

UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS

FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN

LICENCIATURA EN CIENCIA DE LA COMPUTACIÓN



TRABAJO DE DIPLOMA

HERRAMIENTA DE AYUDA AL APRENDIZAJE DE LA MATEMÁTICA PARA LOS ESTUDIANTES DE TERCER GRADO

Autora: Anaisi Santos Carrazana

Tutores: Dr. Mateo Gerónimo Lezcano Brito, MSc. Graciela Carrazana
Quintero

2013-2014



Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Laboratorio

A toda mi familia por su apoyo brindado en todos estos años, en especial a mis padres, a mi abuela y a mi hermana.

A mi tutor Mateo por sus asesorías y orientaciones en este trabajo.

*A toda mi familia: a mis padres, a mi abuela y a mi hermana; por ser los
motores impulsores de todos mis logros.*

A mi amigo Pedro por ayudarme tanto en la tesis.

A los profesores del grupo Informática Educativa.

A todos los profesores que me formaron y enseñaron.

*A todos los que de una u otra forma me han ayudado y han contribuido a la
terminación de este trabajo.*

RESUMEN

En el presente trabajo se abordan algunas ideas sobre las Tecnologías de la Información y las Comunicaciones (TIC), se enfatiza en su empleo como apoyo del proceso docente educativo cubano. Se destacan los principales elementos relacionados con la Enseñanza Asistida por Computadora (EAC) y su incidencia sobre el proceso de enseñanza aprendizaje.

Se presenta un sistema, basado en tecnologías informáticas, concebido como una herramienta de apoyo a la enseñanza de la Matemática en el Tercer Grado de la Escuela Primaria Cubana. La materia objeto de estudio se caracteriza por tratar temas significativos, que establecen las bases para contenidos impartidos en cursos posteriores, lo que justifica la realización del trabajo. La nueva herramienta ofrece a los usuarios dos modos de interactuar: uno dirigido a los estudiantes que cursan la asignatura y otro al profesor que prepara los cursos.

“La Isla de las Matemáticas” persigue dentro de sus objetivos apoyar el proceso de enseñanza aprendizaje de la Matemática de Tercer Grado del Sistema Educativo Cubano, complementar el contenido del libro de texto y el cuaderno de trabajo y fomentar el empleo de la computadora como un medio de enseñanza virtual desde las edades más tempranas, así como facilitar la comprensión de los contenidos impartidos en este grado.

ABSTRACT

In this paper some ideas on Information Technology and Communication (ICT) are addressed, emphasized in his job as supporting the Cuban education system. Highlights the main elements related to Computer Aided Instruction (EAC) and its impact on the teaching-learning process.

A system based on information technology, conceived as a tool to support the teaching of Mathematics in the Third Degree of the Cuban Elementary School is presented. The subject under study is characterized by addressing significant issues, which lay the foundation for subsequent content taught in courses, which justifies the performance of work. The new tool gives users two modes of interaction: one aimed at students taking the course and the other teachers.

"The Island of Mathematics" pursues its objectives in supporting the teaching and learning of mathematics in Third Grade Cuban Educational System, supplement the content of the textbook and workbook and encourage the use of the computer as a through virtual education from the earliest ages, as well as facilitate understanding of the content taught in this degree.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO I. LA ENSEÑANZA ASISTIDA POR COMPUTADORA	5
I.1 OBJETIVOS DEL CAPÍTULO	5
I.2 EL EMPLEO DE LAS TIC EN EL PROCESO DOCENTE EDUCATIVO CUBANO	5
I.3 EVOLUCIÓN DE LA ENSEÑANZA ASISTIDA POR COMPUTADORA (EAC).....	9
I.4 LA ENSEÑANZA DE LA MATEMÁTICA EN LA ESCUELA CUBANA	12
I.5 AUTÓMATAS, LENGUAJES Y GRAMÁTICAS.....	14
I.6 TRADUCTORES, COMPILADORES E INTÉRPRETES.....	18
I.6.1 <i>Análisis lexicológico (scanner)</i>	20
I.6.2 <i>Análisis sintáctico (parsing)</i>	20
I.6.3 <i>Análisis semántico</i>	22
I.6.4 <i>Generación de código intermedio</i>	22
I.6.5 <i>Optimización de código</i>	23
I.6.6 <i>Generación de código</i>	23
I.6.7 <i>Tabla de símbolos</i>	23
I.6.8 <i>Análisis de errores</i>	24
I.7 PATRÓN ARQUITECTÓNICO MODELO-VISTA-CONTROLADOR	24
I.7.1 <i>Funcionamiento del MVC</i>	28
I.8 CONCLUSIONES DEL CAPÍTULO	28
CAPÍTULO II. “LA ISLA DE LAS MATEMÁTICAS”.VISIÓN INTERNA	30
II.1 OBJETIVOS DEL CAPÍTULO.....	30
II.2 MÓDULOS QUE COMPONEN EL SISTEMA.....	30
II.3 ACTORES Y CASOS DE USO	30
II.4 ARQUITECTURA DEL SOFTWARE	32
II.4.1 <i>Utilización de recursos gráficos y de audio</i>	36
II.5 EL INTÉRPRETE	48
II.5.1 <i>Análisis lexicológico (scanner)</i>	48
II.5.2 <i>El análisis sintáctico (parser)</i>	49
II.5.3 <i>Tratamiento de los errores</i>	51
II.5.4 <i>La Tabla de Símbolos</i>	51
II.6 CONCLUSIONES DEL CAPÍTULO.....	52
CAPÍTULO III. APLICACIÓN DOCENTE DEL SOFTWARE	53
III.1 OBJETIVOS DEL CAPÍTULO.....	53
III.2 CARACTERIZACIÓN DE LA ASIGNATURA DE MATEMÁTICA IMPARTIDA AL TERCER GRADO	53
III.2.1 <i>“La Isla de las Matemáticas” en el proceso docente de la asignatura Matemática impartida en el Tercer Grado</i>	56
III.3 “LA ISLA DE LAS MATEMÁTICAS” COMO MEDIO DE ENSEÑANZA	57
III.4 CARACTERÍSTICAS DEL SOFTWARE “LA ISLA DE LAS MATEMÁTICAS”	58
III.5 INDICACIONES PARA EL USO DOCENTE DE LA APLICACIÓN	60
III.6 COMPARACIÓN ENTRE “LA ISLA DE LAS MATEMÁTICAS” Y “LA FERIA DE LAS MATEMÁTICAS”	60

III.7	APORTES DEL SOFTWARE “LA ISLA DE LAS MATEMÁTICAS”	62
III.8	CONCLUSIONES DEL CAPÍTULO.....	63
CONCLUSIONES		64
RECOMENDACIONES		65
REFERENCIAS BIBLIOGRÁFICAS		66

LISTA DE FIGURAS

FIGURA I.1 RELACIÓN ENTRE LOS CONCEPTOS LENGUAJE, GRAMÁTICA Y AUTÓMATA.....	15
FIGURA I.2 FASES DE UN COMPILADOR.....	19
FIGURA I.3 ÁRBOL SINTÁCTICO PARA $X:=A+B*C$	20
FIGURA I.4 PATRÓN ARQUITECTÓNICO MODELO-VISTA-CONTROLADOR.....	26
FIGURA II.1 CASOS DE USO Y ACTORES DEL SISTEMA “LA ISLA DE LAS MATEMÁTICAS”	31
FIGURA II.2 ESTRUCTURA DE LOS PAQUETES DEL SISTEMA “LA ISLA DE LAS MATEMÁTICAS”	33
FIGURA II.3 DIAGRAMA DE PAQUETES.....	33
FIGURA II.4 DIAGRAMA DE CLASES DEL PAQUETE VISTA	34
FIGURA II.5 DIAGRAMA DE CLASES DEL PAQUETE ARCHIVOS	35
FIGURA II.6 DIAGRAMA DE CLASES DEL PAQUETE CONTROLADOR.....	36
FIGURA II.7 TRANSICIÓN DE ESTADOS DE UN HILO	40
FIGURA II.8 CLASE QUE REALIZA PROCESAMIENTO DE IMÁGENES.....	44
FIGURA II.9 VALORES DIFERENTES DE ALFA APLICADOS AL RECTÁNGULO	45
FIGURA II.10 MUESTRA DE AUDIO	46
FIGURA II.11 PRECEDENCIA DE LOS OPERADORES ARITMÉTICOS	50
FIGURA II.12 TRATAMIENTO DE ERRORES.....	51

INTRODUCCIÓN

Cuando se da una mirada al pasado es fácil apreciar que la obra lograda por la humanidad, ha sido producto de la inteligencia, la creatividad y la voluntad colectiva.

Como parte de esta obra, se puede resaltar el auge de las tecnologías de la información y la comunicación (TIC), las cuales desempeñan un papel decisivo en las diferentes esferas de la sociedad a escala mundial.

El impacto social de las TIC toca muy de cerca el proceso educativo cubano como forma de apoyo y soporte material al sistema de clases. El empleo de estas tecnologías da lugar a nuevas posibilidades al ampliar el espectro de los métodos tradicionales de enseñanza, no sustituyéndolos, sino enriqueciéndolos y complementándolos.

El uso de las TIC obliga a modificar la organización del proceso de enseñanza aprendizaje, porque crean entornos educativos que amplían considerablemente las posibilidades del sistema educativo, no sólo desde el punto de vista estructural, sino también de transmisión de conocimientos y desarrollo de destrezas, habilidades y actitudes. La clave para lograr, un proceso docente educativo eficiente se basa en la transformación de la información en conocimiento y este, en educación y aprendizaje significativo (Lezcano and Soler, 2009).

Por otra parte la Enseñanza Asistida por Computadora (EAC) trata de complementar lo que con otros medios de enseñanza no es posible o es difícil de transmitir a los educandos, es por ello que con el de cursar de los años su empleo aumenta cada día más, teniendo presente, claro está, que con la aplicación de la EAC no se intenta reemplazar la acción de otros medios los cuales han demostrado ampliamente su eficacia. Los profesores cubanos conscientes de las potencialidades y ventajas de la EAC, intentan utilizar diversos software como material auxiliar dentro de la propia clase o como herramienta de trabajo independiente, logrando así una mayor motivación por aprender.

Muchas expectativas se han creado alrededor del uso de la computadora y más recientemente, del aprovechamiento de la informática en la educación. Sin embargo, sólo algunas de ellas se han cumplido a cabalidad y las más importantes, están apenas en vías de atenderse (Panqueva and Alvaro, 1994).

Antecedentes

La escuela cubana puede disponer de numerosos software educativos para usarlos en diferentes esferas del proceso de enseñanza-aprendizaje. Algunos se han hecho bajo principios pedagógicos bien fundamentados y otros no.

Los software elaborados utilizan diversas herramientas informáticas que posibilitan el desarrollo de habilidades intelectuales generales (observación, comparación, clasificación, valoración). Algunos de ellos merecen destacarse por el nivel alcanzado, tales son los casos de las colecciones: “A Jugar”, “Multisaber”, “El Navegante” y “Futuro”.

A pesar del sitio destacado que han logrado las colecciones mencionadas, se deben citar algunas carencias que están presentes en la colección “Multisaber”, específicamente en el software “La Feria de las Matemáticas” las cuales, en cierto sentido, limitan su empleo:

- Posee una cantidad fija de ejercicios que no pueden enriquecerse con el intelecto de los profesores y por tanto no es posible ajustarlos a las necesidades diagnosticadas en el grupo con el que se trabaja.
- Una vez que se ha seleccionado un tema determinado, por ejemplo numeración, no es posible remitir al estudiante a un ejercicio específico que se ajuste a los errores detectados, ya que los problemas matemáticos que conforman el software se presentan como una secuencia.
- Debido a su naturaleza estática, se caracterizan por no incluir los ajustes curriculares que se van realizando en el desarrollo de los cursos escolares.

Reconociendo que existen varias colecciones de software educativos cubanos con los cuales se ha intentado cumplir los objetivos trazados en los planes de estudio con relación a la Matemática de Tercer Grado, surgen las siguientes **preguntas de investigación**:

¿Cómo especificar un lenguaje formal, dirigido al profesor, que permita expresar los ejercicios que complementan la aplicación y su vía de solución?

¿Será posible implementar un intérprete que se ajuste a las especificaciones del lenguaje formal establecido?

¿Qué aspectos se tendrán en cuenta para lograr un mejor diseño del sistema mediante la utilización del patrón arquitectónico Modelo-Vista-Controlador?

¿Logrará un nuevo software educativo, de apoyo a la enseñanza de la Matemática impartida al Tercer Grado de la Enseñanza Primaria, superar las deficiencias detectadas al software analizado?

Objetivo General

Implementar un software educativo para apoyar el proceso de enseñanza aprendizaje de la Matemática de Tercer Grado del Sistema Educacional Cubano.

Objetivos Específicos

1. Especificar un lenguaje formal, dirigido al profesor, que permita expresar los ejercicios que complementan la aplicación y su vía de solución.
2. Implementar un intérprete que se ajuste a las especificaciones del lenguaje formal establecido.
3. Diseñar el sistema empleando el patrón de arquitectura de software Modelo-Vista-Controlador.

Justificación

La enseñanza de la Matemática en el Tercer Grado, da inicio a una nueva etapa de exigencias en la formación general. Es importante, entonces lograr que los alumnos realicen suficientes actividades que le permitan transitar por los diferentes niveles del desarrollo cognitivo y asimilar de manera correcta los nuevos contenidos que serán impartidos.

La implementación del nuevo software educativo se traza múltiples propósitos: complementar el contenido del libro de texto y el cuaderno de trabajo, fomentar el empleo de la computadora como un medio de enseñanza virtual desde las edades más tempranas y facilitar la comprensión de los contenidos impartidos en este grado. El nuevo software podrá emplearse como material auxiliar en la clase, las tareas extraclases y en el trabajo individual.

Entre las razones que justifican la realización de este proyecto se puede mencionar el objetivo que se traza de lograr una mayor motivación por la Matemática, con ese fin se

incluirán ejercicios atractivos agrupados por los complejos de materia (tipos de ejercicios): numeración, cálculo (adición, sustracción, multiplicación y división) y fracciones que permitirán la interacción estudiante-computadora, mejorando la preparación de los alumnos.

Por otro lado el nuevo software parte de un análisis previo de carencias detectadas al software “La Feria de las Matemáticas” y persigue la meta de resolverlas, potenciando la adquisición del conocimiento de los educandos.

Estructura de la Tesis

El trabajo, está estructurado en tres capítulos. El Capítulo I “La Enseñanza Asistida por Computadora”, hace un análisis acerca de la instrucción, basada en el desarrollo de aplicaciones informáticas. Además se aborda, el empleo de las TIC en el proceso docente educativo cubano y se referencian los elementos primordiales asociados a la asignatura Matemática, impartida en el Tercer Grado de la Enseñanza Primaria.

El Capítulo II “La Isla de las Matemáticas. Visión Interna” describe el software desde el punto de vista de la implementación, destacando los paquetes y las clases principales que se utilizaron para su construcción.

El Capítulo III “Aplicación docente del software” abarca un conjunto de indicaciones, según el estudio con especialistas, para la puesta en práctica del sistema en el proceso docente de la asignatura de Matemáticas impartida a los estudiantes de Tercer Grado de la Enseñanza Primaria.

El trabajo finaliza con un conjunto de conclusiones que sirven como medida del logro de los objetivos y algunas recomendaciones que permiten su continuación y perfeccionamiento.

CAPÍTULO I. LA ENSEÑANZA ASISTIDA POR COMPUTADORA

I.1 Objetivos del capítulo

En el presente capítulo se aborda el tema de las TICs enfatizando en su empleo en el proceso docente educativo cubano, de manera semejante se destacan los principales elementos relacionados con la EAC y la tendencia histórica desde su propio surgimiento hasta la actualidad. Conjuntamente se describen algunos conceptos vinculados con el patrón arquitectónico Modelo-Vista-Controlador (MVC) y los compiladores debido a que la solución computacional que se le dará al problema planteado utiliza estas técnicas. Además se especifican aspectos importantes que se tienen en cuenta durante la enseñanza de la Matemática a los estudiantes de Tercer Grado.

I.2 El empleo de las TIC en el proceso docente educativo cubano

El crecimiento y expansión de las nuevas Tecnologías de la Información y las Comunicaciones, (TIC) han transformado el significado de la formación académica de una manera radical lo que ha incidido, de forma directa y positiva, en la evolución de la enseñanza tradicional.

A partir de las investigaciones originadas en los años sesenta, los procesos de comunicación global se modifican substancialmente, la computadora pasa de ser un instrumento de cálculo, a convertirse en productora, medio de almacenamiento y distribuidora de cualquier tipo de información digital. La expansión global de Internet posibilita el aprendizaje a distancia apoyado por tecnología, el cual se plantea como una innovación sin precedentes en la historia de la educación (Herrera and Mendoza, 2006).

El progreso de la ciencia y el cúmulo de conocimientos hacen inevitable que se desarrollen nuevos métodos de enseñanza, que se apoyen en las TIC para lograr universalizar la transmisión del conocimiento de forma que se atiendan las demandas educativas insatisfechas por la educación convencional.

Las tecnologías, como una de las expresiones más transformadoras y completas alcanzadas por las ciencias técnicas, se compone de medios, procedimientos y recursos humanos

preparados en el uso de esos medios y en el dominio de esos procedimientos, y se diseña, generalmente, a partir de las características del proceso dentro del cual van a emplearse (Zilberstein Toruncha et al., 2006).

Las llamadas tecnologías de la Información y las Comunicaciones son el resultado obtenido a partir de las posibilidades creadas por la humanidad en torno a la digitalización de datos, productos, servicios y procesos.

Las tecnologías de la comunicación y la información han existido desde la década de los sesenta del siglo pasado. Antes de esa época, el enfoque que se tenía acerca de la computadora era puramente científico y por lo tanto, no se empleaban en otras disciplinas. Sin embargo, se hacían investigaciones acerca de la integración de las telecomunicaciones y la informática, lo cual constituía un avance.

Posteriormente, en los setenta, surge una tendencia a crear las llamadas Computer Graphics (gráficas computarizadas) y se realizaron investigaciones y proyectos de realidad virtual y video llamadas, con el fin de integrar la funcionalidad del teléfono y de la televisión en centrales telefónicas. En la década de los ochenta, se desarrollaron las primeras investigaciones de circuitos integrados que lograron avances considerables en la velocidad de los cálculos y la capacidad de almacenamiento masivo. Se puede decir que todo esto fue un preámbulo, a lo que se conoce hoy como Internet, la red de comunicación más grande que existe a nivel mundial. Estos desarrollos potenciaron lo que, en aquella época, fue llamado “Nuevas Tecnologías de la Información y las Comunicaciones”, herramientas que combinan las aplicaciones informáticas con los recursos de comunicación. La ventaja de su utilización está en la renovación de conceptos relacionados con el espacio y tiempo, en la comunicación y en el aprovechamiento de las posibilidades tecnológicas.

El desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs), es uno de los elementos más distintivos de la sociedad actual y ha provocado una explosión de la cantidad de información disponible, brindando la posibilidad de interacción en el proceso de comunicación. Estas tecnologías se han incorporado en los más disímiles campos, entre ellos en la enseñanza, incidiendo en la aparición de nuevos roles en las instituciones

educativas, para docentes y estudiantes, así como nuevos materiales de enseñanza aprendizaje sobre distintos soportes (Vidal Ledo et al., 2009).

La TIC, es uno de los principales pilares de la comunicación universal en la actualidad. A través de ella se puede acceder a informaciones actuales casi en tiempo real. Todo ello ha influido en la educación, tanto a nivel escolar como familiar, implicando cambios metodológicos en el proceso docente educativo. Estos cambios se deben a que la escuela no debe estar al margen de la sociedad, no puede dejar de lado la TIC, sino que debe formar las nuevas generaciones capacitadas para el empleo de estos medios, es por ello que no se logra concebir la educación apartada de los medios de comunicación.

Cuba, no está ajena a toda esta problemática y el gobierno revolucionario ha trabajado incesantemente en el desarrollo de la Informática Educativa, por ejemplo en una fecha tan temprana como marzo de 1962 el Che avizoraba (Borrego Díaz, 2001): “El mundo camina hacia la era electrónica .. Todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo, quien la domine será un país de vanguardia. Vamos a volcar nuestros esfuerzos en este sentido con audacia revolucionaria”

Grandes esfuerzos se han realizado en estos años para preparar el personal que pueda asumir estos retos e incorpore la tecnología en la educación cubana, como apoyo al proceso de enseñanza aprendizaje desde las edades más tempranas. Como es natural el uso de los nuevos medios ha requerido la realización de estudios acerca del sistema educacional con la consiguiente modificación de los enfoques sobre los procesos cognitivos de los escolares.

Hoy las tecnologías son un complemento en el trabajo del maestro, al brindarle conocimientos sobre determinados contenidos que facilitan el aprendizaje y sirven de soporte material al sistema de clases y a numerosas funciones que tanto el maestro como la escuela deben desarrollar, dándole al escolar el papel protagónico que le corresponde en un nuevo entorno de enseñanza que es más participativo y permite un desarrollo superior de los procesos cognitivos y el interés por el estudio.

Las TICs se están convirtiendo en un elemento clave en el sistema educativo cubano y su incorporación en las aulas posibilita nuevas formas de acceder, generar, transmitir

información y conocimientos, a la vez que permiten flexibilizar el tiempo y el espacio en el que se desarrolla la acción educativa, implicando el uso de estrategias y metodologías docentes nuevas para lograr una enseñanza activa, participativa y constructiva (Moya, 2009).

El empleo de la TIC en el proceso de enseñanza aprendizaje ofrece múltiples ventajas, que se pueden ver desde dos perspectivas (Pérez Marqués, 2005):

Perspectiva del aprendizaje

- Interés y motivación. Motiva a los alumnos y la motivación es uno de los motores del aprendizaje, ya que incita a la actividad y al pensamiento. Logra que los estudiantes dediquen más tiempo a trabajar y, por tanto, es probable que aprendan más.
- Desarrollo de la iniciativa. La necesidad continuada de tomar decisiones ante los planteamientos del software incita la iniciativa. Se promueve un trabajo autónomo riguroso y metódico.
- Aprendizaje a partir de los errores. La retroalimentación (feed back) inmediata a las respuestas y a las acciones de los estudiantes permite que se conozcan los errores, justo en el momento en que se producen, y generalmente el programa ofrece la oportunidad de ensayar nuevas respuestas o formas de actuar para superarlos.
- Aprendizaje cooperativo. Los instrumentos que proporcionan las TIC (fuentes de información, materiales interactivos, correo electrónico, espacio compartido de disco, foros...), facilitan el trabajo en grupo y el cultivo de actitudes sociales, el intercambio de ideas, la cooperación y el desarrollo de la personalidad. El trabajo en grupo estimula a sus integrantes y hace que discutan sobre la mejor solución para un problema.
- Alto grado de interdisciplinariedad. Hace posible la cooperación de disímiles disciplinas.
- Desarrolla la expresión y creatividad. Las herramientas que proporcionan las TIC (procesadores de textos, editores gráficos...) facilitan el desarrollo de habilidades de expresión escrita, gráfica y audiovisual.

Perspectiva para los estudiantes

- Personalización de los procesos de enseñanza y aprendizaje. La existencia de múltiples materiales didácticos y recursos educativos, facilita la individualización de la enseñanza y el aprendizaje; cada alumno puede utilizar los materiales de acuerdo a su estilo de aprendizaje y sus circunstancias personales.
- Autoevaluación. La interactividad que proporcionan las TIC pone al alcance de los estudiantes múltiples materiales para la autoevaluación de sus conocimientos.
- Flexibilidad en los estudios. Los entornos de tele formación y la posibilidad de que los alumnos trabajen ante su ordenador con materiales interactivos.

Las TICs son tecnologías favorables, producto de una razón fundamental: su gran versatilidad le ha permitido, y le continuará permitiendo en los próximos años, una introducción explosiva, transformadora y benéfica en términos productivos, sociales y culturales en general, en múltiples y muy diversas actividades humanas (Zilberstein Toruncha et al., 2006). La introducción de las TICs brinda condiciones óptimas para transformar una enseñanza tradicional, pasiva, fundamentalmente centrada en la transmisión del contenido, en otro tipo de educación más personalizada, práctica, centrada en alcanzar aprendizajes diversos, que posea una real significación para cada estudiante, y que esté dirigida a lograr una dimensión profundamente humana.

I.3 Evolución de la Enseñanza Asistida por Computadora (EAC)

La enseñanza a finales del siglo pasado y en lo recorrido en el actual, ha estado matizada por el uso de medios técnicos auxiliares, dentro de los cuales la computadora ha desempeñado una función preponderante por las ventajas que incorporó, tanto para la explicación de los conceptos como para su apropiación. En la medida que ha ido avanzando la tecnología se han buscado métodos que resulten efectivos para el proceso docente educativo. Se puede afirmar entonces, que a cada paradigma de la informática ha estado asociada una versión didáctica que apoya el proceso de enseñanza aprendizaje de disímiles contenidos (Campos et al., 1997).

Históricamente, los educadores han desarrollado y aplicado diversos medios auxiliares destinados a mejorar el proceso de enseñanza aprendizaje, esos medios han ido modificándose con el tiempo, de acuerdo a los avances de la ciencia. Como era de esperarse, el empleo de la computadora como forma de apoyo al proceso docente ha provocado un interés creciente de los educadores, prácticamente desde el inicio de la "era de la computación". En la práctica, esta idea ha venido a hacerse realidad con la aparición de las computadoras personales, que han permitido una amplia difusión de los sistemas de Enseñanza Asistida por Computadora (EAC)(Bello et al., 2002).

La Enseñanza Asistida por Computadora consiste en tecnología de la información que permite al estudiante acceder a múltiples recursos además de la información simple. Se pretende entonces que esta ayude al escolar en sus procesos de aprendizaje. La EAC puede incluir desde programas de ejercitación hasta aplicaciones que instruyan a los alumnos contenidos sin ayuda del profesor. En el primer caso, la computadora ayuda a que el educando repase lo visto en clases, mientras que en el segundo caso, es la encargada de presentar, ejercitar y evaluar el aprendizaje del estudiante.

En los años cincuenta aparecieron los primeros sistemas de enseñanza, conocidos como programas lineales, en los que ningún factor podía cambiar el orden de enseñanza establecido por el programador. Estos sistemas no tenían en cuenta la posibilidad de que el alumno no hubiera comprendido correctamente los conceptos expuestos hasta el momento, esta limitación tiene su origen debido al diseño e instrumentación de este tipo de sistemas educativos, basado fundamentalmente en los principios conductistas de Skinner (Escalona, 2005). Los programas lineales se caracterizaban además por no ofrecer una enseñanza basada en los niveles de aprendizaje de los estudiantes, es decir, todos los alumnos recibían el mismo conocimiento y exactamente con una misma secuencia. La mayor contribución de este tipo de programa fue resaltar la necesidad de que el alumno trabajara con el material al ritmo que considerara más apropiado.

A finales de los años sesenta y principios de los setenta surgen los sistemas generativos, asociados a una nueva filosofía educativa que plantea: "los alumnos aprenden mejor enfrentándose a problemas de dificultad adecuada, que atendiendo a explicaciones

sistemáticas" (Urretavizcaya Loinaz, 2001). Este procedimiento de enseñanza surge al reconocerse que la propia computadora podría generar el material de enseñanza, ya que esta es capaz de desarrollar preguntas, construir soluciones y diagnosticar las respuestas del alumno, controlando el nivel de dificultad de los problemas.

Los sistemas generativos, establecen el nivel de dificultad de las preguntas que se presentan; para ello se tiene en cuenta, cuál es el concepto que se aborda y con qué grado de detalle requiere ser comprobado. Este tipo de procedimiento de enseñanza en la práctica no era utilizado para todo tipo de instrucción, ya que las dificultades para generar problemas aumentan en ciertas áreas de trabajo. Otro contratiempo de interés es el hecho de que estos medios crean una única solución para una interrogante en concreto y pueden existir múltiples soluciones correctas para una misma pregunta.

La aparición de las computadoras personales, en la década de los ochenta del pasado siglo, facilitó y difundió el acceso a los medios de cómputo y una consecuencia directa de esa realidad fue el comienzo de su uso, en forma masiva, en el proceso docente educativo. Una etapa de desarrollo posterior permitió combinar las técnicas de la Inteligencia Artificial (IA), los modelos psicológicos de los estudiantes y del experto y las teorías de la educación, para hacer surgir los Sistemas Tutores Inteligentes (STI), estos sistemas pretenden actuar de la misma forma que un buen profesor (Campos et al., 1997).

En la década de los noventa se promueve el desarrollo de la multimedia lo que permitió una mayor interacción entre el alumno y la computadora, facilitando la comunicación a través de la incorporación de nuevos recursos de enseñanza tales como:

- Imágenes: las cuales son un importante medio de comunicación, mediante ellas se pueden transmitir ideas y conceptos. Este recurso posee un elevado poder pedagógico y permiten guiar al usuario de un programa docente a realizar acciones que tienden a la adquisición de conocimientos (Valdés et al., 2001).
- Gráficos: utilizados en la mayoría de los sistemas de enseñanza, ya que su utilización brinda resultados positivos, si los docentes hacen un buen uso de las facilidades disponibles (Valdés et al., 2001).

La EAC presenta características que la diferencian de otros medios, como son las destacadas por (Galvis, 2006):

- La computadora está en capacidad para almacenar, procesar y presentar información multimedia en forma interactiva, de modo que es posible crear contextos para aprendizaje en los que se puede dar una relación dialogal con el nivel de concreción o abstracción que haga falta, bajo control del usuario o del diseñador, según convenga.
- Los equipos de cómputo pueden comportarse con distintos niveles de inteligencia, siendo un signo de inteligencia en un sistema de EAC la similitud mayor o menor que pueda tener su ejecución al comportamiento racional de un humano. Entre más inteligente sea el comportamiento de estos sistemas, más flexibilidad otorga al proceso de enseñanza aprendizaje y provee un ambiente más poderoso.
- La computadora facilita distintos grados de interactividad, gracias a su capacidad de procesar información, a los avances en Inteligencia Artificial y a las interfaces de distintos tipos que se pueden crear.
- Los dispositivos de cómputo permiten la conexión y articulación con otros medios y recursos de aprendizaje, facilitando la creación de ambientes cooperativos de enseñanza que aprovechan cualidades únicas e intrínsecas de cada uno de ellos.

I.4 La enseñanza de la matemática en la escuela cubana

La enseñanza de la Matemática en la escuela cubana constituye la base sobre la cual se sustentan los conocimientos posteriores que ha de adquirir el alumno en su proceso de formación y educación.

Desde los primeros grados, la enseñanza de esta ciencia se inicia mediante un sistema de conocimientos, científicamente fundamentados, que se aplican y profundizan de forma constante y que toma cada vez un carácter más generalizado.

La enseñanza de la Matemática brinda un importante aporte a la educación de los alumnos porque permite no solo la solución de problemas de su vida cotidiana, sino también el desarrollo de determinadas cualidades como la responsabilidad, honestidad, colectivismo,

perseverancia, así como la aplicación de los conocimientos y habilidades matemáticas en la participación activa de la vida familiar y social (León Roldán et al., 2012).

La enseñanza de esta ciencia en la escuela cubana tiene como tarea contribuir a la preparación de los jóvenes para la vida laboral y social. Se trata de que los jóvenes dispongan de sólidos conocimientos matemáticos que permitan interpretar los adelantos científicos, que sean capaces de operar con ellos con rapidez, rigor y exactitud de modo consciente y que puedan aplicarlos de forma creadora a la solución de problemas de diversas esferas de la vida en la construcción del socialismo.

El estudio de las matemáticas tiene gran importancia para la formación de los alumnos ya que permite el desarrollo de su pensamiento lógico y los motiva a la búsqueda y el análisis reflexivo del conocimiento centrado en su valor para la sociedad (Rodríguez Izquierdo et al., 2001). El cálculo mental hace un aporte esencial al desarrollo de las capacidades intelectuales generales como el análisis, la síntesis, la comparación, la abstracción y la generalización; también al desarrollo de la memoria y a la concentración de los escolares. Sirve, además, para fundamentar el proceso de formación de los números naturales y las relaciones entre estos. Así mismo forma parte del proceso de fijación de estos números.

Los ejercicios de cálculo mental constituyen la base para la comprensión del proceso de aplicación de un procedimiento escrito que hay que realizar al calcular.

En el Tercer Grado de la Enseñanza Primaria es donde el educando amplía su dominio numérico hasta el número 10 000 mediante la realización de ejercicios cuya vía de solución se sustenta en habilidades como:

- Determinación del menor o mayor número bajo determinadas condiciones o a partir de un conjunto de números específicos.
- Determinación del sucesor o antecesor de un determinado número.
- Representar números naturales en el Sistema de Posición Decimal comprendiendo el concepto de lugar, posición y la importancia de la cifra básica cero, así como determinar la cantidad de unidad de millar, centenas, decenas y unidades.
- Formar números naturales a partir de un conjunto de condiciones.

Es en este grado donde se inicia el aprendizaje de los procedimientos escritos de las cuatro operaciones fundamentales de cálculo, adición, sustracción, multiplicación y división por lo que se incorporan nuevos procedimientos algorítmicos para el cálculo con números naturales hasta 10 000.

Un objetivo fundamental es el conocimiento de los significados prácticos de las operaciones aritméticas, para ello deben dominarse los fundamentos conceptuales para poder operar con conocimiento de causa y poder transitar por los tres niveles de desempeño (Villalón Incháustegui et al., 1990).

Otro de los objetivos de la asignatura Matemática en el Tercer Grado es lograr la comprensión del concepto de fracción, a partir de sus significados prácticos para aplicarlos a la solución de problemas muy sencillos, cuya vía de solución se fundamenta en las habilidades siguientes:

- Identificar fracciones: a partir de una fracción específica leerla, a partir de la lectura de la fracción determinar el número fraccionario que le corresponde.
- Partiendo de imágenes, identificar la fracción que representa.
- Resolver problemas sencillos mediante reflexiones lógicas en que intervenga el significado práctico de las fracciones.

Debido a que en este grado se inicia una nueva etapa de exigencias en la formación académica, es importante que los alumnos realicen suficientes actividades que le permitan transitar por los diferentes niveles del desarrollo cognitivo, con este objetivo fue creado el software “La Isla de las Matemáticas”, el cual utiliza técnicas basadas en la Teoría de Compiladores, por lo que en las siguientes secciones se abordarán algunos temas principales, como los analizadores y autómatas.

I.5 Autómatas, lenguajes y gramáticas

Toda comunicación involucra la utilización de un lenguaje. Así, por ejemplo, las personas se comunican con el resto del mundo a través de los diferentes idiomas (lenguajes naturales) o con las máquinas a través de un conjunto de símbolos (lenguajes artificiales).

Una consideración importante, es la distinción entre los lenguajes formales y los naturales. Se puede decir que la principal diferencia consiste, en que los lenguajes formales, como pueden ser los lenguajes de programación, obedecen a reglas preestablecidas y por tanto, se ajustan a ellas, además de haber sido creados para un fin específico. Sin embargo, los lenguajes naturales, utilizados por el hombre, designan una forma de lenguaje con fines comunicativos y las reglas gramaticales que rigen su estructura, han sido desarrolladas con posterioridad a su surgimiento (Isasi et al., 2001).

Una máquina abstracta u autómatas, es un dispositivo teórico capaz de recibir y transmitir información. Para llevar a cabo esta tarea, manipula los símbolos contenidos en la cadena de entrada, produciendo otra cadena como salida, para lo cual utiliza un conjunto de estados que contienen la información necesaria para poder deducir, dado un símbolo de entrada, cuál será el símbolo de salida en cada momento. Debe destacarse que a pesar de la conexión que existe entre estos conceptos (Figura I.1), los trabajos iniciales sobre gramáticas y máquinas abstractas tienen orígenes distintos.

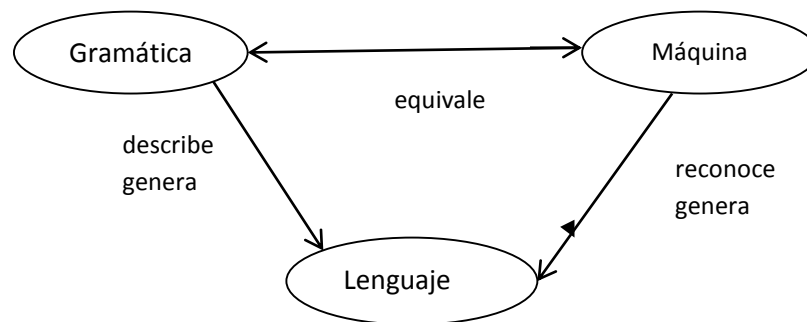


Figura I.1 Relación entre los conceptos lenguaje, gramática y autómatas

En general existen dos esquemas diferentes para definir un lenguaje, los cuales se conocen como esquema generador y esquema reconocedor, en función del principio que se siga para la definición de las cadenas que pertenecen al lenguaje (Kelley, 1995). En el primer caso, o sea, en el caso de los esquemas generadores se trata de un mecanismo que permite “generar” las diferentes sentencias del lenguaje, en el segundo caso se trata de un

mecanismo que permite reconocer si una cierta sentencia pertenece o no a un cierto lenguaje. Los representantes más significativos de estos esquemas son el Esquema de Chomsky como esquema generador de lenguajes a través del concepto de Gramática y la Teoría de Autómatas como esquema reconocedor típico de lenguajes.

El esquema generador, denominado gramática a grandes rasgos, no es más que una manera de definir un lenguaje, proporcionando una lista de las reglas que el lenguaje ha de obedecer.

La gramática se denota por la cuádrupla $G = (N, \Sigma, P, S)$ donde (Brookshear, 1993):

- a) N es el conjunto finito de símbolos no terminales de la gramática.
- b) Σ es el conjunto finito de símbolos terminales de la gramática, $N \cap \Sigma = \emptyset$.
- c) P es un subconjunto finito de $[(N \cup \Sigma)^+ - \Sigma^+] \times (N \cup \Sigma)^*$, llamado el conjunto de composiciones o reglas de producción.
- d) $S \in N$ es un símbolo especial que se denomina símbolo inicial.

Chomsky clasificó las gramáticas en cuatro grandes grupos (G_0, G_1, G_2, G_3), y se cumple la regla de inclusión siguiente: ($G_3 \subseteq G_2 \subseteq G_1 \subseteq G_0$).

La gramática tipo G_0 se denomina gramática *sin restricciones o de estructura de frases*, la tipo G_1 se conoce como *sensible al contexto*, la G_2 se identifica como *independiente del contexto* y por último, la gramática G_3 se denomina *regular*. Cada uno de estos tipos de gramáticas es capaz de generar un tipo de lenguaje.

Clasificación de Chomsky

Sea $G = (N, \Sigma, P, S)$ una gramática (Gálvez, 2005).

- a) Si toda composición es de la forma $A \rightarrow \alpha$ o bien $A \rightarrow \alpha B$ en donde $A, B \in N$, $\alpha \in \Sigma \cup \{\varepsilon\}$, se dice que G es una gramática regular o lineal a la derecha.
- b) Si toda composición es de la forma $A \rightarrow \delta$ en donde $A \in N$ y $\delta \in (N \cup \Sigma)^*$, se dice que G es una gramática libre del contexto.
- c) Si toda composición es de la forma $\alpha A \beta \rightarrow \alpha \delta \beta$ en donde $\alpha, \beta, \delta \in (N \cup \Sigma)^*$ y $A \in N$, se dice que G es una gramática sensible o dependiente al contexto.

- d) Si una gramática G no cumple las restricciones anteriores se denomina gramática sin restricciones.

La teoría de los autómatas proviene del campo de la ingeniería eléctrica. Shannon publicó varios trabajos donde demostraba la aplicación lógica matemática de los circuitos combinados y secuenciales. Posteriormente, sus ideas se desarrollaron para dar lugar a la Teoría de Autómatas. Moore publicó el primer estudio riguroso sobre autómatas y a finales de los años cincuenta, se comenzó a ver la utilidad de los autómatas en relación con los lenguajes (Fernández and Sáez, 1995).

La teoría de lenguajes y gramáticas formales tiene una relación directa con la teoría de máquinas abstractas, siendo posible así establecer entre ambas un isomorfismo. Dado que las gramáticas proporcionan las reglas utilizadas en la generación de las cadenas de un lenguaje, se puede establecer una conexión entre la cadena de lenguajes generados por ciertos tipos de gramáticas y las clases de lenguajes reconocidos por ciertas máquinas. Así, se puede identificar los lenguajes de tipo G_0 con la clase de lenguaje reconocido por la Máquina de Turing, los lenguajes de tipo G_1 con los Autómatas Linealmente Acotados, los de tipo G_2 con los Autómatas de Pila y por último, los tipo G_3 con los Autómatas Finitos (Isasi et al., 2001).

Todas estas máquinas se caracterizan por poseer una cinta a partir de la cual obtiene los símbolos de entrada, un conjunto de estados que representan diferentes fases del análisis de las palabras de entrada, un lugar donde generan la salida y en algunos casos cuentan con dispositivos auxiliares de memoria. Las diferencias entre ellas estriba en la capacidad para escribir en la cinta de entrada, en los distintos tipos de movimientos que pueden realizar sobre la cinta, si tiene o no memoria auxiliar, entre otras.

A partir de una gramática se puede construir una máquina reconocedora del lenguaje generado por dicha gramática, de tal forma que cuando reciba como entrada una cadena de símbolos, indicará si dicha cadena pertenece o no al lenguaje.

Se dice que una máquina reconoce un lenguaje L si es capaz de reconocer todas las sentencias pertenecientes a L y de no reconocer ninguna sentencia que no pertenezca a L (Aho et al., 2007).

La teoría de gramáticas y autómatas tiene múltiples aplicaciones en diversas áreas del conocimiento. Algunos ejemplos de las aplicaciones de los lenguajes regulares y los autómatas finitos son los analizadores léxicos que se utilizan en los compiladores de lenguajes de programación. Normalmente, un analizador léxico es un autómata que se utiliza para el reconocimiento de las palabras empleadas en un lenguaje de programación (variables, palabras reservadas, números, etc.). También se utilizan en los editores de texto para buscar y reemplazar palabras que se equiparan con una expresión regular.

I.6 Traductores, compiladores e intérpretes

Un traductor es un sistema que recibe como entrada un programa escrito en un lenguaje fuente y produce como salida otro escrito en un lenguaje destino, estos a su vez son los encargados de informar los errores sintácticos y semánticos, si están presentes. Los traductores engloban tanto al compilador como al intérprete y los mismos se escriben en un lenguaje designado como lenguaje de implementación.

Por otro lado un compilador es un programa que realiza un proceso de traducción en el cual se convierte un conjunto de declaraciones fuentes, escrito en un lenguaje de alto nivel, a un programa objeto, en código de máquina, listo para su ejecución en la computadora. Como parte importante de este proceso de traducción, el compilador informa a su usuario la presencia de errores en el programa fuente.

Los conocimientos sobre cómo organizar y escribir compiladores han aumentado mucho desde su surgimiento a principios de los años cincuenta. Es difícil dar una fecha exacta de la aparición del primer compilador, porque en un inicio gran parte del trabajo de experimento y aplicación se realizó de manera independiente por varios grupos (Aho et al., 2007). La mayoría de las primeras incursiones en la realización de compiladores está relacionada con la traducción de fórmulas aritméticas a código máquina.

En el año 1950, se consideraba que los compiladores eran programas notablemente difíciles de realizar. El primer compilador de FORTRAN, por ejemplo, necesitó para su implantación dieciocho años de trabajo en grupo. Desde entonces, se han descubierto técnicas sistemáticas para manejar muchas de las importantes tareas que surgen en la compilación.

Conceptualmente, un compilador opera en fases, cada una de las cuales transforma el programa fuente de una representación en otra. La descomposición típica de un compilador se muestra en la Figura I.2. A continuación se puntualizan algunos de los elementos que conforman la misma.

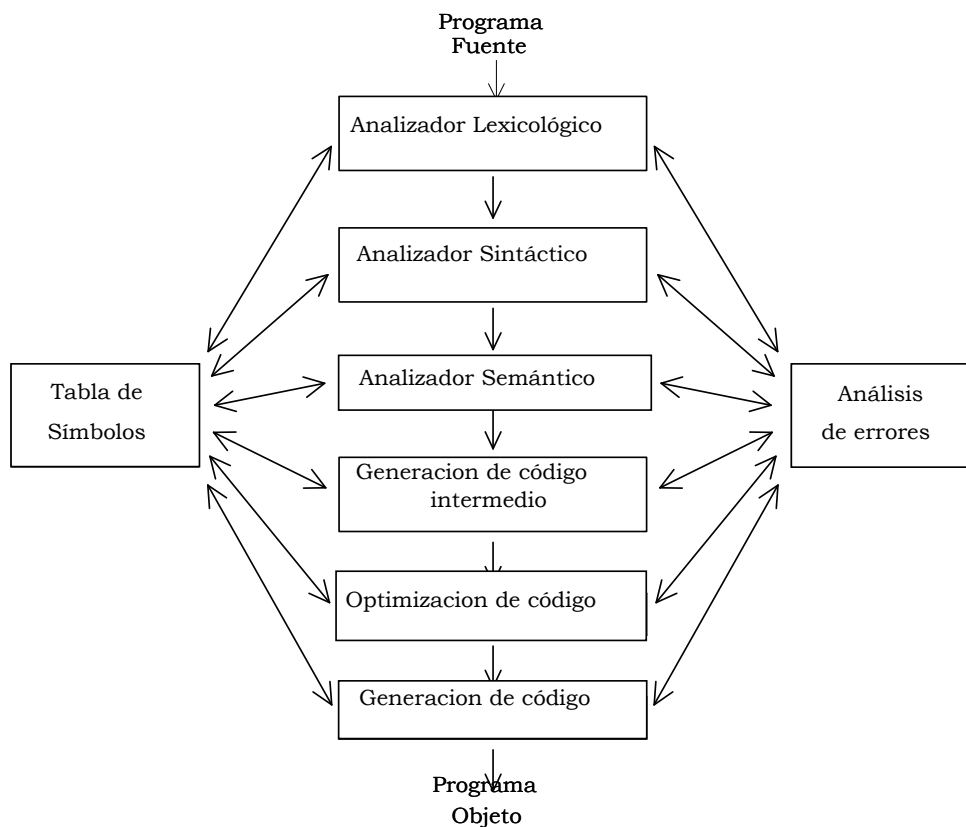


Figura I.2 Fases de un compilador

I.6.1 Análisis lexicológico (scanner)

En la fase de análisis lexicológico se leen los caracteres del programa fuente y se agrupan en cadenas que representan los componentes léxicos. Cada componente léxico es una secuencia lógica coherente de caracteres. Las categorías de los componentes léxicos pueden variar de un lenguaje a otro, pero en general se distinguen las relativas a un identificador, una palabra reservada, un operador o constantes numéricas y literales. A la secuencia de caracteres que representa un componente léxico se le llama lexema, o con su nombre en inglés token.

I.6.2 Análisis sintáctico (parsing)

En la fase de análisis sintáctico se comprueba que la estructura de los componentes básicos sea correcta, según ciertas reglas gramaticales, esto implica agrupar los componentes léxicos del programa fuente en frases gramaticales que el compilador utiliza para sintetizar la salida. Por lo general, las frases gramaticales del programa fuente se representan mediante un árbol de análisis sintáctico.

Un árbol sintáctico es una representación compacta del árbol de análisis sintáctico en el que los operadores aparecen como nodos interiores y los operandos de un operador son los hijos del nodo, para ese operador (Aho et al., 2007). Un ejemplo correspondiente con este tipo de árbol se muestra en la Figura I.3.

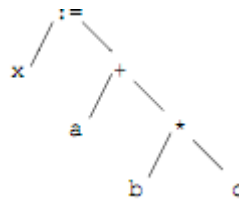


Figura I.3 Árbol sintáctico para $x:=a+b*c$

La estructura sintáctica de un programa normalmente se expresa utilizando reglas recursivas. Por ejemplo, se pueden dar las siguientes reglas como parte de la definición de expresiones:

1. Cualquier identificador es una expresión.

2. Cualquier número es una expresión.
3. Si $expresión_1$ y $expresión_2$ son expresiones, entonces también lo son
 $expresión_1 + expresión_2$
 $expresión_1 * expresión_2$
 $(expresión_1)$

La reglas (1) y (2) son reglas básicas, no recursivas, en tanto la regla (3) define expresiones en función de operadores aplicados a otras expresiones.

De manera similar, muchos lenguajes definen recursivamente las proposiciones mediante reglas como:

- Si $identificador_1$ es un identificador y $expresión_2$ es una expresión, entonces $identificador_1 := expresión_2$ es una proposición.
- Si $expresión_1$ es una expresión y $proposición_2$ es una proposición, entonces
while ($expresión_1$) **do** $proposición_2$
if ($expresión_1$) **then** $proposición_2$ son proposiciones.

La división entre análisis léxico y análisis sintáctico es algo arbitrario. Generalmente se elige una división que simplifique la tarea completa del análisis. Un factor para determinar la división es si una construcción del lenguaje fuente es inherentemente recursivo o no. Las construcciones léxicas no requieren recursividad, mientras que las construcciones sintácticas suelen requerirla.

I.6.2.1 Métodos de análisis sintáctico

El analizador sintáctico obtiene una cadena de componentes léxicos, a través del analizador léxico y comprueba si la cadena puede ser generada por la gramática del lenguaje fuente, para ello el analizador sintáctico informará de cualquier error de sintaxis de manera inteligente.

Existen tres tipos generales de analizadores sintácticos para gramáticas. Los métodos universales de análisis sintácticos, como el algoritmo de Cocke-Younger-Kasami y el de

Earley, los cuales pueden analizar cualquier gramática. Estos métodos sin embargo son demasiado ineficientes para usarlos en la producción de compiladores (Gálvez, 2005).

Los métodos empleados generalmente en los compiladores se clasifican como descendentes o ascendentes. Como sus nombres indican, los analizadores sintácticos descendentes construyen árboles de análisis sintáctico desde arriba (la raíz) hasta abajo (las hojas), mientras que los analizadores sintácticos ascendentes comienzan en las hojas y suben hacia la raíz. En ambos casos, se examina la entrada de izquierda a derecha, un símbolo a la vez.

I.6.3 Análisis semántico

La fase de análisis semántico revisa el programa fuente para tratar de encontrar errores semánticos y reúne la información sobre los tipos para la fase posterior de generación de código. En ella se utiliza la estructura jerárquica determinada por la fase de análisis sintáctico para identificar los operadores y operandos de expresiones y proposiciones (Louden, 1997).

En esta fase, el compilador verifica si cada operador tiene operandos permitidos para la especificación del lenguaje fuente.

I.6.4 Generación de código intermedio

Una vez realizado los análisis sintácticos y semánticos, algunos compiladores generan una representación intermedia explícita del programa fuente, esta representación puede ser considerada como un programa para una máquina abstracta. La representación intermedia generada debe tener dos propiedades importantes, debe ser fácil de producir y fácil de traducir al programa objeto (Vivancos, 2000).

Esta fase genera un código independiente de la máquina. En general, estas representaciones deben hacer algo más que calcular expresiones, también deben manejar construcciones de flujo de control y llamadas a procedimientos.

Las ventajas del empleo de las representaciones intermedias son: facilita la realización de pseudocompiladores y hace factible la optimización de código.

I.6.5 Optimización de código

La fase de optimización de código trata de mejorar las expresiones continuas en un lenguaje de programación intermedio, de modo que resulte un código de máquina más rápido de ejecutar. Hay muchas variaciones en la cantidad de optimización de código que ejecutan los distintos compiladores, los llamados “compiladores optimadores”, dedican una parte significativa de su tiempo a esta fase. Sin embargo, hay optimaciones sencillas que mejoran sensiblemente el tiempo de ejecución del programa objeto sin retardar demasiado la compilación (Aho et al., 2007).

Esta etapa debe realizarse durante las fases de generación de código intermedio y/o generación de instrucciones de máquina, y puede considerarse como una fase aislada o estar integrada con ellas. La optimización del código intermedio debe ser independiente de la máquina.

I.6.6 Generación de código

La fase final de un compilador es la generación de código, que por lo general consiste en código de máquina relocizable o código ensamblador. Las posiciones de memoria se seleccionan para cada una de las variables usadas por el programa. Después, cada una de las instrucciones intermedias se traduce a una secuencia de instrucciones de máquina que ejecutan la misma tarea.

I.6.7 Tabla de símbolos

Una función esencial de un compilador es registrar los identificadores utilizados en el programa fuente y reunir información sobre los distintos atributos de cada identificador. Estos atributos pueden proporcionar información sobre la memoria asignada a un identificador, su tipo, su ámbito y, en el caso de nombres de procedimientos, aspectos

como la cantidad y tipos de argumentos, el método usado para pasar cada argumento y el tipo que devuelve, si lo hay (Louden, 1997).

Una tabla de Símbolos es una estructura de datos que contiene un registro para cada identificador y sus atributos (Aho et al., 2007). Una implementación puede ser una tabla hash, la cual será actualizada a lo largo de todas las fases del proceso de compilación, pues cuando el analizador léxico detecta un identificador en el programa fuente, lo introduce en la tabla de símbolos, sin embargo, normalmente los atributos de un identificador no se pueden determinar durante el análisis léxico, es por ello que las fases restantes introducen información sobre estos en la tabla de símbolos y luego la utilizan de varias formas.

I.6.8 Análisis de errores

Cada fase puede detectar errores que deben tratarse (Aho et al., 2007). De forma general las fases de análisis sintáctico y semántico manejan una gran cantidad de ellos.

Los intérpretes no producen programas objetos (Vivancos, 2000) ya que su finalidad es traducir y ejecutar el código directamente, sentencia a sentencia. Una de las principales ventajas de los intérpretes es la facilidad para localizar y corregir errores por parte del programador.

I.7 Patrón arquitectónico Modelo-Vista-Controlador

La necesidad de modificar software ya elaborados remarca el objetivo de lograr un diseño claro que facilite las labores de mantenimiento. Una idea válida es dividir el código en partes funcionales que se puedan distinguir fácilmente.

Para el caso de los software realizados con fines docentes, esa necesidad se hace más imperiosa ya que los cambios en la forma de enseñar deben ajustarse al desarrollo de la enseñanza y de la tecnología.

La arquitectura de software se centra tanto en los elementos estructurales significativos de los sistemas, subsistemas, clases, componentes y nodos, así como en la colaboración que tiene lugar entre estos elementos a través de las interfaces (Ivar Jacobson, 2000). Esta se

conforma a partir de un conjunto de decisiones significativas sobre la organización del software, como la selección de los elementos estructurales y las interfaces, junto con su comportamiento, describiendo los cimientos del sistema, que son necesarios para comprenderlo y desarrollarlo.

En su forma más simple, la arquitectura es la organización de los componentes pertenecientes a la aplicación, la manera en que estos actúan y las estructuras de datos empleadas (Pressman, 2006).

La arquitectura se desarrolla de forma iterativa durante la fase de elaboración pasando por el análisis de requisitos, diseño, implementación y las pruebas, utilizando los casos de uso significativos y un conjunto de entradas, para implementar la línea base o "esqueleto" del sistema. Ese conjunto de entradas incluye los requisitos asociados a la aplicación y los sistemas heredados que deben utilizarse. Sin embargo, la arquitectura del software está afectada no sólo por la estructura y el comportamiento, sino también por el uso, funcionalidad, rendimiento, flexibilidad, reutilización, facilidad de comprensión, restricciones y la estética.

Un diseño arquitectónico define a una familia de sistemas informáticos, en términos de su organización estructural y patrones que pueden ser utilizados por estos, para satisfacer los requisitos impuestos a la aplicación, conjunto con las restricciones que afectan la manera de aplicar dichos patrones (Reynoso, 2004).

Un patrón es como una solución a un problema de diseño que aparece con frecuencia (Ivar Jacobson, 2000). Este recoge soluciones estándares a problemas de la arquitectura que ocurren reiteradamente. Los patrones de arquitectura de software se centran en las estructuras e interacciones que ocurren entre subsistemas y sistemas.

Uno de los patrones arquitectónicos más conocido y ampliamente utilizado para el diseño de la interfaz gráfica de usuario (GUI), es el patrón arquitectónico Modelo-Vista-Controlador (MVC), que se aprecia en la Figura I.4.

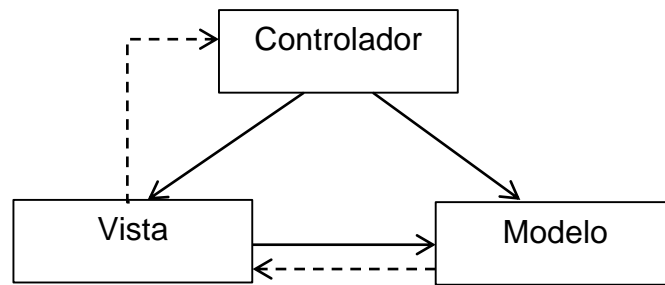


Figura I.4 Patrón Arquitectónico Modelo-Vista-Controlador

El patrón arquitectónico Modelo-Vista-Controlador fue descrito en la década de los ochenta, como una aproximación al diseño de GUI, permitiendo múltiples presentaciones de un objeto y estilos independientes de interacción (Sommerville, 2005). Este surge con el objetivo de reducir el esfuerzo de programación, necesario durante la implementación de sistemas, a través de la estandarización del diseño del software.

El MVC es un paradigma que divide las partes que conforman la aplicación en el Modelo, las Vistas y los Controladores, de esta forma el sistema queda conformado por tres capas: una encapsula los datos, otra es la interfaz y la última es la lógica interna. Esta concepción permite la implementación por separado de cada uno de los elementos componentes, todo lo cual garantiza que la actualización y mantenimiento del software se pueda desarrollar de forma sencilla y en un reducido espacio de tiempo.

El patrón MVC es una filosofía de diseño de aplicaciones, compuesta por (Fernández and Díaz, 2012):

- **Modelo:** es el encargado de la administración de los datos, controlando todas sus transformaciones. Este no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas los cambios del Modelo. Es el Modelo donde se encapsulan las funcionalidades de la aplicación.
- **Vistas:** es la encargada de manejar la visualización de la información almacenada en el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Controlador, pero es posible que trate directamente con el

Modelo a través de una referencia a este, pero nunca podrá realizar cambios en el estado. Pueden ser notificada cuando ocurren cambios de estados en el Modelo.

- **Controlador:** es el responsable de darle significado a las órdenes del usuario, ejecutando la acción adecuada y actuando sobre los datos representados por el Modelo. Controla el flujo entre las vistas y el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia a este.

Tanto la Vista como el Controlador dependen del Modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes (Bascón Pantoja, 2011):

- **Soporte de múltiples vistas:** Dado que la vista está separada del modelo y no hay dependencia directa del modelo con respecto a esta, la interfaz de usuario puede mostrar múltiples visualizaciones de los mismos datos simultáneamente.
- **Adaptación al cambio:** Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación, generalmente no afecta al modelo.
- **División clara del código:** la separación clara entre los componentes permite su implementación por separado.
- **Interfaz de Programación de Aplicaciones API (Application Programming Interface)** bien definida: cualquiera que use la API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- **Conexión entre el Modelo y sus Vistas dinámica:** se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas del programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si posteriormente se

observa que uno de los componentes no funciona adecuadamente, puede reemplazarse sin que las otras piezas se vean afectadas. Este escenario contrasta con la aproximación monolítica típica de muchos programas de pequeña y mediana complejidad, caracterizados por poseer un Frame que contiene todos los elementos necesarios en la aplicación, dentro de los que destacan: el controlador de eventos, los cálculos necesarios y la presentación de los resultados.

I.7.1 Funcionamiento del MVC

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que generalmente sigue el control es el siguiente (Fernández and Díaz, 2012):

1. El usuario interactúa con algún elemento de la interfaz al pulsar un botón, enlace, etc. El controlador recibe a través de los objetos de la interfaz, la notificación de la acción solicitada por el usuario y obtiene de la Vista, los datos asociados a los componentes contenidos en esta.
2. El controlador accede al modelo, actualizándolo de acuerdo a la acción solicitada por el usuario.
3. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada, donde se reflejan los cambios realizados en el modelo. Este último no debe tener conocimiento directo sobre la vista. Sin embargo, el controlador puede ser utilizado para proveer cierta interacción entre el modelo y la vista.
4. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

I.8 Conclusiones del capítulo

La enseñanza es una tarea compleja que necesita de recursos que ayuden a hacerla más amena y productiva. Comprender la necesidad del empleo de las TIC en el proceso docente educativo y la incidencia positiva del empleo de EAC en este proceso es uno de los primeros pasos necesarios para lograr una enseñanza de calidad y transformar la instrucción tradicionalmente pasiva en otro tipo de educación más personalizada y participativa.

En el capítulo se ha expuesto un análisis de los principales aspectos necesarios para la comprensión de la presente propuesta computacional, basada en la Teoría de Compiladores, tales son los conceptos relacionados con el patrón arquitectónico Modelo-Vista-Controlador, autómatas, gramáticas, lenguajes, compiladores e intérpretes. En el próximo capítulo se realizará un análisis interno al software realizado, mostrando una aplicación práctica de los conceptos previamente expuestos.

CAPÍTULO II. “LA ISLA DE LAS MATEMÁTICAS”.VISIÓN INTERNA

II.1 Objetivos del capítulo

En este capítulo se describe la arquitectura del software “La Isla de las Matemáticas”. Se explican los principales aspectos relacionados con su implementación, haciendo énfasis en los paquetes y las clases que lo conforman.

II.2 Módulos que componen el sistema

El software “La Isla de las Matemáticas” está compuesto por dos módulos:

- **Módulo Profesor.** Esta parte del sistema está dirigida a los docentes y permite la gestión de los ejercicios que se le presentarán a los estudiantes. Dispone de diferentes opciones que facilitan el proceso de gestionar los ejercicios y se toma en cuenta el hecho de que está dirigido a profesores que no tienen que ser especialista en computación, entre las facilidades incorporadas se pueden destacar las siguientes: el chequeo ortográfico, la vista previa de los ejercicios, consultar documentos y acceder a la ayuda asociada a la aplicación.
- **Módulo Estudiante:** Esta componente tiene la responsabilidad de presentar los ejercicios a los estudiantes y chequear la validez de las respuestas de los alumnos, informándole si es correcta o incorrecta.

A pesar de que las funcionalidades de estos módulos están relacionadas, cada uno puede funcionar de manera independiente.

II.3 Actores y casos de uso

A partir de la descripción del problema, se elabora el diagrama de actores y casos de uso, el cual representa gráficamente a los procesos y su interacción con los actores.

Como parte del proceso de captura de requisitos; se identifican los actores, los cuales son entidades externas del sistema que, de alguna manera, participan en el desarrollo del caso de uso. Estos últimos se definen según los roles y constituyen artefactos narrativos que

describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario.

El diagrama de casos de uso se utiliza para describir el comportamiento del sistema al afrontar una tarea, los actores que actúan en “La Isla de las Matemáticas” son (Figura II.1):

- Estudiante. Utiliza el módulo del sistema orientado a la ejercitación de los conocimientos adquiridos durante el proceso de enseñanza aprendizaje.
- Profesor: Gestiona los ejercicios que estarán disponibles en la aplicación, los perfecciona y enriquece.

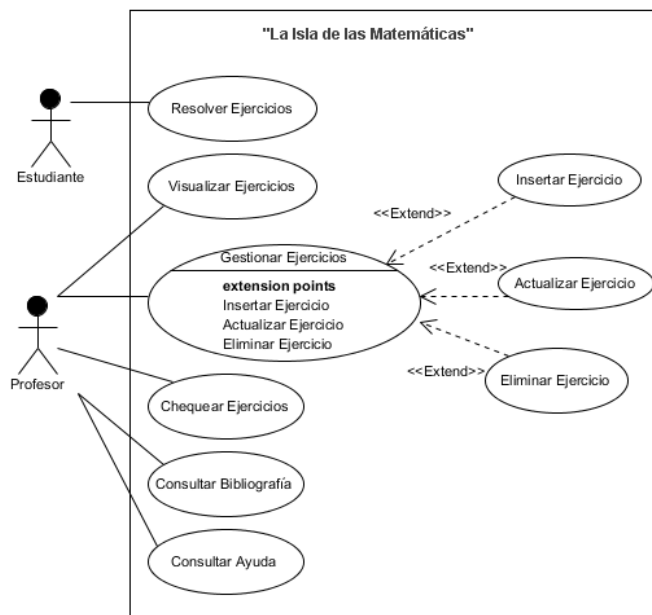


Figura II.1 Casos de uso y actores del sistema “La Isla de las Matemáticas”

A continuación se ilustran los diferentes casos de uso detectados durante el proceso de captura de requisitos del sistema.

- Resolver Ejercicios. Presenta los ejercicios agrupados por las materias de numeración, cálculo y trabajo con fracciones. Estos ejercicios se caracterizan por poseer un planteamiento, asociado con la problemática que se desea resolver y un conjunto de distractores representando posibles respuestas asociadas al planteamiento anterior, incluyendo la respuesta correcta. El sistema se encarga de evaluar las decisiones tomadas por el usuario.

- Visualizar Ejercicios. Permite el acceso simplificado a los ejercicios conocido como vista previa. Muestra el conjunto de atributos que identifican al ejercicio.
- Chequear Ejercicios. Se muestran los ejercicios de una forma muy similar a lo que verá el actor estudiante durante el caso de uso resolver ejercicios. Permite detectar errores de contenidos o incongruencias en los ejercicios antes de usarlos en el proceso de enseñanza aprendizaje.
- Consultar Bibliografía. Permite que el actor profesor interactúe con un conjunto de documentos que contribuyan a aumentar la calidad educativa de los ejercicios presentados a los estudiantes.
- Consultar Ayuda. Brinda acceso a la documentación asociada al software proporcionando una guía para interactuar con él.
- Gestionar Ejercicios. Permite insertar, actualizar y eliminar los ejercicios asociados con la aplicación.

II.4 Arquitectura del software

El paradigma utilizado para el desarrollo del software “La Isla de las Matemáticas”, es el Modelo-Vista-Controlador, con este patrón se logra una división de las diferentes partes que conforman la aplicación, lo que facilita su actualización y mantenimiento de una manera sencilla y en un reducido espacio de tiempo.

Con el propósito de cumplir los requisitos del modelo, el software desarrollado está dividido en tres partes principales: el Modelo que representa los datos de la aplicación, la Vista donde se tiene una presentación del modelo de datos y el Controlador el cual manipula las peticiones realizadas por los usuarios.

La aplicación “La Isla de las Matemáticas” está conformada por un conjunto de paquetes, los cuales agrupan las clases que lo componen y facilitan su mantenimiento.

Los paquetes están contenidos dentro de la carpeta src y existe una correspondencia entre las tres partes principales a tener en cuenta durante el empleo del MVC y los nombres asociados a estos. La Figura II.2 muestra una imagen de la estructura de los paquetes.

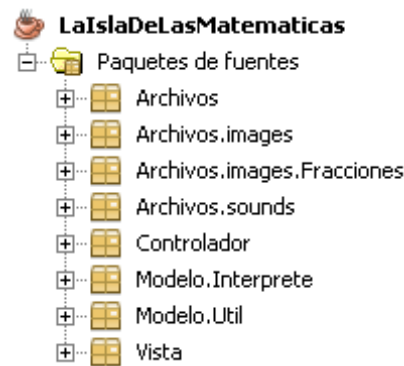


Figura II.2 Estructura de los paquetes del sistema “La Isla de las Matemáticas”

La organización de los elementos del sistema en paquetes, ofrece la ventaja de separar las clases e interfaces asociadas a los distintos módulos de la aplicación, brindando soporte a una vista de nivel superior y permitiendo contemplar el código en forma de agrupamientos más simples. Existe una dependencia entre paquetes que se muestra, gráficamente, en la Figura II.3.

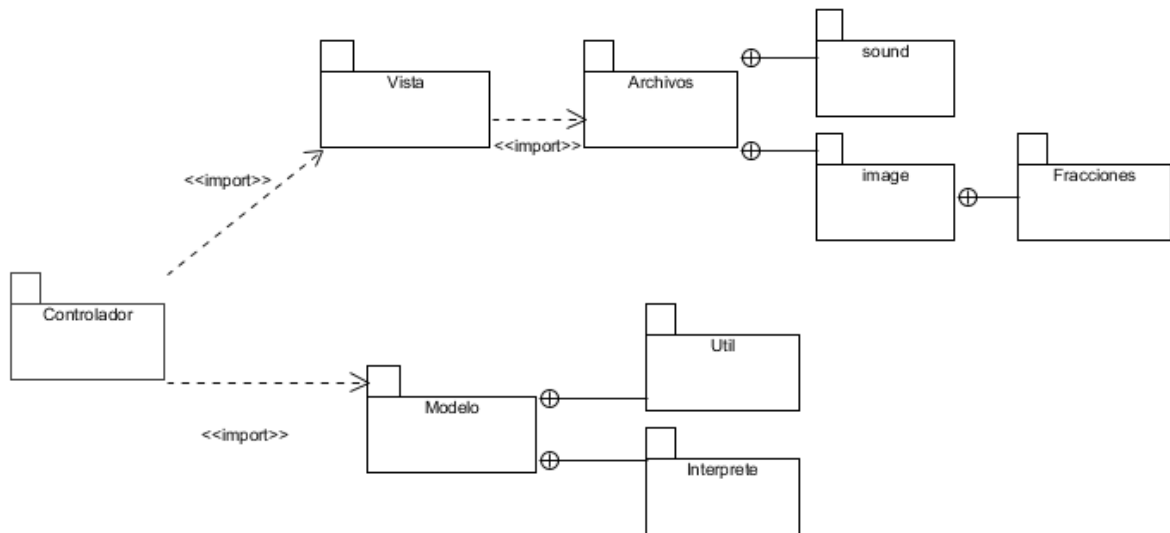


Figura II.3 Diagrama de paquetes

En el caso particular del software “La Isla de las Matemáticas” el Modelo está conformado por las clases asociadas con los distintos autómatas e intérpretes, necesarios para lograr un desempeño eficiente de la aplicación.

La Vista está formada por un conjunto de clases cuyo objetivo principal es la visualización de los datos obtenidos a través del modelo, permitiendo una interacción agradable entre el usuario y el sistema. Este elemento primordial del MVC se hace corresponder con el paquete Vista de la aplicación, ver Figura II.4, que a su vez hace uso del paquete Archivos, (ver Figura II.5), que contiene las clases relacionadas con el tratamiento de imágenes y sonidos. De forma general todas las clases contenidas en el paquete Vista poseen una misma estructura e interacción con los componentes modelo y controlador.

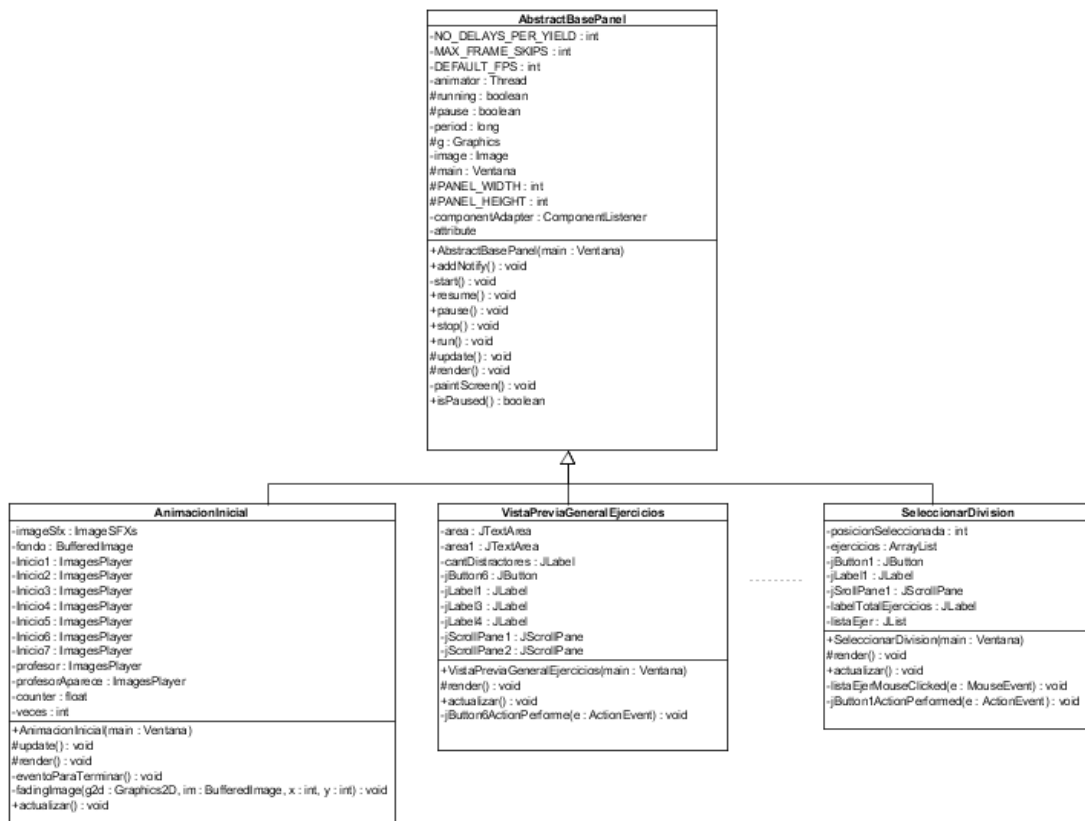


Figura II.4 Diagrama de clases del paquete Vista

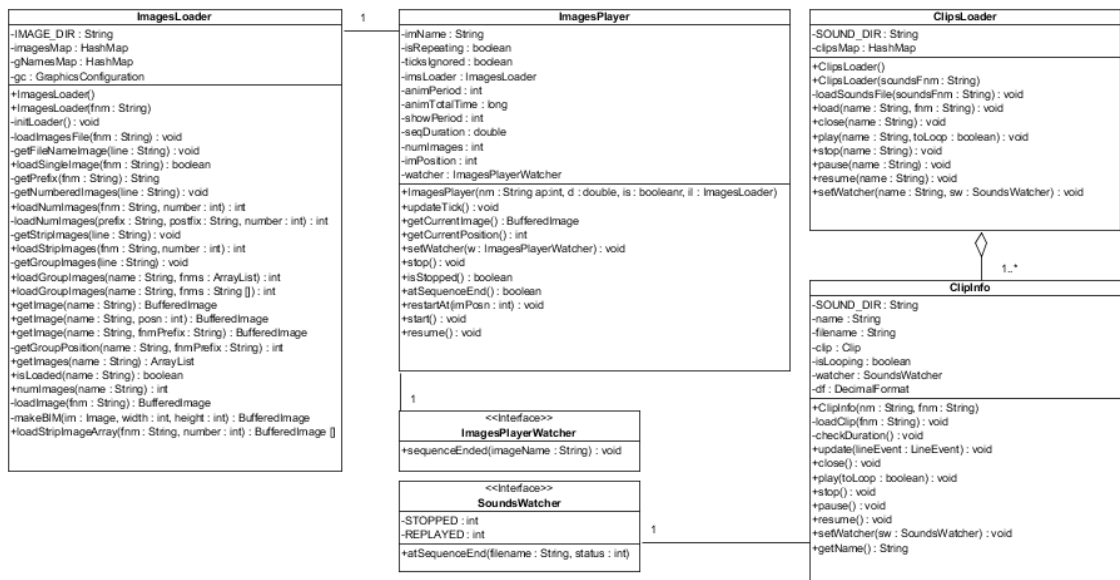


Figura II.5 Diagrama de clases del paquete Archivos

El Controlador es uno de los elementos fundamentales del MVC y es el encargado de responder ante las acciones de los usuarios, e invocar peticiones al modelo y a la vista. En el caso de la presente aplicación, los elementos que conforman en el controlador se encuentran en el paquete Controlador (ver Figura II.6).

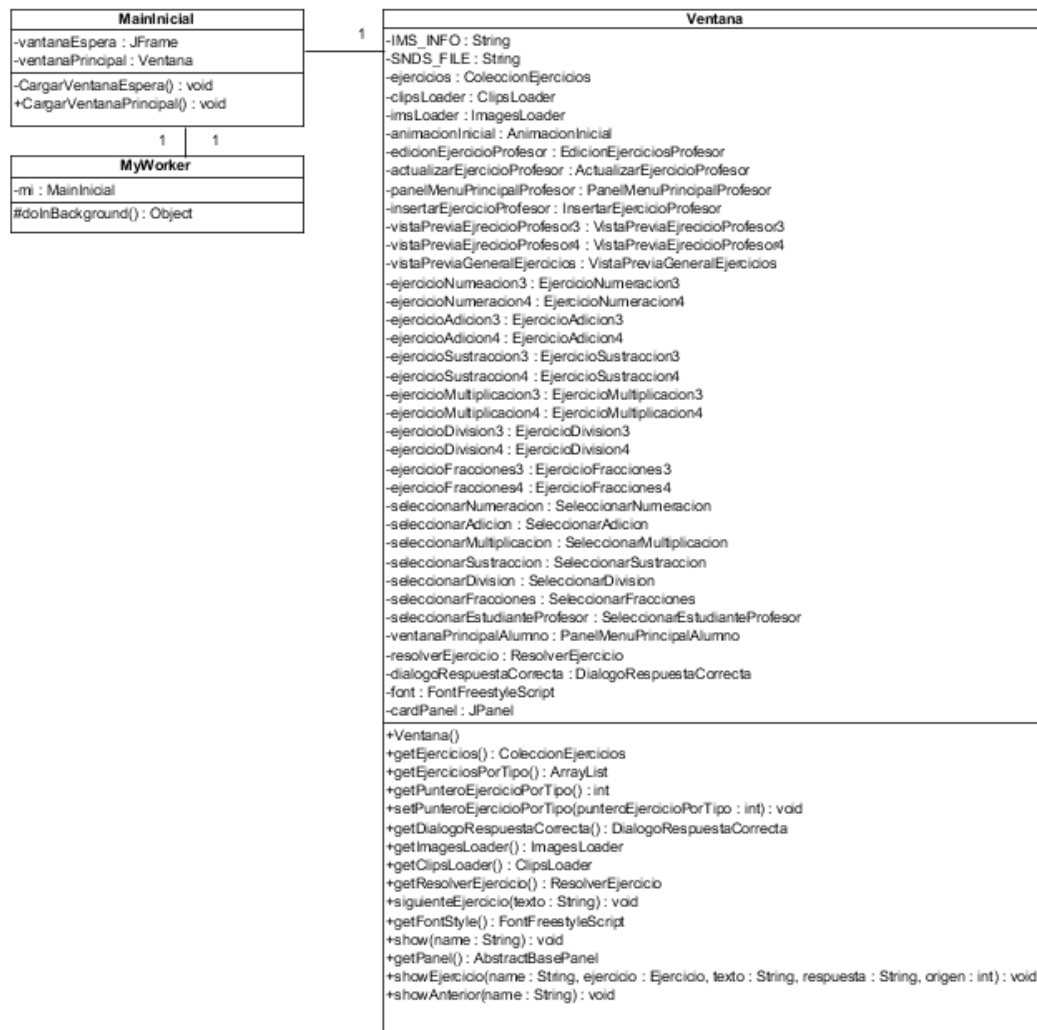


Figura II.6 Diagrama de clases del paquete Controlador

Específicamente en el caso de la aplicación “La Isla de las Matemáticas” existe una única clase controladora, la cual es la encargada de establecer la conexión entre el modelo y las vistas, además de definir la lógica de administración del sistema.

II.4.1 Utilización de recursos gráficos y de audio

Una tecnología primordial para desarrollar software caracterizados por una gran carga de imágenes y animaciones, es la implementación de algoritmos de animación que se desempeñen de forma fiable y rápida a través de varios sistemas operativos como: las versiones de Windows, Linux y Macintosh.

Para el desarrollo de la aplicación “La Isla de las Matemáticas”, se incorpora en la clase `AbstractBasePanel`, un método encargado del trabajo con las animaciones.

En particular esta clase se extiende de `javax.swing.JPanel` y actúa como un lienzo para dibujar gráficos en 2D; como: líneas, círculos, texto e imágenes. Este panel contiene un bucle de animación que está incrustado dentro del hilo local, lo que asegura que se avance a un ritmo constante. El bucle consta de tres etapas: actualización, dibujo y un corto período de sueño, este último permite que el subproceso de animación detenga la ejecución, liberando la unidad de procesamiento central (Central Processing Unit o CPU) para otras tareas, tales como la recolección de basura que realiza la máquina virtual de Java (Java Virtual Machine o JVM). Sin estos instantes de sueño, se podría acaparar casi todo el tiempo de CPU, además de traer consigo la aparición de eventos unidos, ya que cuando la JVM está sobrecargada por solicitudes de repintado puede optar por combinarlas, lo cual significa que parte de las peticiones de dibujo serían omitidas, haciendo que la animación refleje una pérdida de cuadros.

`AbstractBasePanel` actúa como un lienzo en blanco de tamaño fijo, en el cual se utiliza la técnica de doble almacenamiento (double buffering) de dibujo, reduciendo así el parpadeo de los objetos visualizados, ya que si se realizan extensos dibujos directamente en la pantalla, el proceso puede tardar un tiempo suficientemente prolongado como para llegar a ser perceptible por el usuario. Este panel se inserta dentro de un `javax.swing.JFrame` y la ejecución se caracteriza por tener dos hilos principales: el hilo animador encargado de la actualización del dibujo y un hilo de procesamiento de eventos Graphical User Interface (GUI), que es el encargado de las pulsaciones de teclas y movimientos del ratón.

Uno de los aspectos más interesantes durante la implementación de este panel es lo difícil que es saber cuándo el repintado se ha completado. Esto significa, que si el tiempo de descanso en el bucle de animación es muy reducido o si la demora especificada es demasiado larga, entonces la velocidad de la animación se vería afectada y la ejecución del bucle no lograría ser aproximadamente la misma en todas las plataformas, para ello se estableció un espacio de descanso ajustado a los estándares de Frames Per Second (FPS).

El límite inferior para los valores de FPS está determinado por el ojo humano y la Critical Flicker Frequency (CFF), que es la velocidad a la que una luz parpadeante parece ser continua. Esto se produce entre los 10 y 50 Hz, en función de la intensidad de la luz; lo que se traduce en 10 a 50 FPS (Knudsen, 2001).

Un posible límite superior para lograr buenos valores de FPS son las frecuencias de actualización de los monitores. Esto es típicamente de 70 a 90 Hz, es decir, 70 a 90 FPS. Un programa no necesita enviar más cuadros por segundo que la frecuencia de actualización de la tarjeta gráfica. De hecho, una tasa excesiva FPS consume tiempos de CPU innecesarios y se extiende a la tarjeta de visualización (Paul, 2005).

II.4.1.1 Hilos de ejecución en Java

Un hilo es una unidad básica de utilización de la CPU; que incluye un identificador (ID), un controlador de programa, un conjunto de registros y una pila. Este comparte con otros subprocesos que pertenecen al mismo proceso la sección de código, datos y otros recursos del sistema operativo, como los archivos abiertos y las señales (Tanenbaum, 2006). Si un proceso tiene múltiples hilos de control, puede realizar más de una tarea a la vez. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

Las ventajas de la programación multihilo pueden dividirse en cuatro categorías principales (Silberschatz et al., 2013):

- Capacidad de respuesta: el uso de varios hilos en una aplicación interactiva permite, que un programa continúe ejecutándose incluso aunque parte de él esté bloqueado o realizando una operación muy larga, incrementando así la capacidad de respuesta al usuario.
- Recursos compartidos: por defecto, los subprocesos comparten la memoria y los recursos del proceso al que pertenecen. La ventaja de compartir el código y los datos es que permite que la aplicación tenga varios hilos de actividad diferentes dentro del mismo espacio de decisiones.

- Economía de recursos: la asignación de memoria y recursos para la creación de procesos es costosa. Dado que los hilos comparten los recursos del proceso al que pertenecen, es más económico crear y realizar cambios de contexto entre los subprocesos.
- Utilización sobre arquitectura multiprocesador: las ventajas de emplear configuraciones multihilo pueden verse incrementadas significativamente en una arquitectura multiprocesador, donde los hilos pueden ejecutarse en paralelo en los diferentes procesadores.

El hecho de que los subprocesos de ejecución de un mismo proceso compartan los recursos hace que cualquiera de estos hilos pueda modificarlos. Cuando esto ocurre los otros subprocesos acceden a ese dato modificado inmediatamente.

El proceso que posee diferentes hilos asociados, sigue en ejecución mientras al menos uno de sus subproceso de ejecución siga activo. Cuando el proceso finaliza, todos sus hilos de ejecución también han terminado, es en este momento que se considera que el proceso no existe más y todos sus recursos se liberan (Robbins and Robbins, 2003).

Un ejemplo de la utilización de subproceso es tener un hilo atento a la interfaz gráfica (iconos, botones, ventanas), mientras otro subproceso hace una larga operación internamente. De esta forma el programa responde de una manera más ágil a la interacción con el usuario. También pueden ser utilizados por una aplicación servidora para dar servicio a múltiples clientes.

Los hilos son el modelo fundamental de ejecución de programas realizados en el lenguaje de programación Java y su Application Program Interface (API), proporciona un amplio conjunto de características para la creación y gestión de hilos.

Todos los programas Java tienen al menos un hilo de control; incluso el más sencillo que solo posea el método `main()` y se ejecuta como un subproceso en la JVM.

Existen dos técnicas para la creación de hilos en un programa Java. Un método consiste en crear una nueva clase que se derive de la clase `Thread` para después implementar un método `run()`. Una técnica alternativa, que se utiliza frecuentemente, consiste en definir una clase

que implemente la interfaz Runnable, en este caso debe definirse un método run(), ya que el código contenido en este es el que se ejecuta como un subproceso separado (Oaks and Wong, 2004).

II.4.1.2 El ciclo vital de un subproceso

La vida de un proceso pasa por diferentes estados que describen, de cierta forma, la actividad que están realizando en cada instante de tiempo. En general esos estados pueden ser los representados en la Figura II.7.

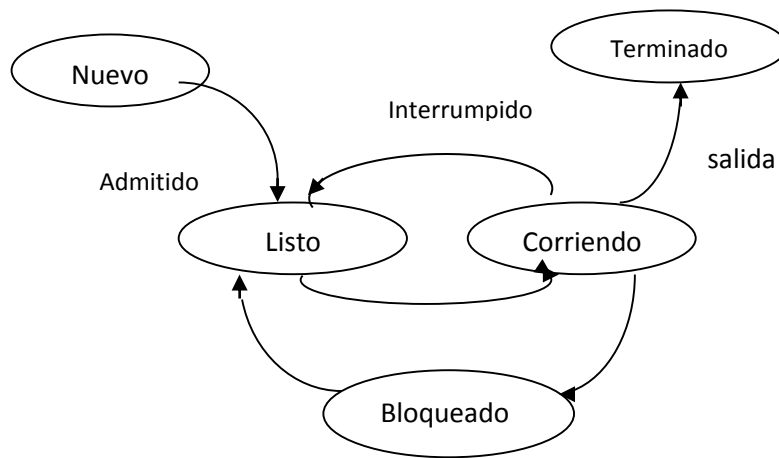


Figura II.7 Transición de estados de un hilo

El estado Nuevo ocurre cuando se crea el proceso, este es el momento en que sucede la transición de programa a proceso. Es decir, entra en este estado justo después de la declaración new y permanece en este hasta que el programa llama al método start de la clase Thread, mediante el cual se hace la transmisión del subproceso al estado Listo.

Por otro lado el estado Listo describe que el subproceso está preparado para ejecutarse y sólo espera a que le asignen la CPU para poderlo hacer.

Un subproceso hace la transmisión del estado Listo al de Ejecución, es decir, empieza a ejecutarse cuando el sistema operativo le asigna un procesador.

Los subproceso que se encuentran en el estado de Ejecución cambian al estado Terminado cuando su método run() termina o cuando finalizan por una excepción no atrapada. Los

hilos que se encuentran en el estado Terminado, no se pueden recuperar. Para evitar que estos finalicen, se puede incluir un bucle en su método run().

El cambio al estado Bloqueado ocurre cuando se intenta realizar una tarea que no se puede completar de manera inmediata y se debe esperar temporalmente hasta que pueda realizarse, por ejemplo, cuando un subproceso envía una solicitud de Entrada/Salida (E/S). En este caso, el sistema operativo bloquea la ejecución del hilo hasta que pueda completarse la solicitud de E/S. Una vez que se ingrese en este estado, el subproceso no puede usar el procesador, aunque esté disponible (Tanenbaum, 2006).

Un hilo entrará en el estado Bloqueado cuando ocurra una de las siguientes situaciones:

- Se invoque al método wait().
- Se llame al método sleep().
- Se está en espera de que se complete alguna operación de E/S.
- Se utilizó el método join() para unirse a otro subproceso.

II.4.1.3 Trabajo con imágenes

Para lograr la implementación del software se empleó el formato gráfico Portable Network Graphics (PNG) el cual incluye, entre las funciones más notables: un gran soporte para la profundidad y corrección del color, conjunto con transparencias de 8 bit. Este formato permite obtener imágenes con áreas translúcidas, lo cual es particularmente útil para efectos como rayos láser y humo. Otras ventajas es la corrección de gama, que permite la presencia de brillo en la imagen. Esta última característica hace que el formato PNG sea una buena opción de almacenamiento (Roelofs, 1999). Aunque no es este el único formato gráfico que existe, ya que se pueden destacar otros como:

- Graphics Interchange Format (GIF) formato que utiliza el esquema de compresión sin pérdida LZW (Lempel-Ziv-Welch), muy eficiente en ilustraciones planas y puede representar un máximo de 256 colores, por consiguiente, es inevitable una cierta pérdida cuando se representan imágenes en color verdadero, tales como las fotografías. Se utiliza, frecuentemente, para representar gráficos de estilo de dibujos animados usando pocos colores (Milburn, 2006).

- Joint Photographic Experts Group (JPEG) formato caracterizado por comprimir la información visual en bloques de 8x8 píxeles, estos bloques se cargan en secuencias cuando se visualiza la imagen. Este proceso de compresión puede causar grandes áreas con un solo color. Los archivos JPEG se utilizan, a menudo, en el trabajo con grandes imágenes fotográficas, aunque este formato no ofrece transparencia (Brundage and Grover, 2006).

Para la manipulación de las imágenes se implementó la clase `ImagesLoader` la cual puede manipular imágenes individuales, tiras y múltiples archivos de dibujos, que representan una secuencia de animación. Esta clase es la encargada de realizar el proceso de carga de las imágenes especificadas en el archivo de configuración `imsInfo.txt`, y para ello el objeto `ImagesLoader` crea dos estructuras de datos principales, las cuales son de tipo `HashMap`, permitiendo que la interfaz de acceso a las imágenes sea uniforme e independiente del carácter especificado en el archivo de configuración.

Esta clase puede cargar cuatro categorías diferentes de imágenes, que se especifican mediante los caracteres **o**, **n**, **s**, **g**, para ello se asume que las imágenes se encuentran almacenadas en un archivo local en el subdirectorio `images`. Estas se almacenan como `BufferedImages` utilizando la función `read()` de `ImageIO`, para convertirlas en imágenes administradas.

La forma típica para utilizar un objeto `ImagesLoader` es suministrar un archivo de configuración que contiene los nombres correspondientes a los registros de imágenes, que deben cargarse durante el proceso de inicio del software. Sin embargo, es posible llamar a los métodos de carga de `ImagesLoader` en cualquier momento durante la ejecución.

A continuación se muestra un fragmento del archivo de configuración `imsInfo.txt` utilizado en “La Isla De Las Matemáticas”:

//Imágenes utilizadas en “La Isla De Las Matemáticas”

o FondoInicio.png

o bandera.png

o TextoNino.png

```
g menu Numeración.png Adición.png Sustracción.png Multiplicación.png División.png
Fracciones.png
g menuActivo NumeraciónSelect.png AdiciónSelect.png SustracciónSelect.png
MultiplicaciónSelect.png DivisiónSelect.png FraccionesSelect.png
n gataGif*.png 6
n patoGif*.png 6
.....
```

El cargador ignora las líneas en blanco y las que comienzan con `//`. La sintaxis de los cuatro formatos de imágenes son:

```
o <imagen>
n < imagen *.ext> <número>
s < imagen > <número>
g <nombre> < imagen > [<imagen >] *
```

- La línea que se inicia con la vocal **o**, especifica la carga de un archivo, llamado `<imagen>`.
- La línea que comienza con la consonante **n**, especifica que se cargue una serie de archivos de imágenes enumeradas.
- La línea que tiene como primer carácter una **s**, permite la carga un archivo en forma de tira de imagen.
- La especificación **g**, al inicio de la línea se interpreta como la orden de cargar un grupo de archivos. Después de cargados, el acceso a las imágenes se hace utilizando una notación de posición o por medio de sus nombres.

II.4.1.4 Efectos visuales para imágenes

Para la realización del presente proyecto fue necesario implementar un método, cuyo objetivo final es la aplicación del efecto visual desvanecedor a las imágenes, disminuyendo así la cantidad de memoria utilizada durante la realización de la animación inicial.

Este método de procesamiento de imágenes se ubica en la clase `ImageSFXs` (Figura II.8) y se caracteriza por no realizar cambios a las imágenes durante la manipulación. La principal

razón para no modificarlas, es para que este efecto pueda utilizarse dentro de un ciclo, además de que así se permite la reutilización de estos recursos gráficos en varios momentos durante la ejecución de la aplicación.

ImageSFXs
+VERTICAL_FLIP : int +HORIZONTAL_FLIP : int +DOUBLE_FLIP : int -gc : GraphicsConfiguration -negOp : RescaleOp -negOpTrans : RescaleOp -blurOp : ConvolveOp
+ImageSFXs() -initEffects() : void +drawFadedImage(g2d : Graphics2D, im : BufferedImage, x : int, y : int, alpha : float) : void

Figura II.8 Clase que realiza procesamiento de imágenes

El método drawFadedImage requiere una referencia al contexto de gráficos, la imagen, las coordenadas (x,y) que especifican dónde se va a dibujar la imagen modificada y un valor float (alpha). Durante su ejecución realiza varias comprobaciones al valor alfa¹ suministrado, informando los posibles errores cometidos.

Para la implementación de este método se tuvo en cuenta la composición, proceso en el cual se combinan dos imágenes, a las cuales se hace referencia a través de los nombres destino y fuente. Java 2D ofrece algunas reglas de composición que especifican diversas formas en que la fuente se puede combinar con el destino. La más útil para la implementación del método drawFadedImage es SRC_OVER (fuente encima de destino).

La clase AlphaComposite de Java 2D añade otro elemento a las reglas de composición, estos son los valores alfa de la fuente y el destino, los cuales pueden ser algo confuso, especialmente cuando ambas imágenes poseen canales alfa. Sin embargo, para el caso SRC_OVER, cuando se desea solo visualizar la imagen de destino, la alfa sólo se aplica a la imagen fuente. Un valor alfa de 0,0f hace desaparecer la fuente y se podrá observar solamente la imagen destino y un valor alfa de 1.0f hace totalmente opaca la fuente; existen diversos grados de translucidez intermedia.

¹ alfa: valor que expresa el grado de debilidad de los colores empleados al dibujar una imagen.

La Figura II.9 Valores diferentes de alfa aplicados al rectángulo muestra el resultado de aplicar tres valores diferentes de alfa a un rectángulo.

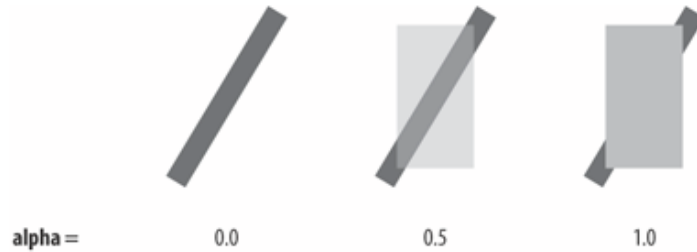


Figura II.9 Valores diferentes de alfa aplicados al rectángulo

II.4.1.5 Trabajo con sonidos

Uno de los elementos primordiales a tener en cuenta durante la realización de software educativos es el empleo de sonidos, que permiten reforzar los mensajes visuales que se intentan transmitir al estudiante, con el objetivo de cumplir este requerimiento se utilizó la API Java Sound de Java.

Las primeras versiones de Java sólo admitían archivos de Windows Wave 8 bits (mono), pero los diferentes formatos se ampliaron en el JDK 1.2 para incluir: Sun Audio (archivos de la UA), archivos de Mac AIFF, Musical Instrument Digital Interface (MIDI), y el formato Rich Media (RMF), permitiendo así que los datos puedan ser de 8 o de 16 bits, mono o estéreo y con frecuencias de muestreo entre 8000 y 48000 Hz (Mulholland and Murphy, 2003).

Debe destacarse que el API Java Sound posee diferentes capacidades de reproducción que le permiten realizar una manipulación de los datos de audio a bajo nivel, además de soportar la captura de audio.

El API Java Sound tiene dos partes principales: un paquete `javax.sound.sampled` para manipular muestras de audio y `javax.sound.midi` para secuencias Musical Instrument Digital (MIDI). Esta API tiene dos paquetes de proveedores de servicios, `javax.sound.sampled.spi` y `javax.sound.midi.spi`, para facilitar la extensibilidad. Estos suelen utilizarse para agregar nuevos dispositivos de audio, destacando los mezcladores, sintetizadores y formatos como el MP3 (Brackeen et al., 2003).

Para la implementación de la aplicación se empleó la API Java Sound con el objetivo de reproducir específicamente muestras de audio.

Una muestra de audio no es más que una serie de fragmentos digitales extraídos de las señales analógicas, como se ilustra en la Figura II.10 Muestra de audio, donde cada fragmento representa la amplitud (intensidad) de la señal en un momento dado (Hacker, 2000).



Figura II.10 Muestra de audio

La calidad del resultado digital depende de dos factores: la resolución de tiempo (tasa de muestreo); la cual se mide en Hertz (Hz), y de la resolución de amplitud (cuantificación), representado por la cantidad de bits que está presente en cada muestra. Por ejemplo, una pista de disco compacto (CD) está normalmente muestreada a 44,1 kHz (44.100 muestras por segundo), y cada muestra utiliza 16 bits para codificar 65536 amplitudes. A medida que aumentan la tasa de muestreo y la cuantificación, también lo hacen las necesidades de memoria, pero mejor será la calidad del sonido cuando el flujo digital se convierte de nuevo en una señal analógica (Brackeen et al., 2003).

Para el trabajo con audio muestreado se empleó la clase Clip que extiende de Line, ya que un objeto de este tipo es lo suficientemente pequeño (por lo general menos de 2 MegaByte (MB)) para ser cargado completamente en la memoria durante la ejecución. Además de la clase Clip se utilizó DataLine que también hereda de la clase Line, la cual incluye la capacidad de determinar la posición de lectura y escritura actual, iniciar, detener, pausar o reanudar el sonido.

Específicamente para facilitar el trabajo con la API Java Sound se implementaron las clases ClipInfo y ClipsLoader.

La clase ClipInfo implementa la interfaz LineListener para detectar cuando el clip ha terminado.

Por otro lado la clase ClipsLoader soporta la reproducción simultánea de varios clips y es la encargada de realizar el proceso de carga de los sonidos, basándose en las lista de nombres suministrados en el archivo de texto clipsInfo.txt o a través de las llamadas a métodos durante la ejecución. Para el caso particular de “La Isla de las Matemáticas” los sonidos utilizados durante la ejecución de la aplicación fueron cargados previamente, por lo que estos se encuentran disponibles para la reproducción una vez que la aplicación se haya iniciado.

La clase ClipsLoader posee un método de chequeo el cual se encarga de emitir un mensaje de advertencia si un clip es de un segundo de duración o menos.

II.4.1.6 Estructuras internas para el trabajo con sonidos

ClipsLoader almacena una colección de objetos ClipInfo en un HashMap. Donde el objeto ClipInfo es el responsables de cargar un clip, realizar pausas, reanudar y detener ese sonido cuando lo solicite ClipsLoader.

El nombre de los clips se obtiene a partir del archivo de información de sonidos clipsInfo.txt, que se carga cuando se crea el objeto ClipsLoader. Este archivo de información se supone que está en el subdirectorío sonidos.

A continuación se muestra un fragmento del archivo de configuración clipsInfo.txt utilizado en “La Isla de las Matemáticas”:

```
//Sonidos utilizados en “La Isla de las Matemáticas”
```

```
// Voces
```

```
MuyBien MuyBien.wav
```

```
QueBien QueBien.wav
```


Rectifica Rectifica.wav

SonidoFondo SonidoFondo.wav

II.5 El intérprete

Teniendo en cuenta que durante el proceso de actualización o inserción de los ejercicios, el profesor puede cometer errores de sintaxis, se hizo necesaria la confección de un compilador. Específicamente en el caso del software desarrollado se implementó un intérprete debido a que no es necesario generar código.

II.5.1 Análisis lexicológico (scanner)

Dentro del compilador, el proceso de análisis léxico es el encargado de agrupar los diferentes caracteres del flujo de entrada en tokens y determinar si están bien conformados, reportando los errores cuando existan. Los tokens son los símbolos léxicos del lenguaje y suelen contener información adicional. Una vez que estos son identificados, se transmiten al siguiente nivel de análisis.

Para la implementación del intérprete se utilizó el software libre ANTLR, el cual es un sofisticado generador de analizadores gramaticales, utilizado para implementar compiladores e intérpretes. El software genera código en Java y también se desarrolló en ese lenguaje.

Uno de los grandes puntos a favor de ANTLR es que posee un ambiente integrado de desarrollo (Integrated Develoment Enviroment – IDE), llamado ANTLRworks el cual se considera bastante completo para el diseño de gramáticas y lenguajes. Este visualiza las reglas gramaticales, busca errores y conflictos y si los encuentra muestra gráficamente cuál es el conflicto y su causa, incluso puede resolver, de manera automática, algunos de ellos, tales como los recursivos a la izquierda (left-recursion conflicts) (Bovet and Parr, 2008).

Estas capacidades son importantes para realizar, con mayor facilidad, el proceso de búsqueda de errores conceptuales en el diseño del lenguaje y en la detección de las ambigüedades.

Otras ventajas de ANTLR son (Parr, 2009):

- El código generado es muy legible y fácil de comprender (este es uno de los objetivos fundamentales de ANTLR), y el hecho de que genere parsers LL ayuda mucho a este objetivo.
- Integra la tecnología StringTemplate que está diseñada específicamente para generar textos estructurados como es el caso del código fuente.
- Los reconocedores generados por ANTLR emiten automáticamente muy buenos mensajes de error y se recuperan de forma inteligente.
- Es software libre y se encuentra bajo la licencia Berkeley Software Distribution (BSD).

El analizador lexicológico utilizado por “La Isla de las Matemáticas”, reconoce los operadores clásicos utilizados en el trabajo con expresiones aritméticas, como son la suma, resta, multiplicación y división, así como un conjunto de literales los cuales se emplean durante las especificaciones de las funciones disponibles por la aplicación. Se pueden mencionar: menor (...), mayor (...), leernumero (...), escribirnumero (...), entre otras.

En el scanner se reconocen también variables de tipo entero. Pero necesariamente han de comenzar con una letra.

II.5.2 El análisis sintáctico (parser)

En la fase de análisis sintáctico se aplican las reglas sintácticas del lenguaje, analizando el flujo de tokens, con el objetivo de determinar si cumplen las convenciones estructurales de la definición sintáctica del lenguaje.

II.5.2.1 La gramática

Los lenguajes de programación tienen reglas que describen la estructura sintáctica de aplicaciones bien formadas. La sintaxis de las construcciones de los lenguajes se puede describir por medio de notación BNF (Backus-Naur) o gramáticas independientes del contexto. Estas últimas permiten el uso de la recursividad y estructuras anidadas, elementos

significativos ya que estas facilidades no pueden representarse mediante expresiones regulares.

Las gramáticas independientes del contexto ofrecen ventajas significativas a los diseñadores de lenguajes y a los escritores de compiladores (Aho et al., 2007):

- Brindan una especificación sintáctica precisa de los lenguajes de programación.
- A partir de algunas clases de gramáticas se puede construir automáticamente un analizador sintáctico eficiente que determine si un programa fuente está sintácticamente bien formado.
- Las nuevas construcciones se pueden añadir con facilidad a un lenguaje cuando existe una aplicación basada en una descripción gramatical del lenguaje.

Teniendo presente los requerimientos del lenguaje que se quiere representar, se definió una Gramática Libre del Contexto (GLC), la cual consta de símbolos terminales y no terminales, permitiendo especificar la estructura sintáctica del lenguaje establecido para la gestión de los ejercicios que conforman la aplicación, además de que las GLC constituyen la base de los diferentes esquemas de traducción, por esas razones se escogió este tipo de gramática para el proyecto.

Dentro de los símbolos terminales se destacan los operadores aritméticos, para los cuales fue necesario definir la precedencia (ver Figura II.11) y si son asociativos a la derecha o a la izquierda, informando así al generador del parser el orden en que se deben realizar las operaciones.

```
exp      :      expMult ( ('+' expMult ) | ( '-' expMult ) ) *
          ;

expMult  :      term ( ('*' | '/' ) term ) *
          ;
```

Figura II.11 Precedencia de los operadores aritméticos

En este caso se empleó la técnica de dividir las expresiones en niveles de precedencia, de manera que si dos operadores tienen diferente prioridad están en niveles diferentes.

Para ello se comenzó con una regla general llamada **exp** que representa una expresión completa, esta es la encargada de verificar la correspondencia de aquellas expresiones que posean alguno de los operadores con la menor precedencia, es decir la suma y la resta, además de referenciar a la regla **expMult** la cual examinará la coincidencia de aquellas subexpresiones que contengan operadores pertenecientes al siguiente nivel de prioridad, en este caso la multiplicación y división.

II.5.3 Tratamiento de los errores

El empleo de ANTLR facilita la detección, reporte y recuperación de errores, ya que esta herramienta permite elevar excepciones, que son capturadas por los manipuladores especificados después del punto y coma al final de cada regla. La Figura II.12 muestra cómo fue realizado el tratamiento de errores.

La idea básica es forzar todos los errores a elevar algún tipo de excepción y luego darles tratamiento, emitiendo un mensaje al usuario que le informa, de una manera más explícita y sencilla, el error que ha cometido.

```
term      :      MENOR '(' INT ',' INT ')' + ')'
          |      MAYOR '(' INT ',' INT ')' + ')'
          |      ID
          |      INT
          |      '(' exp ')'
;catch [RecognitionException re] {
    mostrarOptionPane=1;
    javax.swing.JOptionPane.showMessageDialog(null,"Usted está intentando realizar operaciones"+
    "inválidas \n al especificar la forma de resolver el ejercicio.\n Para más información"+
    "remítase a la ayuda.", "Error", javax.swing.JOptionPane.ERROR_MESSAGE);
}
```

Figura II.12 Tratamiento de errores

II.5.4 La Tabla de Símbolos

La tabla de símbolos contiene las entradas para los identificadores y enteros que se utilizan durante la creación de los ejercicios. En este caso, la tabla de símbolos se implementó usando una Tabla Hash que contiene como llave una cadena, a la cual se le asocia un objeto de tipo Elemento, permitiendo así el tratamiento de las variables que toman un valor en un intervalo especificado por el usuario. Las variables se van insertando cuando aparecen por primera vez y continuamente se van actualizando los valores que contienen.

II.6 Conclusiones del capítulo

En el presente capítulo se han mostrado los módulos que componen el software “La Isla de las Matemáticas”. Además de realizarse una descripción de cada uno de los casos de usos y actores, identificados durante el proceso de captura de requisitos del sistema. Conjuntamente se abordaron los principales aspectos relacionados con el intérprete. Para la realización del intérprete se empleó la herramienta ANTLR.

CAPÍTULO III. APLICACIÓN DOCENTE DEL SOFTWARE

III.1 Objetivos del capítulo

En los momentos actuales se trabaja para elevar la calidad del aprendizaje de los estudiantes en las diferentes educaciones, lo que ha generado un conjunto de precisiones en el orden de la dirección del proceso de enseñanza-aprendizaje de las asignaturas. En el caso particular de la Matemática, se han producido cambios en el enfoque metodológico general de la asignatura, con el objetivo de contribuir a que los estudiantes adquieran una concepción científica del mundo, una cultura integral y las actitudes necesarias para ser ciudadanos plenos, útiles a nuestra sociedad, sensibles y responsables ante los problemas de naturaleza a escala local, nacional, regional y mundial. En este capítulo se exponen algunas ideas acerca del uso que debe dársele al sistema expuesto en el Capítulo II, el cual integra muchos de los ajustes curriculares realizados en la asignatura de Matemática impartida el Tercer Grado de la Enseñanza Primaria.

III.2 Caracterización de la asignatura de Matemática impartida al Tercer Grado

La Matemática es una disciplina de gran importancia, no solo por la aritmética de la vida práctica, sino porque además constituye una herramienta fundamental en diversos campos, entre ellos las ciencias naturales, la ingeniería, la medicina, las ciencias sociales, etc. Esta materia, a partir de sus nociones básicas y del razonamiento lógico, estudia las propiedades y relaciones cuantitativas entre los números, las figuras geométricas y los símbolos.

En la Educación Primaria la formación matemática de los escolares tiene gran valor, ya que en la actualidad el desarrollo científico-técnico y social alcanzado, exige de una sólida preparación de las nuevas generaciones en esta asignatura.

La tarea fundamental, es entonces, enseñar a los alumnos a pensar, por lo que entre los objetivos de la enseñanza, se destaca el desarrollo del pensamiento en los alumnos mediante la elaboración de tareas de aprendizaje y procedimientos metodológicos.

El Tercer Grado de la Enseñanza Primaria marca un momento importante en el Primer Ciclo, ya que si bien los dos primeros grados están muy unidos por el tipo de objetivos que se plantea y las tareas que desarrollan, el Tercer Grado se caracteriza por responder a un nivel más alto de exigencias a partir de lo logrado en los cursos anteriores y establece las bases para iniciar el Segundo Ciclo de la Educación Primaria.

Los programas y orientaciones metodológicas de las diferentes asignaturas del Plan de Estudio para el Tercer Grado, enfatizan en el desarrollo de las habilidades intelectuales y la formación educativa de los estudiantes. Específicamente en lo referente a la asignatura de Matemática se fortalece el trabajo con problemas, lo cual es muy importante si se parte de considerar que la solución de problemas es un proceso sumamente complejo que exige del escolar el despliegue de una intensa actividad cognoscitiva. Por esta razón, en el Tercer Grado se le da peso a la solución de problemas en todas las unidades.

Investigaciones realizadas en nuestro país han mostrado que en las condiciones actuales los estudiantes de Tercer Grado, al solucionar problemas no realizan un análisis adecuado de la situación presentada. Se ha visto que una vez planteada la tarea ellos comienzan a ejecutarla sin haberse detenido lo suficiente a esclarecer las condiciones y los medios de realización. Con una enseñanza apropiada que toma en cuenta no solo la resolución del problema, sino también la orientación hacia el propio proceso de solución, se irá eliminando esta “tendencia a la ejecución” (Rodríguez Izquierdo et al., 2001).

En el Tercer Grado los alumnos amplían sus conocimientos acerca de los números naturales; sobre la base del dominio de los números hasta 100 y los conocimientos del sistema de posición decimal, mediante la realización de consideraciones de analogías, aprenden como se forman números naturales mayores que 100 y hasta 10000, así como procedimientos de comparación y ordenamiento.

En este grado se inicia el aprendizaje de los procedimientos escritos de las cuatro operaciones fundamentales de cálculo; para lo cual deben ofrecerse una serie de indicaciones que los escolares deben interiorizar hasta su automatización. Especial énfasis hay que hacer en relación con la sustracción y la división que al ser las operaciones

inversas, presentan mayores dificultades. En el tratamiento metodológico hay que tener en cuenta los niveles de dificultad para cada operación.

A partir del Curso 2004-2005 se han producido cambios en los objetivos y contenidos de los programas de la asignatura Matemática, estos se han dado primero producto de la invitación de nuestro país para formar parte del Laboratorio Latinoamericanos para la Evaluación de la Calidad de la Educación (LLECE), por lo que fue necesario realizar los ajustes que se le hicieron a la asignatura para estar en consonancia con el resto de los países latinoamericanos participantes. Uno de los principales cambios realizados en el programa de estudio correspondiente a la asignatura de Matemática impartida al Tercer Grado, consiste en la introducción del concepto de fracción, tanto como parte de una unidad como parte de un conjunto, lo que permite solucionar situaciones que se presentan en la vida práctica mediante la aplicación de conocimientos matemáticos.

Es en este grado también, donde se continúa elevando el nivel de dificultad en la solución de ejercicios con texto y problemas, ya que se solucionan por primera vez problemas compuestos dependientes y se siguen utilizando formas y técnicas de trabajo que permiten desarrollar habilidades para hallar, cada vez con mayor independencia, una vía de solución para ejercicios o problemas dados. En particular los alumnos deben aprender a interpretar información sobre la vida cotidiana y su entorno más cercano, además de ejercitar las habilidades mediante el trabajo con los software educativos relacionados con los contenidos matemáticos de grados anteriores y con los que se introducen.

Las principales habilidades que debe adquirir el estudiante al vencer la asignatura son las siguientes:

- Representar, leer y escribir los números naturales hasta 10000.
- Profundizar en el conocimiento de las cuatro operaciones fundamentales de cálculo y las relaciones entre la adición-sustracción, multiplicación-división, adición-multiplicación, así como la conmutatividad y asociatividad de la adición y multiplicación de los números naturales y la distributividad de la multiplicación con respecto a la adición.
- Solucionar de forma independiente ejercicios con textos y problemas.

- Seleccionar la vía de solución que se considere más conveniente para resolver determinados ejercicios.
- Comprender el concepto de fracción a partir de sus significados prácticos para aplicarlos a la solución de problemas sencillos.

III.2.1 “La Isla de las Matemáticas” en el proceso docente de la asignatura Matemática impartida en el Tercer Grado

La asignatura de Matemática impartida en el Tercer Grado, está constituida por cuatro temas, los cuales se describen de manera general a continuación.

- Tema I “Los números naturales”
 - Consolidación de los números naturales hasta 100 y del cálculo con estos números.
 - Los números naturales hasta 10000. Identificación de regularidades en el mundo circundante y de patrones en situaciones numéricas y geométricas. Seriación
 - El orden de los números naturales hasta 10000.
- Tema II “Adición y sustracción hasta 10000”
 - Adición y sustracción hasta 10000. Cálculo oral.
 - El procedimiento escrito de la adición.
 - El procedimiento escrito de la sustracción
 - Ejercitación de la adición y sustracción. Aplicación
- Tema III “Multiplicación y división hasta 10000”
 - Multiplicación y división hasta 10000. Introducción del concepto de fracción como parte de la unidad.
 - Procedimiento escrito de la multiplicación.
 - Procedimiento escrito de la división. Introducción del concepto de fracción como parte de un conjunto.
 - Ejercitación de la multiplicación y la división. Aplicación. Cálculo de promedios. Resolución de problemas que impliquen la recogida, la descripción y la interpretación de datos.

- Tema IV “Geometría”
 - Relaciones de posición entre puntos y rectas y entre puntos.
 - Relaciones de posición entre rectas.
 - Paralelogramo (rectángulo y cuadrado).
 - Concepto intuitivo de movimiento.
 - Prisma (ortopedro y cubo).
 - Circunferencia, círculo, cilindro.

El software “La Isla de las Matemáticas” sirve como repositorio de documentos relacionados con los aspectos metodológicos y materiales de apoyo, de manera que los profesores puedan disponer de una fuente complementaria que permita reforzar su preparación pedagógica.

Para confeccionar el sistema se tomaron en cuenta los criterios de profesores expertos en la materia y solo se excluyó el cuarto tema referente a la geometría, para los restantes temas se incluye un conjunto de ejercicios que pueden ser perfeccionados o enriquecidos por parte del docente.

Se recomienda para el uso de la aplicación desarrollar una actividad inicial en la cual los estudiantes se familiaricen con el software. Después de esta actividad podrá orientarse su uso como forma de apoyo y soporte material al sistema de clases o como herramienta de trabajo independiente, logrando así una mayor motivación por aprender.

III.3 “La Isla de las Matemáticas” como medio de enseñanza

Se entiende por medio de enseñanza aquellos elementos materiales cuya función estriba en facilitar la comunicación que se establece entre educadores y educandos (Zilberstein Toruncha et al., 2006).

Los medios de enseñanza constituyen una función didáctica de gran importancia en el proceso docente educativo, específicamente, en la enseñanza aprendizaje de la Matemática juegan un papel esencial para el logro de muchos de los objetivos propuestos en los programas (Estrada Doallo et al., 2002). Estos son un componente importante dentro de la clase de Matemática tanto para el profesor como para el estudiante, pues contribuye al

cumplimiento de los objetivos de la actividad. Además de que fomentan la motivación de la actividad docente y estimula al estudiante hacia su realización.

Existen diferentes razones que desde el punto de vista filosófico apoyan y explican el papel de los medios en el proceso de enseñanza aprendizaje. Una de las más importantes es que los medios permiten materializar el objeto del conocimiento actuando sobre el sistema senso-racional del conocimiento en el aprendizaje, en este caso, dirigido por la labor orientadora del profesor (Lau Apó et al., 2001).

En el aspecto didáctico metodológico se pueden reconocer las siguientes formas en la utilización de medios de enseñanza (Estrada Doallo et al., 2002):

- Aplicarlos en la adquisición de conocimientos, especialmente en la transmisión de información, y la formación de ideas e intuiciones diferenciadas, así como de conceptos claros.
- Utilizarlos para la motivación y estimulación de determinadas actuaciones y para la formación de las actividades esperadas.
- Emplearlos para dirigir transcurso de actuaciones y series de operaciones en la formación de habilidades y capacidades incluyendo el control del nivel alcanzado.

De forma general el software “La Isla de las Matemáticas” como medio de enseñanza, cumple funciones: instructiva, cibernética, formativa y motivadora-innovadora-creadora, por lo que se debe insistir en la utilización de esta aplicación, ya que:

- Los objetivos y contenidos a los que responde, contribuyen al desarrollo de la personalidad del estudiante.
- Promueve los conocimientos, habilidades, relaciones intermaterias y potencialidades de los estudiantes.

III.4 Características del software “La Isla de las Matemáticas”

Un escenario adecuado para trabajar en la dirección de integrar a los avances pedagógicos y las nuevas tecnologías de la información y las comunicaciones (TIC), con la intención de dar respuesta a los cambios que reclama la actual sociedad, puede ser la confección de materiales didácticos, que ayuden a la preparación de los estudiantes, aprovechando las

posibilidades formativas, educativas y el ambiente motivacional de estas herramientas (Rodríguez Fernández et al., 2007) .

Un ejemplo dentro de estos materiales didácticos lo constituye el sistema presentado en el capítulo anterior, el cual posee la ventaja de permitir actualizar constantemente los ejercicios contenidos en dicha aplicación, por parte del profesor, agregando problemas de acuerdo al contexto en que se desarrollen las clases y sujetos a las características del grupo de estudiantes con el que se esté trabajando.

Es decir, las TIC vistas desde el panorama educativo, pueden enriquecer y transformar radicalmente las prácticas pedagógicas y científicas en los diferentes niveles educacionales, elevando significativamente la motivación por aprender (Zilberstein Toruncha et al., 2006).

En particular la aplicación “La Isla de las Matemáticas” presenta un grupo de características que le permiten darle solución al conjunto de carencias detectadas, en software precedentes, como es el caso del software “La Feria de las Matemáticas”. Entre las principales características que se pueden destacar se encuentran:

- Permite establecer un nuevo tipo de relación entre el profesor y la aplicación, mejorando la calidad del proceso docente educativo.
- Amplia los horizontes, permitiendo que el software también sirva de material complementario a los profesores, a través el empleo de los documentos metodológicos contenidos en este y dando la posibilidad de ajustarlos a los estilos actuales.
- Facilita el uso de dicha aplicación, al incluir una ayuda que trata de ilustrar y comentar los diferentes aspectos que podrían presentar dificultades durante el uso del software.
- Brinda una mayor cantidad de potencialidades que asisten al profesor durante la administración de los ejercicios, permitiéndole ajustarlos al diagnóstico del grupo con que trabaja.
- Posee un ambiente agradable y de fácil utilización por el usuario.

III.5 Indicaciones para el uso docente de la aplicación

Este software se caracteriza por presentar novedades respecto a la interacción usuario-aplicación, por ello es importante presentar algunas recomendaciones que ayudarán a usar de forma más efectiva la herramienta que se presenta. Las sugerencias de los expertos, de seguro le darán más riqueza a estas breves indicaciones iniciales.

- Llevar a cabo una pequeña fase de familiarización por parte de los profesores con la aplicación, de forma tal que se establezcan las bases para usarla en forma eficiente.
- Consultar la bibliografía que pone a disposición la aplicación, así como enriquecerla con nuevas informaciones que permitan mejorar la preparación pedagógica de los profesores.
- Emplear fundamentalmente el software en clases que ejerciten los contenidos abordados en la aplicación, no en aquellas clases de presentación de nuevos contenidos, ya que esto podría dificultar el proceso de asimilación.
- En la primera clase que se utilice esta aplicación, debe dedicarse un pequeño momento para familiarizar a los estudiantes con ella.
- Remitir al estudiante al software como parte del trabajo independiente, contribuyendo así a la ejercitación de las habilidades de cálculo, numeración y trabajo con fracciones.
- Se puede comenzar la ejercitación de los contenidos, por cualquiera de los complejos de materia siempre y cuando el estudiante ya posea los conocimientos sobre este tema.

III.6 Comparación entre “La Isla de las Matemáticas” y “La Feria de las Matemáticas”

De manera general el software “La Isla de las Matemáticas”, trata de darle solución a un conjunto de carencias detectadas, específicamente al módulo del software “La Feria de las Matemáticas”, que aborda los ejercicios dirigidos a los estudiantes del Tercer Grado de la Enseñanza Primaria.

Desde el punto de vista metodológico los ejercicios en ambos software se tratan de manera semejante, es decir, estos últimos se presentan a los estudiantes con una estructura muy similar que consta de: un planteamiento del problema, posibles distractores que incluyan la respuesta correcta del ejercicio e imágenes; permitiéndole al estudiante reflexionar sobre cómo solucionar el problema y posteriormente comprobar su respuesta. El software “La Feria de las Matemáticas” es una herramienta que se corresponde con los contenidos de la asignatura de Matemática en el 1^{er} Ciclo, dígase de primero a cuarto grado, mientras que el sistema aquí presentado, está orientado a apoyar la enseñanza de los contenidos impartidos al Tercer Grado, pero desde una perspectiva más innovadora, donde el profesor posee un conjunto de privilegios que le permiten administrar completamente la aplicación, de forma tal que la misma se adapte al diagnóstico del grupo con que trabaja.

Respecto a la implementación computacional de ambas aplicaciones, se debe mencionar que existen, diferencias, tales como:

- El software “La Isla de las Matemáticas” fue implementado con el objetivo de que la aplicación sea multiplataforma y por eso puede ejecutarse sobre diversos equipos y sistemas operativos. Para ejecutarla solo es necesario tener instalada la versión adecuada al sistema operativo que use de la máquina virtual de Java. “La Feria de las Matemáticas” en cambio solo puede ejecutarse en los sistemas operativos de la Microsoft: Windows 95, Windows 98, Windows 2000 y Windows XP.
- Debido a que el software está orientado a los estudiantes, se incluyen animaciones que permiten obtener un entorno ameno. En el caso de “La Isla de las Matemáticas” las animaciones se implementaron de manera interna en la propia aplicación utilizando el lenguaje Java, mientras que en “La Feria de las Matemáticas” las animaciones que se presentan fueron realizadas con el uso del Macromedia Flash 5, lo cual implica que el ActiveX de dicho sistema, debe estar instalado en la computadora donde se ejecutará la aplicación, en caso de no estar, se presentarán problemas que impedirán la ejecución correcta. Para Windows XP esto no es problema ya que el propio sistema operativo lo trae instalado. Para los restantes

sistemas Windows, basta con que tengan dicho componente instalado previamente, para lograr así una correcta ejecución.

Aunque entre ambas aplicaciones existan algunas diferencias se debe referir que también poseen aspectos comunes, por ejemplo, cuentan con un instalador, el cual guiará al usuario final paso a paso y convertirá este proceso en algo muy simple.

III.7 Aportes del software “La Isla de las Matemáticas”

La creación del software Educativo “La Isla de las Matemáticas”, le permite al docente contar con un material auxiliar que puede incluirse en los sistemas de clases, donde los educandos ejercitan a partir de la interacción con la aplicación, las habilidades anteriormente desarrolladas en los complejos de materia numeración, cálculo con las cuatro operaciones matemáticas y el trabajo con fracciones.

Los ejercicios correspondientes al complejo de materia fracciones, son de gran utilidad para los estudiantes y profesores debido a que el libro de texto de la asignatura Matemática de Tercer Grado, no incluye este tipo de ejercicios, solo se cuenta con un cuaderno complementario de la asignatura, el cual no es un material desechable, lo que limita su uso.

La propuesta metodológica para la organización de los sistemas de conocimientos correspondientes a la asignatura de Matemática de Tercer Grado, permite:

- Ofrecer las posibilidades a los docentes de incorporar los ejercicios correspondientes a los complejos de materia, y emplear estos en los sistemas de clases, en correspondencia con el diagnóstico de los estudiantes.
- Presentar ejercicios que se correspondan con el complejo de materia de fracciones, los cuales no aparecen en el libro de texto, ni en el software educativo “La Feria de las Matemáticas”.
- Incorporar nuevos ejercicios en los que se incluyan datos numéricos tomados de la prensa o fechas históricas permitiendo así presentar ejercicios que propicien la interdisciplinariedad.

- Incluir ejercicios que se ajusten al complejo de materia de numeración que respondan al estilo y las exigencias actuales, los cuales no aparecen en el libro de texto, cuaderno o software ya existentes.
- Disminuir la “tendencia a la ejecución”, por parte del escolar, sin haberse detenido previamente a esclarecer las condiciones y encontrar la vía correcta de solución.

III.8 Conclusiones del capítulo

Con el objetivo de que esta aplicación sirva de ayuda a la enseñanza de la asignatura de Matemática, impartida al Tercer Grado de la Enseñanza Primaria, se realizó un estudio exhaustivo en colaboración con profesores, con el objetivo de determinar los aspectos indispensables a ser abordados en la aplicación, logrando así realizar una herramienta que puede utilizarse en el proceso docente educativo de esta materia, sirviendo de ayuda a los profesores, además, de contribuir al logro de un proceso de enseñanza aprendizaje eficiente y acorde a las condiciones científico-técnicas y sociales de Cuba.

CONCLUSIONES

El presente trabajo ha expuesto los detalles de la concepción y realización de un nuevo sistema orientado a la enseñanza de la Matemática en el Tercer Grado de la Enseñanza Primaria, que se traza el objetivo de apoyar el proceso de enseñanza aprendizaje de dicha asignatura. El software elaborado, totalmente funcional, satisface los objetivos propuestos y posee las siguientes características.

- Posee un lenguaje formal, dirigido al profesor, que facilita la gestión de ejercicios matemáticos que se ajusten al diagnóstico del grupo.
- El sistema contiene un intérprete que se ajusta a las especificaciones del lenguaje mencionado anteriormente y facilita la interacción usuario-aplicación.
- El sistema desarrollado se basa en el patrón de arquitectura de software Modelo-Vista-Controlador. Lo que facilita su mantenimiento y futuras modificaciones.
- Se realizó un estudio de los contenidos del programa de la asignatura de Matemática impartida al Tercer Grado, a partir del cual se determinaron los temas a incluir en el sistema.

RECOMENDACIONES

Ningún trabajo, por detallado que se haya hecho, se puede dar por terminado, sobre todo cuando se elabora un producto orientado a una tarea tan compleja como la enseñanza. Por ese motivo se presentan las siguientes recomendaciones, las cuales de cierta manera contribuyen al perfeccionamiento del estado actual del sistema.

- Establecer “La Isla de las Matemáticas”, como software para la enseñanza de la asignatura de Matemática impartida a los estudiantes de Tercer Grado de la Enseñanza Primaria.
- Realizar un experimento pedagógico que valide la herramienta y que sirva de retroalimentación para mejorarlo o ampliarlo.
- Estudiar la posible inclusión del sistema en una red de computadoras, quizás adaptando la herramienta actual a una forma servidora con un cliente que habría que elaborar.

REFERENCIAS BIBLIOGRÁFICAS

- AHO, A. V., LAM, M. S., SETHI, R. & ULLMAN, J. D. 2007. *Compilers: principles, techniques, & tools*, Pearson/Addison Wesley.
- BASCÓN PANTOJA, E. 2011. El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing. *Revista Acta Nova*, 2.
- BELLO, R., GARCÍA, Z., GARCÍA, M. M. & LOBATO, A. R. 2002. Aplicaciones de la Inteligencia Artificial. In: NOCHE, E. L. (ed.) *Aplicaciones de la Inteligencia Artificial*. Jalisco, México: Universidad de Guadalajara.
- BORREGO DÍAZ, O. 2001. *Che el camino del fuego*.
- BOVET, J. & PARR, T. 2008. ANTLRWorks: an ANTLR grammar development environment. *Software: Practice and Experience*, 38, 1305-1332.
- BRACKEEN, D., BARKER, B. & VANHELSUWÉ, L. 2003. *Developing Games in Java™*, New Riders Publishing.
- BROOKSHEAR, J. G. 1993. *Teoría de la computación: lenguajes formales, autómatas y complejidad*.
- BRUNDAGE, B. & GROVER, C. 2006. *Digital Photography: The Missing Manual*
- CAMPOS, S. A., RODRÍGUEZ, J. P. F. & RUIZ, O. B. 1997. Evolución de la enseñanza asistida por computadoras.
- ESCALONA, M. 2005. Los Ordenadores en el proceso de Enseñanza-Aprendizaje de las ciencias. Fundamentos para su utilización. *Revista Iberoamericana de Educación*. Instituto Pre-Vocacional de Ciencias Pedagógicas “Rafael Cruz Pérez”.
- ESTRADA DOALLO, M., SÁNCHEZ SANTISTEBAN, J. & LIMIA TORRES, A. 2002. LA ELABORACIÓN DE MEDIOS DE ENSEÑANZA CON EL PROGRAMA DE GEOMETRÍA DINÁMICA CABRI GEOMETRE II. 6.
- FERNÁNDEZ, G. & SÁEZ, F. 1995. *Fundamentos de la informática: lógica, autómatas, algoritmos y lenguajes*.
- FERNÁNDEZ, Y. & DÍAZ, Y. 2012. Patrón Modelo-Vista-Controlador. *Telemática*.
- GÁLVEZ, S. 2005. *Compiladores*.
- GALVIS, A. 2006. Software Educativo Multimedia: Aspectos críticos en su ciclo de vida.

- HACKER, S. 2000. *MP3: The Definitive Guide*
- HERRERA, L. & MENDOZA, N. 2006. ¿Tecnología Educativa o Nuevas Tecnologías en Educación? *Revista Cognición*.
- ISASI, P., MARTÍNEZ, P. & BORRAJO, D. 2001. *Lenguajes, Gramáticas y Autómatas. Un enfoque práctico*.
- IVAR JACOBSON, G. B., JAMES RUMBAUGH 2000. *El proceso unificado de desarrollo de software*.
- KELLEY, D. 1995. *Teoría de autómatas y lenguajes formales*.
- KNUDSEN, J. 2001. *Java 2D Graphics*, O'Reilly.
- LAU APÓ, F., ARIAS LEYVA, G., RODRÍGUEZ IZQUIERDO, J., VILLEGAS JIMÉNEZ, E., COPPERY CANO, S., RIQUENES TARRAGÓ, A., RODRÍGUEZ BEN, J., MENDOZA RODRÍGUEZ, M., GUZMÁN DE ARMAS, L. & GONZÁLEZ FERNÁNDEZ, N. 2001. *Programa director de las asignaturas priorizadas para la enseñanza primaria*.
- LEÓN ROLDÁN, T., BARCIA MARTÍNEZ, R., ALBARRAN PEDROSO, J., MARTÍNEZ SOTELO, Y., PARETS GÓMEZ, A., BERMUDÉZ BRITO, J. E., GONZÁLEZ ROJAS, R., TIMONEDA VALDÉS, M. & ÁLVAREZ PÉREZ, M. 2012. *Sugerencias de trabajo metodológico para el fortalecimiento de la matemática en la educación primaria*., Cuba.
- LEZCANO, M. & SOLER, Y. 2009. Consideraciones sobre la tecnología educativa en el proceso de enseñanza-aprendizaje. Una experiencia en la asignatura Estructura de Datos. *Revista Iberoamericana de Educación*.
- LOUDEN, K. C. 1997. *Compiler Construction: Principles and Practice*.
- MILBURN, K. 2006. *Digital Photography: Expert Techniques, 2nd Edition*
- MOYA, A. M. 2009. Las nuevas tecnologías en la educación. *Innovación y experiencias educativas*. Granada, 2009, n. 24, noviembre.
- MULHOLLAND, A. & MURPHY, G. 2003. *Java 1.4 Game Programming*
- OAKS, S. & WONG, H. 2004. *Java Threads*, O'Reilly.
- PANQUEVA, G. & ALVARO, H. 1994. Mejoramiento educativo apoyado con informática: enfoque estratégico. *Informática educativa*, 7, 49-91.

- PARR, T. 2009. ANTLR parser generator. *Online Stand Dezember*.
- PAUL, J. 2005. *Digital Video Hacks* O'Reilly.
- PÉREZ MARQUÉS, G. 2005. Impacto de las TIC en la educación: funciones y limitaciones 10-15.
- PRESSMAN, R. S. 2006. *Ingeniería del software. Un enfoque práctico*.
- REYNOSO, C. B. 2004. Introducción a la Arquitectura de Software. *Universidad de Buenos Aires*, 33.
- ROBBINS, K. A. & ROBBINS, S. 2003. *Unix™ Systems Programming: Communication, Concurrency, and Threads*, Prentice Hall PTR.
- RODRÍGUEZ FERNÁNDEZ, M. M., WILLIAM MESA, A., CHAVIANO CONDE, R. & RAMÍREZ GARCÍA, E. C. 2007. PAGINA WEB PARA LA ASIGNATURA MATEMÁTICA I EN INGENIERÍA MECÁNICA. 5.
- RODRÍGUEZ IZQUIERDO, J., MONTEAGUDO RODRÍGUEZ, D. & RODRÍGUEZ BELLO, G. 2001. *Programas. Tercer grado.*, Cuba.
- ROELOFS, G. 1999. *PNG: The Definitive Guide*, O'Reilly & Associates.
- SILBERSCHATZ, A., GAGNE, G. & BAER, P. 2013. *Operating System Concepts*.
- SOMMERVILLE, I. 2005. *Ingeniería del software*.
- TANENBAUM, A. S. 2006. *Operating Systems Design and Implementation*.
- URRETAVIZCAYA LOINAZ, M. 2001. Sistemas Inteligentes en el ámbito de la educación. *Revista Iberoamericana de Inteligencia Artificial*. Valencia, España: Asociación Española para la Inteligencia Artificial.
- VALDÉS, M. A., MENÉNDEZ, L. M. & VALDÉS, V. G. 2001. Utilización de textos y gráficos en la enseñanza asistida por ordenador. *REVISTA PIXEL-BIT. NÚMERO 17*
- VIDAL LEDO, M., NOLLA CAO, N. & DIEGO OLITE, F. 2009. Plataformas didácticas como tecnología educativa. *Educación Médica Superior*, 23.
- VILLALÓN INCHÁUSTEGUI, M., PEÑA GÁLVEZ, R. L., LEÓN FIGUERAS, N., GAREA ALONSO, L., BELLO DOMÍNGUEZ, M., VERELA PILOTO, L. & RIZO CABRERA, C. 1990. *Orientaciones metodológicas.*, Cuba.
- VIVANCOS, E. 2000. *Compiler I: Introduction to phase the analysis*.

ZILBERSTEIN TORUNCHA, J., HERRERO TUNIS, E., BOLLOTO CARMONA, G.,
CASTAÑEDA HEVIA, Á. E., CAÑAS LOUZAU, T., FERNÁNDEZ, A. M.,
VALDÉS MONTALVO, N., COLLAZO DELGADO, R. & RODRÍGUEZ
LAMAS, R. 2006. *Preparación pedagógica integral para profesores integrales.*,
Cuba.