó Mensaje	Suceso
Panel2	FrameDatos	NINGUNO	NINGUNO
Label1	Label5	Datos Principales	NINGUNO
Label2	Et1	Estado :	NINGUNO
Label3	EtEstado	EtEstado	NINGUNO
Label4	Et2	Capital :	NINGUNO
Label5	EtCapital	EtCapital	NINGUNO

Panel3	Panel3	NINGUNO	NINGUNO
Button5	BitBtn1	Retrocede	Click
Button6	BitBtn2	Avanza	Click
Button7	BitBtn3	Terminar	Click

Descripción general de la pantalla de esta sección

- Requerimiento funcional:* Pantalla que muestra el mapa de la República Mexicana, sus estados y nombres de cada uno de ellos incluyendo las capitales correspondientes.
- Entrada:* Se requiere que el usuario seleccione las flechas de avanza y retrocede para que se ilumine el estado correspondiente y aparezcan sus datos (*estado y capital*) en el cuadro superior.
- Proceso:* La máquina muestra la información cada que se seleccionan las flechas de avanza y retrocede.
- Salida:* Caja de imagen que muestra el mapa y tiene la opción de delimitar el área deseada, etiquetas para imprimir los detalles de cada estado.

Cuando el usuario hace Click sobre el botón *Retrocede* tiene dos opciones ya que este botón es utilizado para dos módulos (*Conoce y Localiza*) la opción de regresar al estado anterior si es que desea volver a visualizarlo ó tiene la posibilidad de localizar un estado diferente (módulo *Aprende*) ó si es que se equivocó (*módulo localizar*).

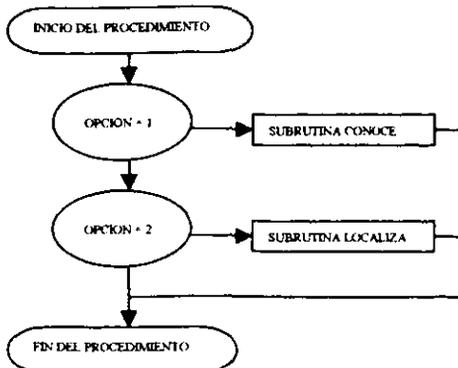


Figura 4-10 Procedimiento del botón Retrocede.

El procedimiento anterior ejecuta la subrutina Conoce y esta a su vez se encarga de ejecutar las subrutinas Intervalo, Limpiar_Mapas e Iluminar_estado.

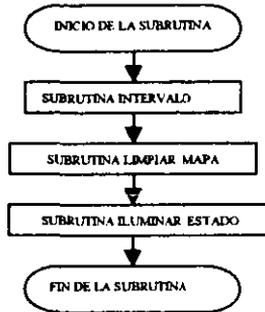


Figura 4-11 Diagrama de flujo de la subrutina Conoce.

Intervalo se encarga de decidir si se incrementa o se reduce el número indicador de estado ya que sólo se puede recorrer desde el estado 1 al 31, al llegar a cada uno de los límites deja de aumentar o reducir el incremento.

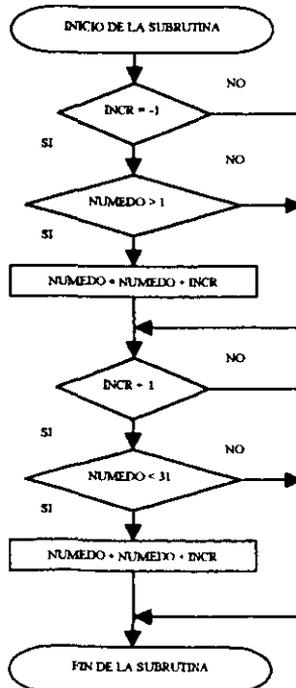


Figura 4-12 Diagrama de flujo de la subrutina Intervalo.

Limpiar_Mapa cada que es ejecutada se encarga de borrar el estado anterior en el mapa para que se ilumine el siguiente. En este procedimiento se utiliza un circuito y se cambia el color de la brocha y posteriormente se ilumina cada una de las áreas en el mapa.

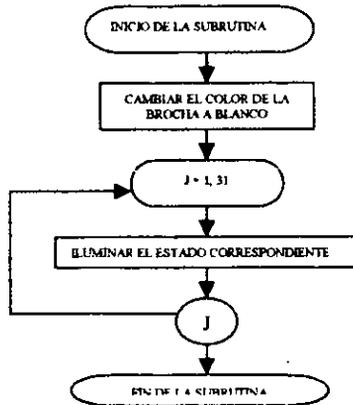


Figura 4-12 Diagrama de flujo de la subrutina Limpiar_Mapa.

Al hacer Click sobre el botón Terminar, automáticamente se desactivan las opciones y se ocultan aquellos controles que ya no se visualizarán en la pantalla principal debido a que nos regresa a está para seleccionar una nueva opción.

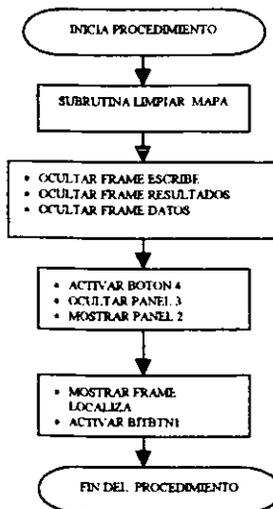


Figura 4-13 Diagrama de flujo de la subrutina Limpiar_Mapa

Al hacer Click sobre el botón *Avanza* se tienen cuatro posibles actividades, de acuerdo al módulo en el que nos encontremos Conocer, Localizar o Escribir, ya que este botón se implementó para trabajar en las tres diferentes opciones dentro del programa.

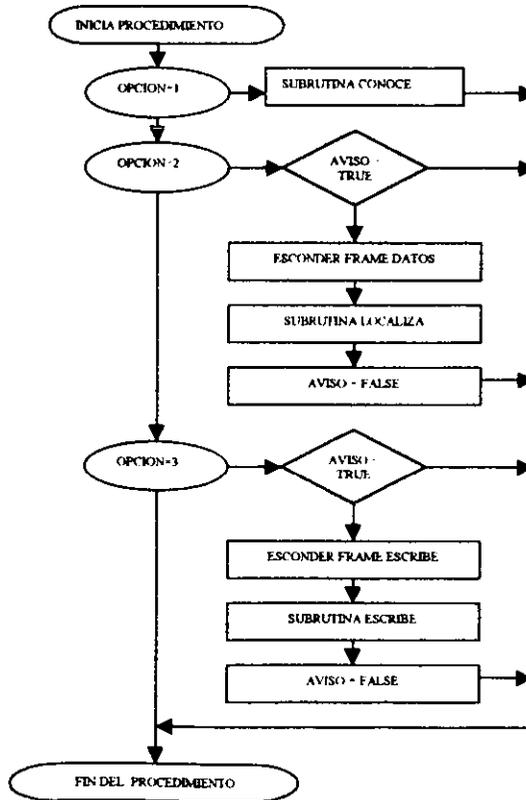


Figura 4-13 Diagrama de flujo del botón Avanza

4.4.3 Opción LOCALIZAR

En el módulo localizar, el alumno interactúa con la computadora seleccionando la capital del estado que ésta pregunta al azar, haciendo un click sobre éste, cuando el alumno acierta la máquina le incrementa puntos, de lo contrario le indica el número de errores que ha cometido.



Figura 4-14 Pantalla de la opción Localizar

Tabla de objetos

Objeto	Nombre	Letrero ó mensaje	Suceso
Panel4	FrameResultados	NINGUNO	NINGUNO
Label5	Label7	Resultados	NINGUNO
Label6	Label1	Aciertos	NINGUNO
Label7	Label2	Errores	NINGUNO
Label8	Label3	2	NINGUNO
Label9	Label4	1	NINGUNO

Panel5	FrameLocaliza	NINGUNO	NINGUNO
Label10	Label6	Localiza la capital del estado de:	NINGUNO
Label11	EtEstado2	Yucatán	NINGUNO

- a) *Requerimiento funcional:* Pantalla que permite al alumno interactuar con la computadora y jugar seleccionando la capital de los estados indicados para obtener un buen promedio de puntos..
- b) *Entrada:* Se requiere que el usuario haga un click sobre la capital del estado que la máquina le indica, posteriormente seleccione la flecha de Continuar para visualizar el siguiente estado hasta completar 10 oportunidades.
- c) *Proceso:* La máquina muestra en una etiqueta un estado al azar para que el alumno localice su respectiva capital, posteriormente puede hacer click en el botón *Avanza* para seleccionar el siguiente estado.
- d) *Salida:* Etiqueta para indicar un estado al azar y cuadros de mensaje para indicar los resultados de las actividades.

Sección de códigos para esta opción

Se explica al principio la Función Aleatorio, ésta nos regresa un número al azar entre 1 y 31 que no puede repetirse ya que en cada ocasión el estado debe ser diferente.

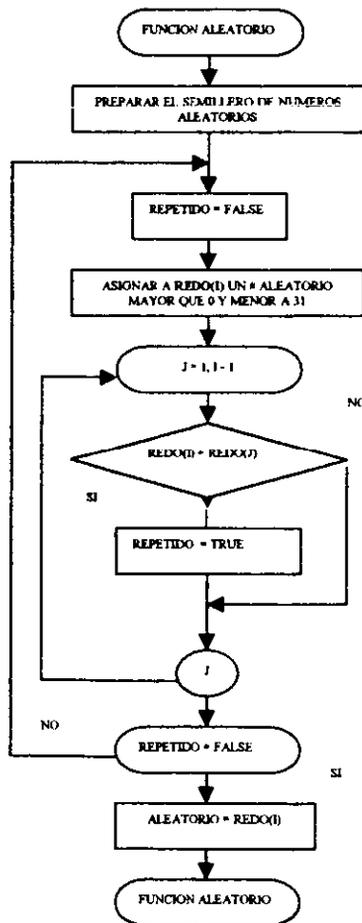


Figura 4-15 Diagrama de flujo de la función Aleatorio.

La subrutina Localiza se encarga de ejecutar la subrutina limpia_estado para que borre los estados iluminados anteriormente, llama a la función Aleatorio para saber qué estado será el que se ilumine ahora, finalmente lo ilumina empleando la subrutina Iluminar_estado y muestra en la etiqueta el estado que debe ser localizado.

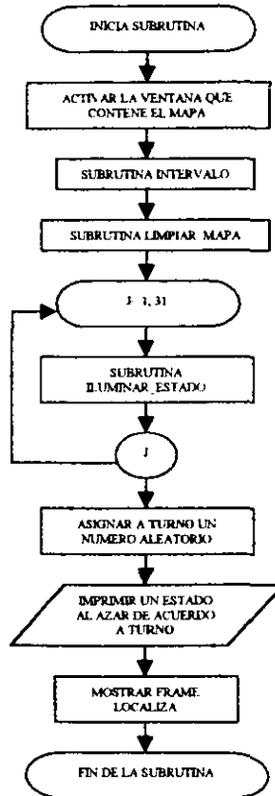


Figura 4-16 Diagrama de flujo de la subrutina Localiza.

Uno de los procesos más importantes en este módulo es el seleccionar la capital correcta sobre el mapa, esta selección se lleva a cabo cuando se hace Click sobre el mapa con el botón izquierdo del mouse, la máquina busca las coordenadas sobre las cuales debe hacerse click en el registro RecordMap almacenadas en los campos PosX, PosY.

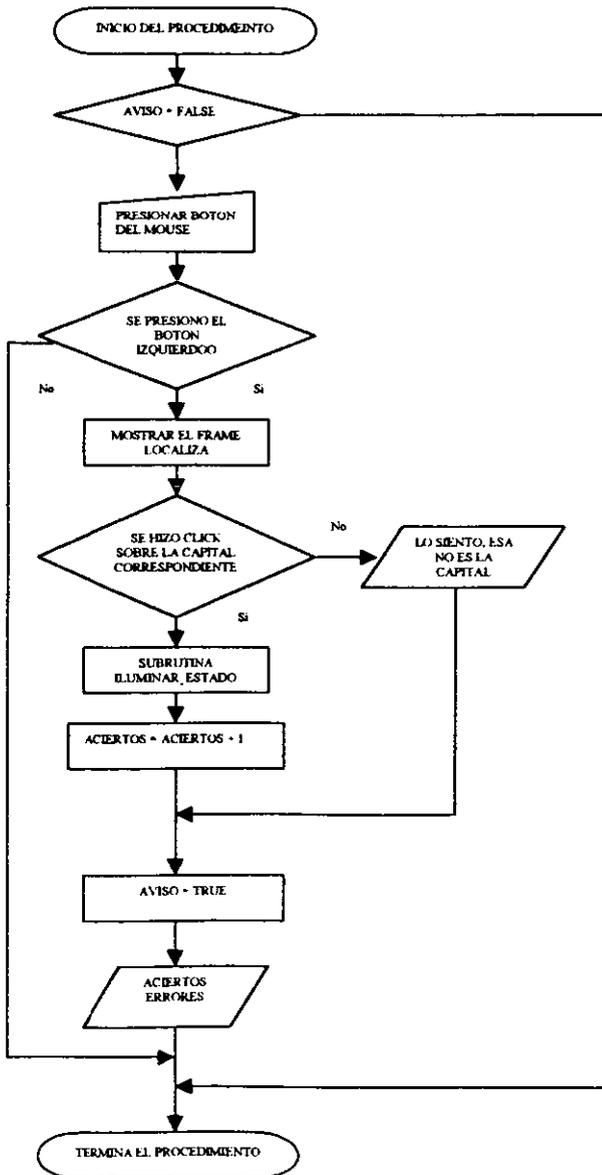


Figura 4-17 Actividades asignadas al eventoMouseDown del Mapa.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

4.4.4 Opción ESCRIBE LA CAPITAL

Es la última sección del software y se encarga de preguntar aleatoriamente la hora que indican las manecillas del reloj, el usuario tendrá que escribir en las cajas correspondientes la hora y el minuto para ser evaluado por la máquina.



Figura. 4-17 Pantalla de la opción Escribe la Capital

Tabla de objetos para esta sección

Objeto	Nombre	Letrero ó mensaje	Suceso
Panel4	FrameEscribe	NINGUNO	NINGUNO
Label12	Label8	Escribe la capital del estado de.	NINGUNO
Label13	EscEdo	Aguascalientes	NINGUNO
Label14	Label9	Capital :	NINGUNO
Edt1	Edt1	NINGUNO	NINGUNO

Descripción general de la pantalla de esta sección

- a) *Requerimiento funcional:* Pantalla que califica al preguntar sobre la capital de algún estado de la República Mexicana.
- b) *Entrada:* Se requiere que el usuario escriba la capital del estado que aparece en la parte inferior de la pantalla.
- c) *Proceso:* La máquina selecciona al azar 10 estados diferentes y los muestra en la pantalla inferior, posteriormente pregunta al usuario la capital y evalúa dicha respuesta de acuerdo a la correspondencia con el estado y su escritura correcta, finalmente muestra los errores y aciertos correspondientes.
- d) *Salida:* Se ilumina el estado que fue escrito correctamente e indica los posibles aciertos y errores.

Se incluye la *subrutina Escribe*, que se encarga de llamar a las subrutinas INTERVALO, LIMPIAR e ILUMINAR, explicadas en el tema anterior, una vez que preguntó al azar la capital del estado, muestra la caja de edición de texto que es el control en el que se escribirá la respuesta correcta.

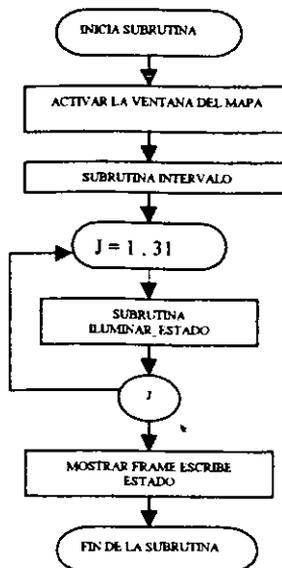


Figura 4-17 Diagrama de flujo de la subrutina Escribe.

PRESIONAR ENTER DEPUES DE ESCRIBIR LA CAPITAL DEL ESTADO CORRESPONDIENTE

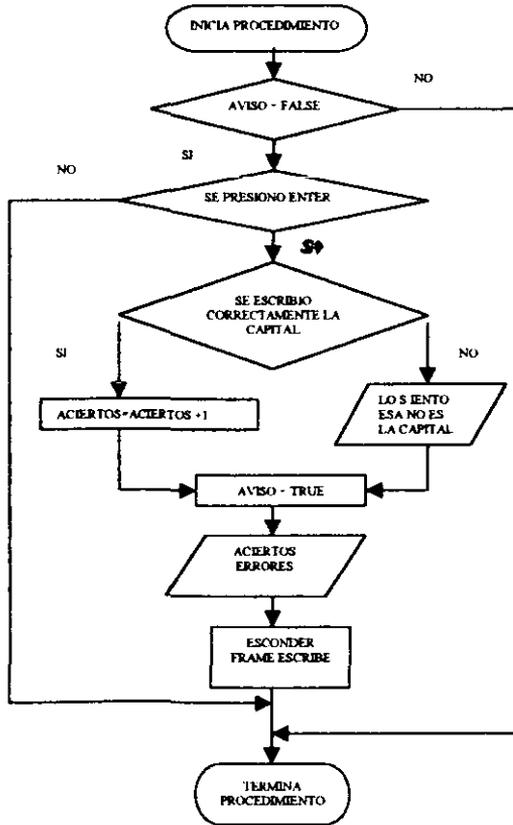


Figura 4-18 Procedimiento asignado al ENTER después de escribir la Capital.

4.5 Análisis de la implantación del software

La implantación de este software se llevó a cabo en grupos del 3er. Grado de educación primaria y resultó muy positivo ya que desde el primer momento de conocerlo se identificaron con éste y se familiarizaron en una forma muy rápida con cada una de las actividades que éste contiene.

Se aplicó en una forma muy parecida a l software del reloj, pero en este caso sólo hubo tres grupos de 3er. Grado, por lo tanto se tuvo que decidir hacer la comparación excluyendo uno de los tres grupos para facilitar la evaluación.

Uno de los grupos estudió la localización ene le salón de clase utilizando un mapa tradicional que el profesor proporcionó y el otro grupo realizó sus actividades en el laboratorio de computación utilizando el software mencionado.

Al aplicar el examen a los estudiantes los resultados fueron los siguientes:

Grupo	Escala de calificaciones resultantes con escalas de 5 a 10				
	[6-7)	[7-8)	[8-9)	[9-10)	Total
Tercero A	5	15	22	5	47
Tercero B	1	3	23	19	46

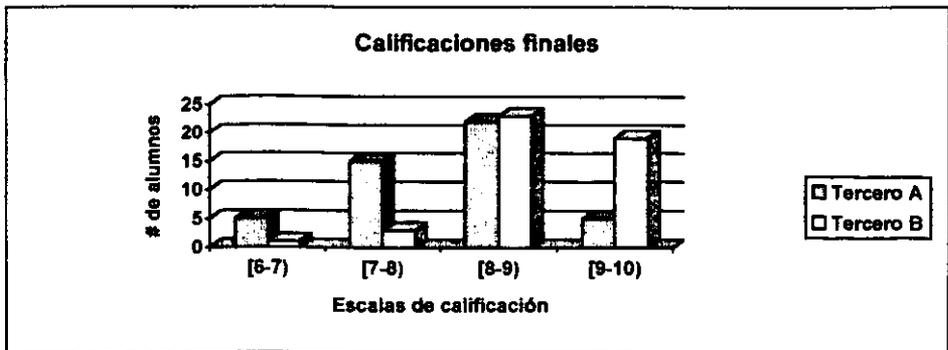


Figura 4-19. Representación gráfica de resultados de evaluación de alumnos.

4.5.1 Análisis y discusión de resultados

De acuerdo a los resultados obtenidos, el avance fue más notorio con este programa que fue aplicado a los alumnos del tercero B y tomando en cuenta que independientemente de las actividades realizadas por cada uno de los grupos, el examen fue exactamente el mismo para cada uno de ellos y se aplicó en un salón de clase tradicional.

CONCLUSIONES DEL PROYECTO

Para concluir con esta memoria de desempeño profesional, quiero mencionar que los objetivos planteados al principio de este proyecto se pudieron llevar a cabo de manera exitosa, ya que como se explicó durante el desarrollo del mismo, se pudieron crear dos aplicaciones específicas de software educativo que contribuyeron en cierta forma en el apoyo del aprendizaje de algunos grupos de la institución educativa a la que se destinó dichos programas.

Esto se sustenta con los estudios o estadísticas que se llevaron a cabo una vez que el software resultante se utilizó en la primaria La Salle Francia, ya que como se mencionó anteriormente, los alumnos mejoraron gradualmente su nivel académico e incluso se observó que la apatía en algunos casos fue en disminución al ser atraídos por la presentación del software en cuestión.

Es importante resaltar en el proyecto que al trabajar en combinación de gente involucrada al 100% en los procesos de enseñanza se obtiene mejores resultados que aquellos en los que se plantea un modelo pero no se apoya en la visión o en las necesidades que este grupo puede aportar, el proyecto se llevó a cabo gracias a la intervención de los profesores de 2do. Y 3er. Grado de la institución, que con su experiencia directa en el área de educación, aportaron una serie de actividades que ellos realmente necesitaban aplicar entre sus alumnos y que no se cubrían en algún otro software existente en el mercado.

Aunando esfuerzos, aplicando algunas herramientas de matemáticas aplicadas y computación y la valiosa ayuda de los profesores, que de acuerdo a sus necesidades, plantearon diferentes actividades, se logró llevar a cabo este pequeño proyecto de software educativo, el cual en sus inicios se aplicó a un sector de la sección primaria, pero que gracias a los resultados satisfactorios observados, comenzó a difundirse en los demás grados que también aportaron ideas de actividades nuevas y no solamente en la sección

primaria ya que también se difundió en el área de preescolar y poco a poco fue tomando fuerza.

Es importante hacer hincapié y alentar este tipo de esfuerzos para erradicar en lo posible, la adaptación de las actividades de los educadores a las características que ofrecen los programas omitiendo necesidades que son necesarias para su desempeño académico al utilizar el software educativo que hoy en día existe en el mercado, si no al contrario que el software seleccionado se acople a las necesidades reales en las instituciones educativas, ya que esto trae como consecuencia una serie de resultados satisfactorios que facilita el desempeño en ambas partes.

Inicialmente se desarrolló el software *del reloj* y el *mapa de la República Mexicana*, posteriormente debido a la aceptación observada de estos programas tanto por alumnos como profesores, se crearon nuevos títulos como: *servidores públicos*, *Así soy yo¹* (*El cuerpo humano y hábitos de higiene*), *Jugando con los números*, *Jugando con las letras*, *RoboMat*, etc., que contienen actividades de diferentes materias como Español, Ciencias Naturales, Matemáticas, etc. pensadas en los temarios actuales de la SEP y que incluso algunos de ellos tuvieron que adaptarse a las condiciones del hardware¹ de los laboratorios del colegio en las demás secciones que solamente contaban con máquinas de procesadores AT286 e incluso todavía XT8088.

Con todo esto es importante remarcar la importancia que tiene actualmente la computadora en el apoyo del aprendizaje del alumnado no solo en primaria ya que este se extiende a cualquier grado de educación.

Tomando en cuenta todas las ventajas y desventajas que esto acarrea como lo menciono en los capítulos anteriores, podemos darnos una idea de los beneficios que acarrea el usarla como una herramienta más en las aulas.

¹ **Hardware:** son los componentes físicos de la computadora.

Lo importante es crear prototipos e incluso partir de lo empírico, para que así, al perfeccionar estas técnicas la tendencia sea ir innovando otras herramientas de diseño de mayor utilidad y enfocarse directamente a los temas específicos y sin desviarse hacia otros tópicos que pueden ofrecer otras variedades de programas.

Todavía queda bastante por hacer pero creo que es necesario el ir aportando en grande o pequeña medida, el esfuerzo tanto en educadores como en programadores para que el software educativo alcance las expectativas deseadas.

ANEXO

I. CODIGOS DEL PROGRAMA "EL RELOJ"

MODULO PRINCIPAL

```
Private Sub MUSICA(archivo As String)
On Error GoTo error
  MMControl1.DeviceType = "WaveAudio"
  MMControl1.filename = archivo
  MMControl1.Command = "Open"
  MMControl1.Command = "Prev"
  MMControl1.Command = "play"
Exit Sub
error:
  'No abrir el controlador de multimedia cuando no este disponible
End Sub

Public Sub Mostrar_Objetos(C)
Dim j As Integer
Imagen.Visible = False : BotonFin.Visible = True
Select Case C
  Case 1
    Cuadro3.Visible = True
    LetreroG.Caption = ""
    LetreroG.Visible = True
    flecha(1).Visible = True
    flecha(2).Visible = True
  Case 2
    Cuadro3.Visible = False
    Cuadro2.Visible = True
    FContesta.Visible = True
  Case 3
    Cuadro3.Visible = True
    Cuadro2.Visible = True
    FContesta.Visible = False
    LetreroH.Visible = False
    LetreroM.Visible = False
    TextoM1.Visible = False
    TextoH1.Visible = False
    flecha(1).Visible = False
    flecha(2).Visible = False
End Select
End Sub
```

Funciones PuntoX y PuntoY

Public Function PuntoX(Angulo, Magnitud) As Double

PuntoX = Magnitud * Cos(Angulo / 57.3)

End Function

Public Function PuntoY(Angulo, Magnitud) As Double

PuntoY = Magnitud * Sin(Angulo / 57.3)

End Function

Public Sub ManecillaChica(AnguloCh, ColorLinea, Grosor)

Dibujando la manecilla chica (Horario)

Pintura.DrawWidth = Grosor

Pintura.ForeColor = QBColor(ColorLinea)

Pintura.Line (PuntoX(AnguloCh, -0.85), PuntoY(AnguloCh, -0.85))-(PuntoX(AnguloCh + 235, 0.25), PuntoY(AnguloCh + 235, 0.25))

Pintura.Line -(PuntoX(AnguloCh - 8, 1.8), PuntoY(AnguloCh - 8, 1.8))

Pintura.Line -(PuntoX(AnguloCh - 19, 1.9), PuntoY(AnguloCh - 19, 1.9))

Pintura.Line -(PuntoX(AnguloCh, 2.3), PuntoY(AnguloCh, 2.3))

Pintura.Line -(PuntoX(AnguloCh + 19, 1.9), PuntoY(AnguloCh + 19, 1.9))

Pintura.Line -(PuntoX(AnguloCh + 8, 1.8), PuntoY(AnguloCh + 8, 1.8))

Pintura.Line -(PuntoX(AnguloCh - 235, 0.25), PuntoY(AnguloCh - 235, 0.25))

Pintura.Line -(PuntoX(AnguloCh, -0.85), PuntoY(AnguloCh, -0.85))

End Sub

Public Sub ManecillaGrande(AnguloG, ColorLinea, Grosor)

Dibujando la manecilla grande (Minutero)

Pintura.DrawWidth = Grosor

Pintura.ForeColor = QBColor(ColorLinea)

Pintura.Line (PuntoX(AnguloG, -1.7), PuntoY(AnguloG, -1.7))-(PuntoX(AnguloG + 215, 0.5), PuntoY(AnguloG + 215, 0.5))

Pintura.Line -(PuntoX(AnguloG - 6, 3.6), PuntoY(AnguloG - 6, 3.6))

Pintura.Line -(PuntoX(AnguloG - 15, 3.8), PuntoY(AnguloG - 15, 3.8))

Pintura.Line -(PuntoX(AnguloG, 4.6), PuntoY(AnguloG, 4.6))

Pintura.Line -(PuntoX(AnguloG + 15, 3.8), PuntoY(AnguloG + 15, 3.8))

Pintura.Line -(PuntoX(AnguloG + 6, 3.6), PuntoY(AnguloG + 6, 3.6))

Pintura.Line -(PuntoX(AnguloG - 215, 0.5), PuntoY(AnguloG - 215, 0.5))

Pintura.Line -(PuntoX(AnguloG, -1.7), PuntoY(AnguloG, -1.7))

End Sub

Public Sub Comp_Coloca_Manecillas(C)

Hora = Int(12 * Rnd + 1)

Minuto = Int(59 * Rnd)

AnguloG = 90 - ((360 / 60) * Minuto)

AnguloCh = (90 - ((360 / 12) * Hora)) - (Minuto / 2)

Call Manecillas(AnguloG, AnguloCh)

```
LetreroH.Caption = Str$(Hora)
LetreroM.Caption = Str$(Minuto)
End Sub
```

```
Public Sub Manecillas(Angulo1, Angulo2)
Pintura.Cls
Call ImprimeNumeros
Call ManecillaGrande(Angulo1, 12, 8)
Call ManecillaGrande(Angulo1, 1, 2)
Call ManecillaChica(Angulo2, 12, 8)
Call ManecillaChica(Angulo2, 14, 2)
Pintura.Circle (0, 0), 0.25
End Sub
```

```
Public Sub Imprime_Numeros()
'Imprimiendo los números de las horas
Pintura.DrawWidth = 3
Angulo = 90
For j = 1 To 60
    Angulo = Angulo - (360 / 60)
    X1 = PuntoX(Angulo, 5)
    Y1 = PuntoY(Angulo, 5)
    Pintura.PSet (X1, Y1), QBColor(12)
Next j
Pintura.ForeColor = QBColor(5)
Pintura.FontSize = 18
Pintura.FontItalic = True
Pintura.FontBold = True
Pintura.FontName = "Comic Sans ms"
Angulo = 90
Pintura.DrawWidth = 8
For j = 1 To 12
    Angulo = Angulo - (360 / 12)
    X1 = PuntoX(Angulo, 5.7)
    Y1 = PuntoY(Angulo, 5.7)
    Pintura.CurrentX = X1 - 0.7
    Pintura.CurrentY = Y1 + 0.7
    Pintura.Print j
    X1 = PuntoX(Angulo, 5)
    Y1 = PuntoY(Angulo, 5)
    Pintura.PSet (X1, Y1), QBColor(12)
Next j
End Sub
```

```
Private Sub boton_Click(index As Integer)
Dim j As Integer
```

```

opcion = index
Contador = 1
If index <> 4 Then
    Call Mostrar_Objetos(index)
    Panel1.Visible = True
End If
Desactivar = False
Select Case index
Case 1
    Call musica("Reloj")
    LetreroG.Caption = teoria(Contador)
    Timer1.Enabled = True
Case 2
    Desactivar = True
    Aciertos = 0
    LetreroG.Visible = False
    Cuadro2.Visible = True
    Call Comp_Coloca_Manecillas(Contador)
    s = MsgBox(" Indica la hora señalada por el reloj ", 64, "Atención")
Case 3
    Desactivar = True
    Aciertos = 0
    LetreroG.Visible = False
    Cuadro2.Visible = True
    Call Comp_Indica_Hora(Contador)
    s = MsgBox(" Coloca las manecillas del reloj en la hora indicada abajo y cuando estes
    listo haz click en Califica ahora", 64, "Atención")
Case 4
    s = MsgBox("¿ Deseas salir del programa ?", 4, "Salida")
    If s = vbYes Then
        End
    End If
End Select
End Sub

Private Sub CaminaHora()
    Hora = Hora + 1
    If Hora > 12 Then Hora = 1
    AnguloCh = AnguloCh - (360 / 12)
    Minuto = 0
    'AnguloCh = AnguloCh - (360 / (60 * 12))
    Call Manecillas(AnguloG, AnguloCh)
End Sub

Public Sub CaminaMinuto(Direccion)
'Direccion=1 : Avanza; Direccion =2 : Retrocede

```

```
If Direccion = 1 Then
    AnguloG = AnguloG - (360 / 60)
    AnguloCh = AnguloCh - (360 / (60 * 12))
    Minuto = Minuto + 1
Else
    AnguloG = AnguloG + (360 / 60)
    AnguloCh = AnguloCh + (360 / (60 * 12))
    Minuto = Minuto - 1
End If
If Minuto = 60 Then
    Minuto = 0
    Hora = Hora + 1
    If Hora = 13 Then
        Hora = 1
    End If
End If
If Minuto = -1 Then
    Minuto = 59
    Hora = Hora - 1
    If Hora = 0 Then
        Hora = 12
    End If
End If
Call Manecillas(AnguloG, AnguloCh)
LetreroH.Caption = Format$(Hora, "00")
LetreroM.Caption = Format$(Minuto, "00")
End Sub

Private Sub Timer1_Timer()
    Call CaminaMinuto(1)
End Sub

Private Sub Timer2_Timer()
    k = k + 1
    If k < Tiempo Then
        Call CaminaMinuto(1)
    Else
        Timer2.Enabled = False
        TextoM1.Text = ""
        TextoM1.Visible = True
        TextoM1.SetFocus
    End If
End Sub

Private Sub Timer3_Timer()
    k = k + 1
```

```

If k < Tiempo Then
  Call CaminaHora(1)
Else
  Timer3.Enabled = False
  TextoH1.Text = ""
  TextoH1.Visible = True
  TextoH1.SetFocus
End If
End Sub

```

```

Private Sub Form_Load()
On Error GoTo SubError
'Valores iniciales del Controlador de multimedia
MMControl1.Notify = False
MMControl1.Wait = True
MMControl1.Shareable = False
MMControl1.DeviceType = "WaveAudio"
MMControl1.filename = "passport.mid"
MMControl1.Command = "Open"
MMControl1.Command = "play"
'Almacenamiento de información en el arreglo de variables
teoria(1) = "Máquina que sirve para medir el tiempo. Divide al día en HORAS, MINUTOS
y SEGUNDOS. Tiene una manecilla GRANDE que indica los minutos y una manecilla
CHICA que indica las horas."
  teoria(2) = "El MINUTERO o manecilla grande indica los minutos, lo hace de 5 en 5 y
cada vez que el minuterero llegue a un número habrán pasado 5 minutos."
  teoria(3) = "Observa la manecilla GRANDE avanza más rápido que la pequeña."
  teoria(4) = "Observa con atención como avanza el minuterero y en cuanto se detenga
indica cuantos minutos han pasado."
  teoria(5) = "La manecilla chica u HORARIO indica las horas, lo hace de 1 en 1 y cada
vez que la MANECILLA CHICA llegue al número, habrá pasado 1 HORA."
  teoria(6) = "Observa como la manecilla CHICA u HORARIO no apunta exactamente a
un número. Esto es porque el MINUTERO no está exactamente en el doce, ya que avanzó
unos minutos."
  teoria(7) = "Cuando la manecilla GRANDE ó MINUTERO esta exactamente en el doce
la manecilla CHICA u horario apunta precisamente a la hora indicada."
  teoria(8) = "Observa con atención como avanza el HORARIO y en cuanto se detenga
indica cuantas horas han pasado."
  teoria(9) = "Una hora tiene 60 minutos, por eso cada vuelta de la manecilla GRANDE o
MINUTERO, es igual a una HORA."
  teoria(10) = "Mira atentamente como avanza la manecilla CHICA, hasta que la
GRANDE llega al doce. En este preciso instante cambia la hora indicada."
  teoria(11) = "Cuando la manecilla GRANDE o MINUTERO está exactamente en el doce
la manecilla CHICA u HORARIO apunta precisamente a la hora indicada."
Exit Sub
SubError:

```

```
No inicializar controlador de multimedia cuando no esta disponible
End Sub
```

```
Private Sub Form_Activate()
Dim j, Angulo, X1, Y1 As Integer
Pintura.Scale (-6, 6)-(6, -6)
'Origen de las flechas
AnguloG = 90
AnguloCh = 90
Call Manecillas(AnguloG, AnguloCh)
'Iniciar a las 12
Minuto = 0
Hora = 12
LetreroH.Caption = Format$(Hora, "00")
LetreroM.Caption = Format$(Minuto, "00")
LetreroG.Visible = False
flecha(1).Visible = False
flecha(2).Visible = False
BotonFin.Visible = False
Cuadro2.Visible = False
Panel1.Visible = False
Panel3.Visible = False
VMensaje.Visible = False
TextoH1.Visible = False
TextoM1.Visible = False
End Sub
```

MODULO APRENDE

```
Private Sub TextoH1_GotFocus()
If Len(TextoH1.Text) > 0 Then
    TextoH1.SelStart = 0
    TextoH1.SelLength = Len(TextoH1.Text)
End If
End Sub

Private Sub TextoH1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    Respuesta = Val(TextoH1.Text)
    If Respuesta = (Tiempo - 1) Then
        Call mensaje("!!! Muy bien !!!", "Resultado")
    Else
        Call mensaje(" Error... ", "Resultado")
    End If
    TextoH1.Visible = False
    LetreroH.Visible = True
End Sub
```

```
End If
End Sub
```

```
Private Sub mensaje(m, l)
Panel1.Enabled = False
VMensaje.Visible = True
Label1.Caption = l
Label2.Caption = m
End Sub
```

```
Private Sub ControlHora_Click(index As Integer)
If index = 1 Then
    Hora = Hora + 1
    If Hora > 12 Then Hora = 1
    AnguloCh = AnguloCh - (360 / 12)
Else
    Hora = Hora - 1
    If Hora < 1 Then Hora = 12
    AnguloCh = AnguloCh + (360 / 12)
End If
Call CaminaMinuto(index)
End Sub
```

```
Private Sub ControlMin_Click(index As Integer)
    Call CaminaMinuto(index)
End Sub
```

```
Private Sub flecha_Click(index As Integer)
If index = 1 Then
    If Contador > 1 Then Contador = Contador - 1
Else
    If Contador < 12 Then Contador = Contador + 1
End If
Select Case opcion
Case 1
    Call Actividad(Contador)
Case 3
End Select
End Sub
```

```
Public Sub Actividad(C)
Dim j As Integer
Randomize
LetreroG.Caption = ""
LetreroG.Visible = True
Select Case C
```

Case 1 To 3

```
Timer1.Enabled = True  
LetreroG.Caption = teoria(Contador)
```

Case 4

```
LetreroG.Visible = False  
LetreroM.Visible = False  
Timer1.Enabled = False  
Timer2.Enabled = False  
AnguloCh = 90  
AnguloG = 90 + (360 / 60)  
Minuto = 59  
Hora = 11  
Call CaminaMinuto(1)  
Continua = False  
Call mensaje(teoria(C), "Atención")
```

Case 5 To 7, 9 To 11

```
Timer1.Enabled = True  
LetreroG.Caption = teoria(C)
```

Case 8

```
LetreroG.Visible = False  
LetreroM.Visible = False  
Timer1.Enabled = False  
Timer2.Enabled = False  
AnguloCh = 90  
AnguloG = 90 + (360 / 60)  
Minuto = 59  
Hora = 11  
Call CaminaMinuto(1)  
Continua = False  
Call mensaje(teoria(C), "Atención")
```

Case 12

```
Call mensaje("Espero que hayas aprendido bastante acerca del reloj", "Fin de la actividad")  
End Select  
End Sub
```

Private Sub BotonFin_Click()

```
Dim j As Integer  
LetreroG.Visible = False  
flecha(1).Visible = False  
flecha(2).Visible = False  
LetreroG.Visible = False  
Cuadro2.Visible = False  
LetreroH.Visible = True  
LetreroM.Visible = True  
TextoM1.Visible = False
```

```

BotonFin.Visible = False
Cuadro3.Visible = True
Imagen.Visible = True
Panel1.Visible = False
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
End Sub

```

MODULO BUSCAR LA HORA Y ESCRIBIR LA HORA

```

Private Sub BotonCalif_Click()
If Contador <= 10 Then
  Contador = Contador + 1
  Select Case opcion
  Case 3
  If Minuto = MinutoA And Hora = HoraA Then
    Aciertos = Aciertos + 1
    Call MUSICA("ding.wav")
    Call mensaje(" ¡¡¡ Perfecto !!! ", "Sigue adelante")
  Else
    Call mensaje(" Esa no es la respuesta correcta ", "Atención")
    Call MUSICA("Chord.wav")
  End If
  Call Comp_Indica_Hora(Contador)
  Case 2
  HoraA = Val(TextoH.Text)
  MinutoA = Val(TextoM2.Text)
  If Minuto = MinutoA And Hora = HoraA Then
    Aciertos = Aciertos + 1
    Call MUSICA("ding.wav")
    Call mensaje(" ¡¡¡ Perfecto !!! ", "Sigue adelante")
  Else
    Call MUSICA("Chord.wav")
    Call mensaje(" Esa no es la respuesta correcta ", "Atención")
  End If
  Call Comp_Coloca_Manecillas(Contador)
End Select
Else
  Panel3.Visible = True
  EtAciertos = Str$(Aciertos)
  EtErrores = Str$(10 - Aciertos)
  Cuadro2.Visible = False
  For j = 1 To 4
    boton(j).Visible = True
  Next j

```

```
End If
End Sub
```

Cuando el apuntador se coloca sobre la caja de texto *TextoH*, se selecciona el texto anterior para que cuando el usuario escriba la nueva hora, la hora escrita anteriormente se borre en forma automática.

```
Private Sub TextoH_GotFocus()
If Len(TextoH.Text) > 0 Then
    TextoH.SelStart = 0
    TextoH.SelLength = Len(TextoH.Text)
End If
End Sub
```

Al terminar de escribir la hora y presionar la tecla ENTER, automáticamente pasa el apuntador a la caja de textos *TextoM2*, para escribir los minutos indicados.

```
Private Sub TextoH_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    TextoM2.SetFocus
End If
End Sub
```

De la misma forma que en cualquier caja de texto, se implementó el código para seleccionar el dato escrito anteriormente y borrarlo en forma automática al escribir el siguiente dato.

```
Private Sub TextoM2_GotFocus()
If Len(TextoM2.Text) > 0 Then
    TextoM2.SelStart = 0
    TextoM2.SelLength = Len(TextoM2.Text)
End If
End Sub
```

II. CODIGOS DEL PROGRAMA "EL MAPA DE LA REPUBLICA MEXICANA"

MODULO PRINCIPAL

var

Ventana: TVentana;

type

```
RecordMap = Record
  Edo : array [1..31] of String;
  Cap : array [1..31] of String;
  Posx : array [1..31] of Integer;
  Posy : array [1..31] of Integer;
end;
```

procedure TVentana.FormCreate(Sender: TObject);

var

j : integer;

begin

NumEdo := 0;

Mex.Edo[1]:= 'Aguascalientes'; Mex.Cap[1]:= 'Aguascalientes'; Mex.PosX[1]:= 313;
Mex.PosY[1]:= 267;

Mex.Edo[2]:= 'Baja California Norte'; Mex.Cap[2]:= 'Mexicali'; Mex.PosX[2]:= 40;
Mex.PosY[2]:= 24;

Mex.Edo[3]:= 'Baja California Sur'; Mex.Cap[3]:= 'La Paz'; Mex.PosX[3]:= 136;
Mex.PosY[3]:= 218;

Mex.Edo[4]:= 'Campeche'; Mex.Cap[4]:= 'Campeche'; Mex.PosX[4]:= 584;
Mex.PosY[4]:= 312;

Mex.Edo[5]:= 'Chiapas'; Mex.Cap[5]:= 'Tuxtla Gutierrez'; Mex.PosX[5]:= 524;
Mex.PosY[5]:= 383;

Mex.Edo[6]:= 'Chihuahua'; Mex.Cap[6]:= 'Chihuahua'; Mex.PosX[6]:= 241;
Mex.PosY[6]:= 124;

Mex.Edo[7]:= 'Coahuila'; Mex.Cap[7]:= 'Saltillo'; Mex.PosX[7]:= 336; Mex.PosY[7]:= 191;

Mex.Edo[8]:= 'Colima'; Mex.Cap[8]:= 'Colima'; Mex.PosX[8]:= 274; Mex.PosY[8]:= 329;

Mex.Edo[9]:= 'Durango'; Mex.Cap[9]:= 'Durango'; Mex.PosX[9]:= 259;

Mex.PosY[9]:= 216;

Mex.Edo[10]:= 'Estado de México'; Mex.Cap[10]:= 'Toluca'; Mex.PosX[10]:= 370;
Mex.PosY[10]:= 328;

Mex.Edo[11]:= 'Guanajuato'; Mex.Cap[11]:= 'Guanajuato'; Mex.PosX[11]:= 336;
Mex.PosY[11]:= 287;

Mex.Edo[12]:= 'Guerrero'; Mex.Cap[12]:= 'Chilpancingo'; Mex.PosX[12]:= 378;
Mex.PosY[12]:= 367;

Mex.Edo[13]:= 'Hidalgo'; Mex.Cap[13]:= 'Pachuca'; Mex.PosX[13]:= 390;
Mex.PosY[13]:= 305;

Mex.Edo[14]:= 'Jalisco'; Mex.Cap[14]:= 'Guadalajara'; Mex.PosX[14]:= 286;
Mex.PosY[14]:= 296;

```

Mex.Edo[15]='Michoacan'; Mex.Cap[15]='Morelia';Mex.PosX[15]=333;
Mex.PosY[15]=322;
Mex.Edo[16]='Morelos'; Mex.Cap[16]='Cuernavaca';Mex.PosX[16]=382;
Mex.PosY[16]=336;
Mex.Edo[17]='Nayarit'; Mex.Cap[17]='Tepic';Mex.PosX[17]=253;
Mex.PosY[17]=274;
Mex.Edo[18]='Nuevo León'; Mex.Cap[18]='Monterrey';Mex.PosX[18]=360;
Mex.PosY[18]=184;
Mex.Edo[19]='Oaxaca'; Mx.Cap[19]='Oaxaca';Mex.PosX[19]=437;
Mex.PosY[19]=378;
Mex.Edo[20]='Puebla'; Mex.Cap[20]='Puebla';Mex.PosX[20]=398;
Mex.PosY[20]=335;
Mex.Edo[21]='Quintana Roo'; Mex.Cap[21]='Chetumal';Mex.PosX[21]=626;
Mex.PosY[21]=320;
Mex.Edo[22]='Queretaro'; Mex.Cap[22]='Queretaro';Mex.PosX[22]=360;
Mex.PosY[22]=296;
Mex.Edo[23]='San Luis Potosí'; Mex.Cap[23]='San Luis Potosí'; Mex.PosX[23]=345;
Mex.PosY[23]=263;
Mex.Edo[24]='Sinaloa'; Mex.Cap[24]='Culiacán';Mex.PosX[24]=195;
Mex.PosY[24]=196;
Mex.Edo[25]='Sonora'; Mex.Cap[25]='Hermosillo';Mex.PosX[25]=133;
Mex.PosY[25]=103;
Mex.Edo[26]='Tabasco'; Mex.Cap[26]='Villa Hermosa';Mex.PosX[26]=528;
Mex.PosY[26]=351;
Mex.Edo[27]='Tamaulipas'; Mex.Cap[27]='Ciudad Victoria';Mex.PosX[27]=383;
Mex.PosY[27]=229;
Mex.Edo[28]='Tlaxcala'; Mex.Cap[28]='Tlaxcala';Mex.PosX[28]=404;
Mex.PosY[28]=323;
Mex.Edo[29]='Veracruz'; Mex.Cap[29]='Jalapa';Mex.PosX[29]=434;
Mex.PosY[29]=321;
Mex.Edo[30]='Yucatán'; Mex.Cap[30]='Merida';Mex.PosX[30]=597;
Mex.PosY[30]=285;
Mex.Edo[31]='Zacatecas'; Mex.Cap[31]='Zacatecas';Mex.PosX[31]=302;
Mex.PosY[31]=249;
Panel3.Visible:=False;
FrameDatos.Visible:=False;
FrameLocaliza.Visible:=False;
for j:=1 to 31 do
  Iluminar_Estado(j,1);
Mapa.Enabled:=False;
end;

```

```

Procedure Iluminar_Estado(NumEdo, ColorEdo : Integer);
begin
  Case ColorEdo of
    1 : Ventana.Mapa.Canvas.Brush.Color := clscrollBar;

```

```
2 : Ventana.Mapa.Canvas.Brush.Color := clTeal;  
3 : Ventana.Mapa.Canvas.Brush.Color := clMaroon;  
end;
```

```
Ventana.Mapa.Canvas.Floodfill(Mex.PosX[NumEdo],Mex.PosY[NumEdo],clBlack,fsBorder);  
Ventana.Mapa.Canvas.Pen.color:=clPurple;  
Ventana.Mapa.Canvas.Pen.Width :=2;  
Ventana.Mapa.Canvas.Brush.Color:= clRed;  
Ventana.Mapa.Canvas.Ellipse(Mex.PosX[NumEdo]-5,Mex.PosY[NumEdo]-5,Mex.PosX[NumEdo]+2,Mex.PosY[NumEdo]+2);  
Ventana.EtEstado.Caption := Mex.Edo[NumEdo];  
Ventana.EtCapital.Caption := Mex.Cap[NumEdo];  
end;
```

```
procedure TVentana.Boton1Click(Sender: TObject);  
begin  
Opcion := 1;  
Boton4.Enabled:=False;  
FrameDatos.Visible:=True;  
Panel2.Visible:=False;  
Panel3.Visible:=True;  
NumEdo:=0;  
Conoce(1);  
end;
```

```
procedure TVentana.Boton2Click(Sender: TObject);  
begin  
Opcion :=2 ;  
{Inicializando los datos}  
FrameResultados.left:=24;  
FrameResultados.Top:=248;  
FrameResultados.Visible:=True;  
Label3.Caption:='0';  
Label4.Caption:='0';  
Aviso:=False;  
Acertos:=0;  
BitBtn1.Enabled:=False;  
Boton4.Enabled:=False;  
Panel2.Visible:=False;  
Panel3.Visible:=True;  
NumEdo:=0;  
Localiza(1);  
end;
```

```

procedure TVentana.Boton3Click(Sender: TObject);
begin
  Opcion :=3;
  {Inicializando los datos}
  FrameResultados.Left:=440;
  FrameResultados.Top:=40;
  FrameResultados.Visible:=True;
  FrameEscribe.Visible:=True;
  Label3.Caption:='0';
  Label4.Caption:='0';
  Aviso:=False;
  Aciertos:=0;
  BitBtn1.Enabled:=False;
  Boton4.Enabled:=False;
  Panel2.Visible:=False;
  Panel3.Visible:=True;
  NumEdo:=0;
  Escribe(1);
end;

```

```

procedure TVentana.Boton4Click(Sender: TObject);
begin
  if MessageDlg('¿ Realmente deseas salir ?',
    mtConfirmation, [mbYes, mbNo], 0) = mrYes then
  begin
    MessageDlg('Saliendo del programa de La República Mexicana', mtInformation,
      [mbOk], 0);
    Close;
  end;
end;

```

MODULO APRENDE

```

procedure TVentana.BitBtn1Click(Sender: TObject);
begin
  Case opcion of
    1: Conoce(-1);
    2: Localiza(-1);
  end;
end;

```

```

Procedure Conoce(Incr : integer);
begin
  Intervalo(Incr);
  Limpiar_Mapas(1);
  Iluminar_estado(NumEdo,1); end;

```

```
Procedure Intervalo(Incr : Integer);
begin
  if Incr = -1 then
    begin
      if NumEdo > 1 then
        NumEdo := NumEdo + Incr;
      end;
    If Incr = 1 then
      begin
        if NumEdo < 31 then
          NumEdo := NumEdo + Incr;
        end;
      end;
    end;
```

```
Procedure Limpiar_Mapa(i:integer);
begin
  Ventana.Mapa.Canvas.Brush.Color := clWhite;
  for i:=1 to 31 do
    begin
      Ventana.Mapa.Canvas.Floodfill(Mex.PosX[i]-5,Mex.PosY[i]-5,clBlack,fsBorder);
    end;
  end;
```

```
procedure TVentana.BitBtn2Click(Sender: TObject);
begin
  Case opcion of
    1: Conoce(1);
    2: begin
        if Aviso=True then
          begin
            FrameDatos.Visible:=False;
            Localiza(1);
            Aviso:=False;
          end;
        end;
    3: begin
        if Aviso=True then
          begin
            Edit1.Text:="";
            FrameEscribe.Visible:=False;
            Escribe(1);
            Aviso:=False;
          end;
        end;
    end;
  end;
```

```

procedure TVentana.BitBtn3Click(Sender: TObject);
begin
  Limpiar_Mapa(1);
  FrameEscribe.Visible:=False;
  FrameResultados.Visible:=False;
  FrameDatos.Visible:=False;
  Boton4.Enabled:=True;
  Panel3.Visible:=False;
  Panel2.Visible:=True;
  FrameLocaliza.Visible:=False;
  BitBtn1.Enabled:=True;
  edit1.text:="";
end;

```

MODULO LOCALIZAR

```

Function Aleatorio(i : Integer):Integer;
var
  Repetido : Boolean;
  j : Integer;
begin
  Randomize;
  Repeat
    Repetido := False;
    REdo[j]:=Random(31)+1;
    for j:=1 to i-1 do
      if (REdo[j]=REdo[j]) then
        Repetido := True;
    Until (Repetido = False);
  Aleatorio := REdo[j];
end;

Procedure Localiza(Incr : Integer);
var
  j : Integer;
begin
  Ventana.Mapa.Enabled:=true;
  Intervalo(Incr);
  Limpiar_Mapa(1);
  for j:=1 to 31 do
    Iluminar_Estado(j,1);
  Turno := Aleatorio(NumEdo);
  Ventana.EtEstado2.Caption := Mex.Edo[Turno];
  Ventana.FrameLocaliza.Visible:=True;
end;

```

```

procedure TVentana.MapaMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  if Aviso=False then
  begin
    if Button = mbleft then
    begin
      FrameLocaliza.Visible := False;
      if (x <= Mex.PosX[Turno]+10) and (x >= Mex.PosX[Turno]-10) and
        (y <= Mex.PosY[Turno]+10) and (y >= Mex.PosY[Turno]-10) then
      begin
        FrameDatos.Visible := True;
        {Mapa.canvas.brush.color := clRed;
        Mapa.canvas.FloodFill(Mex.PosX[Turno]-7,Mex.PosY[Turno]-7,clBlack,fsBorder);}
        Iluminar_Estado(Turno,3);
        Aciertos:=Aciertos+1;
      end
    else
    begin
      MessageDlg('¡Lo siento!, esa no es la capital...', mtError,[mbOk],0);
    end;
    Aviso:=True;
    Label3.Caption:=IntToStr(Aciertos);
    Label4.Caption:=IntToStr(NumEdo-Aciertos);
  end;
end;
end;
end;

```

MODULO ESCRIBE LA CAPITAL

```

Procedure Escribe(Incr : Integer);
var
  j : Integer;
begin
  Ventana.Mapa.Enabled:=True;
  Intervalo(Incr);
  Limpiar_Mapa(1);
  for j:=1 to 31 do
    Iluminar_Estado(j,1);
    Turno := Aleatorio(NumEdo);
    Iluminar_Estado(Turno,2);
    Ventana.EscEdo.Caption := Mex.Edo[Turno];
    Ventana.FrameEscribe.Visible:=True;
    Ventana.Mapa.Enabled:=False;
  end;
end;

```

```
procedure TVentana.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if Aviso=False then
  begin
    if Key = char(13) then
    begin
      if Uppercase(Edit1.Text)=UpperCase(Mex.Cap[Turno]) then
        Aciertos:=Aciertos+1
      else
        MessageDlg('¡Lo siento!, esa no es la capital...', mtError,[mbOk],0);
        Aviso:=True;
        Label3.Caption:=IntToStr(Aciertos);
        Label4.Caption:=IntToStr(NumEdo-Aciertos);
        FrameEscribe.Visible:=False;
      end;
    end;
  end;
```

III. ESTABLECIMIENTOS VISITADOS PARA EL ESTUDIO DE MERCADO

BODEGA DEL SOFTWARE, av. Revolución 1134.

CIENCIA EN VIDEO, Manuel Doblado 623 col. Las Fuentes.

COMPUDABO, (Portales y Rojo Gómez).

CYC EDUCARE, vía Adolfo López Mateos, Naucaipan.

EL PALACIO DE HIERRO (Durango, Polanco, Perisur, Santa Fé, Coyoacán, Centro).

FLIPPER, Circuito Juristas, Satélite.

GABSOL, Pensilvania 91, Local 1, Col. Nápoles.

GRUPO ASAE PROESA, av. Cuahutémoc 698 col. Narvarte.

LA CASA DE COMPUTO, Av. Patriotismo 110, San Pedro De Los Pinos.

LIBRERÍA AVALON, Av. Vasco De Quiroga 3800, Local 306, Sta. Fé.

LIBRERÍA COLORINES (Col. Coapa Y Condesa)

MICROEQUIPOS ADMINISTRATIVOS, Circuito Metalurgistas N.2 Local H-3 Satélite.

MR. LASER MÉXICO (Polanco, Centro, Universidad).

MULTICOM, Torcuato Tarso 245, Polanco.

SEICA COMPUTERS, Plaza Girasoles, Calzada Del Hueso 465.

SUPER BBS COMPUTADORAS (Plaza Comercial Galerías, Plaza Comercial Olimpos).

SUPER SOFTWARE, Polanco.

VAN-VEL, Av. Lindavista 251-504, Col. Lindavista.

BIBLIOGRAFIA

Luis Joyanes Aguilar, 1993, **Fundamentos de programación, algoritmos y estructuras de datos**, Editorial Mc-Graw-Hill.

Roger S. Pressman, 1992, **Ingeniería de software, un enfoque práctico**, Editorial Mc-Graw-Hill.

David P. Ausubel, 1991, **Psicología educativa**, Editorial Trillas.

Boyd Westfall Stash, 1986, **Investigación de mercados**, Editorial UTEHA.

Ceballos Fco. Manuel, 1993, **Microsoft Visual Basic aplicaciones para Windows**, Adisson-Wesley Iberoamericana.

Cornell Gary, 1994, **Manual de Visual basic 3 para Windows**, Mc-Graw-Hill.

Maya Martinez Joél, 1998, **Delphi 1**, Editorial Libra.

Michelle M. Manning, 1996, **Delphi 2 guía oficial de Borland**, Prentice Hall.

Larry Long, 1995, **Introducción a las computadoras y al procesamiento de la información**, Prentice Hall.