

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DPTO. DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**"DESARROLLO DE UN SOFTWARE MULTIMEDIA  
EDUCATIVO PARA EL ÁREA DE MATEMÁTICAS DE  
QUINTO AÑO DE EDUCACIÓN BÁSICA"**

**Previa a la obtención del Título de:**

**INGENIERO EN SISTEMAS E INFORMÁTICA**

**POR: RUTH GARCÍA GAVILANES**

**Sangolquí, 20 de Junio del 2007**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue realizado en su totalidad por la Srta. Ruth Olimpia García Gavilanes como requerimiento parcial a la obtención del título de **INGENIERO EN SISTEMAS E INFORMÁTICA**

---

Fecha

---

**Ing. Carlos Rojas**

**DIRECTOR DE TESIS**

## **DEDICATORIA**

Esta tesis va dedicada principalmente a Dios quien nunca me ha abandonado. Segundo, a mis padres, porque querían mucho ver realizados mi sueño de ser Ingeniera. A mi familia materna y paterna pero en especial a mi tía Cecilia, quien me dio el ejemplo que nunca es tarde para ir en búsqueda de nuestros sueños.

Dedico también esta tesis a mis amigos que siempre protestaban porque no tenía mucho tiempo para salir con ellos, ahora ya tendré más tiempo, al menos hasta la maestría.

## **AGRADECIMIENTOS**

Son en los momentos difíciles que el ser humano mide su fuerza para seguir adelante en sus luchas. La derrota no es lo que determina quien uno es, si no el valor de levantarse cuantas veces sea necesario con las ganas de seguir adelante y sin perder la esperanza de un mañana mejor, aunque el camino este lleno de piedras. Como mi padre siempre dice, como el ave Fénix.

Agradezco mucho a mis padres por su apoyo en aquellos momentos que perdía la calma y el control por tanta burocracia involucrada en todo este proceso de graduación, especialmente a mi madre quien me ayudó muchísimo a arreglar mis papeles. Agradezco también a algunos de mis amigos por haberme ayudado a recuperar algo de la fuerza que había perdido en el camino y tener la paciencia de escuchar mis problemas y dudas. Agradezco también a Dios por darme la seguridad de que siempre hay esperanzas.

Una de las satisfacciones más grandes es poder ver que se ha terminado una asignatura pendiente, hoy pude terminar una de las tantas que tengo en mente, al menos es una menos y eso se siente bien.

**Ruth García Gavilanes**

# ÍNDICE DE CONTENIDOS

<b>CAPITULO 1</b> .....	3
<b>1 INTRODUCCIÓN</b> .....	3
<b>1.1 Prologo</b> .....	3
<b>1.2 Objetivos</b> .....	5
1.2.1 Objetivo General.....	5
1.2.2 Objetivos Especificos.....	6
<b>1.3 Alcance</b> .....	6
<b>1.4 Metodología</b> .....	7
1.4.1 Metodología Tecnológica.....	7
1.4.2 Metodología Pedagógica.....	7
<b>CAPITULO 2</b> .....	9
<b>2 MARCO TEÓRICO</b> .....	9
<b>2.1 Pedagogía Constructivista</b> .....	9
2.1.1 La importancia de tener metodologías de aprendizaje.....	9
2.1.2 Raíces de la Pedagogía Constructivista.....	9
2.1.3 Concepto y Tipos de Constructivismo.....	10
2.1.4 Constructivismo aplicado a la educación.....	10
2.1.5 Por qué el Constructivismo debe ser adoptado.....	11
2.1.6 El Constructivismo en el programa de matemáticas.....	12
<b>2.2 Especificación del Alcance de los Temas de Matemáticas</b> .....	13
<b>2.3 Especificación de Requerimientos del software</b> .....	14
2.3.1 Importancia de la Especificación de Requerimientos del Software.....	15
2.3.2 Características de un ERS.....	15
2.3.3 IEEE 830.....	15
<b>2.4 Fundamentos Multimedia</b> .....	17
2.4.1 Concepto de Sistemas Hipermediales.....	17
2.4.2 Importancia de una Metodología para el Desarrollo de Sistemas Hipermediales.....	18
2.4.3 Metodologías Existentes para el Desarrollos de Sistemas Hipermediales?.....	19
2.4.4 Metodología Object Oriented Hypermedia Design Methodology (OOHDM).....	19
2.4.5 Obtención de Requerimientos.....	20
2.4.6 Diseño Conceptual.....	22
2.4.7 Diseño Navegacional.....	24
2.4.8 Diseño de interfaces abstractas.....	30
2.4.9 Implementación.....	34

<b>CAPITULO 3.....</b>	<b>35</b>
<b>3 ANÁLISIS .....</b>	<b>35</b>
<b>3.1 Especificación de Requerimientos de Software .....</b>	<b>35</b>
3.1.1 Alcance.....	35
3.1.2 Definiciones, Acrónimos y Abreviaciones.....	36
3.1.3 Acrónimos .....	37
3.1.4 Abreviaciones.....	37
3.1.5 Descripción Global .....	38
3.1.6 Requerimientos Específicos .....	41
<b>3.2 Obtención de Requerimientos.....</b>	<b>44</b>
3.2.1 Identificación de Roles y Tarea .....	44
3.2.2 Especificación de Escenarios.....	45
3.2.3 Especificación de casos de uso .....	47
3.2.4 Especificación de UIDs.....	48
3.2.5 Validación de Casos de Uso y UID.....	50
<b>3.3 Diagramas .....</b>	<b>50</b>
3.3.1 Diseño Conceptual .....	50
3.3.2 Diseño De Navegación.....	50
3.3.3 Diseño De Interfaces Abstractas .....	52
<b>CAPITULO 4.....</b>	<b>57</b>
<b>4 IMPLANTACIÓN .....</b>	<b>57</b>
<b>4.1 PLAN DE IMPLANTACIÓN .....</b>	<b>57</b>
4.1.1 Determinación de recursos en implementación del software ...	57
4.1.2 Selección de Recursos.....	58
4.1.3 Características del Sistema.....	58
4.1.4 Manual De Usuario.....	61
4.1.5 CRONOGRAMA .....	62
4.1.6 RESPONSABLES .....	62
<b>CAPITULO 5.....</b>	<b>64</b>
<b>5 PLAN DE PRUEBAS .....</b>	<b>64</b>
<b>5.1 Necesidad de Comprobar la Calidad del Software .....</b>	<b>64</b>
5.1.1 Aceptación por parte del cliente final.....	65
5.1.2 Pasos a seguir para las pruebas .....	65
5.1.3 Definir Pruebas de Caja Blanca.....	66
5.1.4 Definir Pruebas de Caja Negra.....	67
5.1.5 Pruebas con el cliente (profesor y estudiante) .....	67
5.1.6 Criterios de Calidad.....	67
5.1.7 Definir Plan de Pruebas.....	68

<b>5.2</b>	<b>Ejecución De Pruebas</b> .....	68
5.2.1	Revisión por Director y Codirector.....	68
5.2.2	Pruebas de Caja Blanca.....	68
	Vamos a considerar tres casos de pruebas de caja blanca.....	68
5.2.3	Pruebas de Caja Negra.....	82
5.2.4	Pruebas con el cliente (Profesor y estudiantes) .....	83
<b>5.3</b>	<b>Resultados de Pruebas</b> .....	86
<b>6</b>	<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	90
<b>6.1</b>	<b>CONCLUSIONES</b> .....	90
<b>6.2</b>	<b>RECOMENDACIONES</b> .....	92
<b>7</b>	<b>BIBLIOGRAFÍA</b> .....	95
<b>8</b>	<b>ANEXOS</b> .....	98

## LISTADO DE TABLAS

Tabla 2.1	Temas de Matemáticas considerados en el Software.....	13
Tabla 3.1	Identificación de Roles y Tareas .....	44
Tabla 4.1	Especificación de requerimientos de hardware y software .....	57
Tabla 5.1	Pruebas de Caja Negra: Funcionalidad del Software.....	83
Tabla 5.2	Pruebas con el Cliente: Funcionalidad del Software .....	84
Tabla 5.3	Pruebas con el Cliente: Aspectos pedagógicos .....	85
Tabla 5.4	Pruebas Globales: Aspectos Globales (niños).....	86
Tabla 5.5	Resultados: Caja Negra y Criterios del profesor .....	87

## LISTADO DE FIGURAS

Figura 2.1	Etapas OHDM.....	19
Figura 2.2	Ejemplos de UID.....	22
Figura 2.3	Modelo Conceptual para una revista en línea.....	24
Figura 2.4	Sintaxis para definir a los atributos de los nodos .....	26
Figura 2.5	Definición de nodo .....	26

Figura 2.6. Esquema navegacional de la revista en línea.....	27
Figura 2.7 Diagrama de Contexto de una revista en línea.....	29
Figura 2.8 Clase “EnContexto” para “Historia” dentro de contextos de “Destacados.”.....	30
Figura 2.9 ADVs pueden ser agregaciones. ....	31
Figura 2.10 Diagrama de Configuración .....	33
Figura 2.11 ADV y su relación con objetos de interfaz real. ....	33
Figura 3.1 UID1.....	48
Figura 3.2 UID2.....	49
Figura 3.3 UID3.....	49
Figura 3.4 UID4.....	49
Figura 3.5 UID5.....	49
Figura 3.6 Diseño Conceptual.....	50
Figura 3.7 Esquema de Navegación de Clases .....	51
Figura 3.3 Esquema De Navegación De Contextos.....	52
Figura 3.9 Vista General .....	53
Figura 3.10 Menu Principal .....	54
Figura 3.11 ADV Tema .....	54
Figura 3.12 ADV Evaluación.....	55
Figura 3.13 ADV Implementación 1 .....	56
Figura 3.14 ADV Evaluación.....	56
Figura 4.1 Sistema basado en Módulos.....	59
Figura 4.2 Textos y Sonidos .....	60
Figura 4.3 Niño indio y niña mulata.....	61
Figura 4.4 Pescado tomate.....	61
Figura 5.1 Entrada De Usuario y autenticación: No Se Puede Cargar XML.....	69
Figura 5.2 Entrada de usuario y autenticación: Datos incorrectos .....	70
Figura 5.3 Entrada de usuario y autenticación: datos correctos.....	70
Figura 5.4 Ejercicios de Sistema Decimal: números cargados de más de 5 dígitos .....	74
Figura 5.5 Ejercicios de Sistema Decimal: números ingresados de más de 5 dígitos .....	74
Figura 5.6 Ejercicios de Sistema Decimal: números ingresados de más de 5 dígitos .....	75



Figura 5.7 Ejercicios de Sistema Decimal: se ingresa número incorrecto ....	75
Figura 5.8 Ejercicios de Sistema Decimal: se ingresa la respuesta correcta	76
Figura 5.9 Minitest: Respuesta correcta es mal ubicada en XML .....	80
Figura 5.10 Minitest: Imagen no puede ser cargada.....	80
Figura 5.11 Minitest: respuesta incorrecta .....	81
Figura 5.12 Minitest: se llega al final de evaluación.....	81
Figura 5.13 Criterios del profesor y pruebas de caja negra .....	87
Figura 5.14 Opinión de niños con respecto al Software.....	88
Figura 5.15 Suma de opiniones de los niños con respecto al Software .....	89

## **LISTADO DE ANEXOS**

ANEXO A - PLANIFICACIÓN MICROCURRICULAR PARA EL ÁREA DE MATEMÁTICAS DEL QUINTO AÑO DE BÁSICA .....	98
ANEXO B: MANUAL DE USUARIO.....	108

## RESUMEN

El presente documento tiene el objetivo de desarrollar un software de matemáticas para el quinto año de básica. El Capítulo I cubre rápidamente los conceptos de Software Educativo además de mencionar aquellos que están vigentes en Quito y que han sido resultado de iniciativas similares al de la ESPE COMUNITARIA. Por otro lado, este capítulo plantea los objetivos y el alcance y hace una muy breve introducción a la Metodología Tecnológica y Pedagógica que se usa en este proyecto.

El Capítulo II cubre temas relacionados al marco teórico de la Metodología Constructivista y cómo ésta es aplicada a la educación y al software multimedia. Adicionalmente se explica por qué debe ser adoptada en la enseñanza y se lista los temas de matemáticas que el sistema embarcará. Por otro lado, se detallan los temas a tratar en la Especificación de Requerimientos del Software. Finalmente se presenta conceptos de Sistemas Hipermediales y explicación de toda las cinco fases de la metodología OOHDM (Obtención de Requerimientos, Diseño Conceptual, Diseño Navegacional, Diseño de Interfaces abstractas e Implementación).

En el Capítulo III se prosigue a realizar, de acuerdo al sistema desarrollado, todo lo que fue explicado en el capítulo anterior.

El Capítulo IV trata acerca de todos los detalles necesarios para la Implantación del Sistema, tales como recursos necesarios, las características del sistema, Manual de Usuario, cronograma y responsables. Por otro lado, el Capítulo V realiza el plan de pruebas del sistema desarrollado (Caja Negra y Caja Blanca) y proporciona gráficos estadísticos acerca de los resultados obtenidos en la implantación del software en la Escuela Fiscal Acosta Soberón.

Finalmente se proporciona varias interesantes conclusiones y recomendaciones.

## **ABSTRACT**

The purpose of this document is to develop an Educational Math Software for the fifth grade of elementary school. The first chapter briefly covers concepts related to Educational Software products and present those being used by some schools in Quito, which are also the result of initiatives similar to ESPE COMUNITARIA. On the other hand, the objectives and scope of this project are also found in this chapter as well as an introduction to the Technological and Pedagogical Methodology that has been used.

The second chapter covers the theory of Constructivist Methodology and how it is applied in education and multimedia software products. It also explains why this methodology should be used in teaching and lists the math subjects that will be present in the system. Furthermore, the topics covered in the Software Requirements Specification are listed. Finally, the concepts related to Hypermedia Systems and the five phases of the OOHDM methodology are explained (Obtaining Requirements, Conceptual Model Design, Navigational Design, Abstract Interface Design and Implementation).

The third chapter carries out everything that was specified in the previous chapter according to the system developed in this project.

The fourth chapter presents the aspects that were needed in the implementation of the System such as technical requirements and resources, characteristics of the system, User Manual, schedule and responsibilities. On the other hand, the chapter fifth executes the user acceptance testing plan (Black and White Box) and presents statistical graphics of the results obtained after the installation of the software in "Escuela Fiscal Acosta Soberón."

Finally, the conclusions and recommendations are listed.

# CAPITULO 1

## INTRODUCCIÓN

### 1.1 Prologo

Se denomina **Software educativo** todo programa para ordenador creado con la finalidad de ser utilizado como medio didáctico en el proceso de enseñanza y de aprendizaje. Los programas educativos, independiente del área al que estén enfocados, comparten cinco **características esenciales**: didácticos, utilizan el ordenador, son interactivos, individualizan el trabajo y son fáciles de usar.

En el Ecuador hay varias empresas que se dedican a desarrollar Software Educativo en varias áreas empleando personajes y sonidos de la cultura ecuatoriana. Algunos sistemas con similares objetivos a la de la ESPE COMUNITARIA son la iniciativa EDUFUTURO de la Prefectura de Pichincha y la Fundación Chasquinet.

Chasquinet sin embargo, a pesar de ofrecer sistema Multimedia, no ofrece descargas gratis en su portal de sistemas hipermediales específicos para cada año escolar. Por otro lado EDUFUTURO si ofrece este tipo de software especializado para la escuela básica en las áreas cognitivas y es libre de costo. Ellos desarrollan contenidos curriculares de segundo a séptimo año de educación básica en las áreas cognitivas. Este software se caracteriza por presentar material educativo contextualizado a las características de la población ecuatoriana, de forma lúdica y asequibles a niños y niñas de hasta 12 años. En consideración a la población objetivo de este material, este software contiene gráficos en dos y tres dimensiones, animaciones interactivas; fotografías panorámicas interactivas, música, locuciones y sonidos de fondo, y juegos interactivos. Este software es una guía para el docente por lo que se trabaja para que cada maestra o maestro enriquezca este material con contenidos del entorno inmediato de la comunidad.

De la misma forma, la ESPE, conciente de la importancia de la educación y de la difícil situación del país donde la igualdad en el nivel educativo es difícil, ha deseado crear el programa ESPE-COMUNITARIA, el mismo que busca fortalecer la enseñanza de niños de

5 a 12 años a través de programas hipermediales sin costo destinados en un principio a escuelas de bajos recursos económicos.

El presente proyecto busca proporcionar un programa de multimedia en el área de matemáticas para niños de quinto grado basado en los temas planteados por el Consejo Nacional de Educación. Este programa hace uso de dibujos, animaciones y sonidos para intentar apoyar en la educación de matemática en una forma interactiva, deductiva y divertida. Además, el software multimedia educativo intentará presentar personajes, paisajes, modos de hablar, y costumbres del Ecuador. Esto hace que los niños y niñas se identifiquen rápidamente con las actividades que el software propone además de dar valor a la cultura ecuatoriana.

De esta forma, se procurará sacar la mayor ventaja del programa para que se pueda hacer del ordenador un laboratorio matemático que permitirá experimentar, suplir carencia en el bagaje matemático del alumno, desarrollar la intuición, conjeturar, comprobar, demostrar y, en definitiva “ver las situaciones matemáticas” de una forma práctica haciendo de este proyecto un valioso instrumento didáctico.

En el presente momento, se han desarrollado ya programas de matemáticas para tercero, cuarto y sexto grado de primaria. El presente trabajo, como ya se mencionó, se dedicará apoyar la educación de matemáticas del quinto grado de básica.

Las matemáticas ha sido considerada como una disciplina de un gran valor formativo además de algo necesario, como contenido, para cualquier tipo de estudio que se realice. Es por esto que la enseñanza de las matemáticas en la educación básica requiere en su gran mayoría más esfuerzo por parte de los profesores para lograr la atención de los estudiantes, pues esta no puede enseñarse en los primeros niveles como una teoría formal, abstracta, porque el niño no es capaz de entenderla y tampoco ve la necesidad de una teoría de este tipo. Como hacer que esta materia sea más fácil, comprensible y divertida?

El presente proyecto usa las ventajas de la multimedia y la tecnología para intentar facilitar el proceso de enseñanza de las matemáticas del quinto año de educación. El programa desarrolla actividades prácticas, intuitivas, relativas sobre todo a números, al espacio y a la medida, actividades que son esenciales pues no sólo el estudiante se

sentirá enormemente motivado, sino que se ve obligado a utilizar nociones matemáticas en actividades que reflejan la realidad. Hay que aclarar que este software educativo no es una solución sino una herramienta de trabajo que apoya el proceso de enseñanza-aprendizaje de las matemáticas.

El programa es un paquete interactivo de las matemáticas de quinto de básica que ayuda a estudiantes a visualizar y a entender algunos de los temas más importantes entre la teoría impuesta por el Ministerio de Educación para este nivel. Mientras observan y exploran las imágenes y sonidos del programa, los estudiantes aprenden, practican, se ejercitan y son evaluados.

Es importante hacer mención que el software contará con un submódulo de autoevaluación, el cual tendrá la finalidad de evaluar los conocimientos adquiridos en cada unidad. La metodología empleada para la elaboración del programa de matemáticas multimedia es el OOHDM (Object Oriented Hypermedia Design Model) que es una reciente metodología de desarrollo de Sistemas Hipermediales que incorpora las ideas de orientación a objetos, permitiendo modelar las especificaciones de la documentación hipermedia que posteriormente vaya a ser producida.

Cabe mencionar que el software trabaja desde un enfoque pedagógico, el mismo que tendrá en cuenta los conocimientos didácticos aplicados a las herramientas multimedia. El sistema que se elaborará podrá ser usado por escuelas, específicamente para el quinto año de educación básica, el mismo que se basará en los temas más importantes del pensum de estudio establecido por el Ministerio de Educación para Matemáticas.

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

- Desarrollar un Sistema Multimedia para la Enseñanza de Matemáticas que servirá como herramienta de apoyo para facilitar la enseñanza a los alumnos del quinto año de educación básica.

### **1.2.2 Objetivos Específicos**

- Implementar la metodología pedagógica, en este caso la constructivista, en el desarrollo del Sistema multimedia de Enseñanza de Matemáticas.
- Incorporar en el Sistema Hipermedial algunos elementos de la cultura ecuatoriana, con la finalidad que el estudiante pueda asociar su entorno con el aprendizaje de Matemáticas
- Realizar el análisis y especificación de requerimientos, usando la norma IEEE 830.
- Determinación específica del submódulo de autoevaluación del Aprendizaje con preguntas dinámicas y cambiantes, con el cual se podrá analizar el nivel de conocimientos adquiridos por los alumnos sin que las preguntas sean las mismas en todas las ocasiones.
- Implementar el sistema

### **1.3 Alcance**

El desarrollo del presente sistema se basa en la elaboración práctica y didáctica de un programa que intenta apoyar la enseñanza de Matemáticas orientada a los niños de 5<sup>to</sup> año de educación básica. Además se ha implementado el sistema en la escuela fiscal “Acosta Soberón” cuyos resultados se detallan en el capítulo 4, aunque cabe resaltar que el software no ha sido elaborado específicamente para esta escuela. El tiempo esperado para el desarrollo de este sistema fué de cuarenta y un días pero el tiempo usado realmente fue de 90 días incluido la implementación, elaboración de manuales y correcciones.

Por otro lado, este tipo de aplicación tiene un alto grado de flexibilidad y estabilidad, por lo que la determinación y seguimiento de la metodología OOHMD durante el desarrollo del sistema ha permitido obtener un producto acorde a los requerimientos planteados, logrando así la utilización del mismo en un entorno real.

Cabe mencionar que cada uno de los contenidos contempla una exposición teórica, juegos para ejercitar la teoría y una auto-evaluación cuyas preguntas serán cambiantes. Además se puede acceder desde cualquier parte del sistema a la ayuda, mapa de navegación e Índice.

La aplicación a desarrollarse cubre los temas expuestos a continuación:

#### **a) Sistema Numérico**

Números decimales

- Operaciones: suma, resta, multiplicación y división exacta e inexacta
- Aplicaciones

Números romanos: lectura y escritura

## **b) Sistema de Funciones**

Operaciones con conjuntos: unión, intersección y diferencia.

Operadores combinados de suma, resta y multiplicación.

Ubicación en una cuadrícula.

## **c) Sistema Geométrico y de Medida**

Áreas de triángulos y cuadriláteros

Identificación y caracterización.

## **d) Sistema de Estadística y Probabilidad**

Representación e interpretación de diagramas de barras.

## **1.4 Metodología**

### **1.4.1 Metodología Tecnológica**

La metodología OOHDM (*Object Oriented Hypermedia Design Method*) fue propuesta por D. Schwabe y G. Rossi (1995) y establece que el desarrollo de un Hiperdocumento es un proceso de cuatro fases en el que se combinan diferentes estilos de desarrollo como el incremental, iterativo y prototipado. Las tres primeras fases son de diseño, en las que se obtiene un conjunto de modelos orientados a objeto que describen el documento que será construido en la última fase. Estas cuatro fases son explicadas detalladamente en el Marco Teóricos (capítulo 2).

### **1.4.2 Metodología Pedagógica**

El aprendizaje constructivista consiste en observar, manipular, analizar la relación causa-efecto y el desarrollo de los hechos, y buscar soluciones para los problemas presentes.



Esta pedagogía plantea usar instrumentos para que el alumno de soluciones a partir de su saber previo, promoviendo así el desarrollo de creatividad e independencia de los mismos. Esta metodología también es explicada en detalle en el Marco Teórico (capítulo 2).

## **CAPITULO 2**

### **MARCO TEÓRICO**

#### **2.1 Pedagogía Constructivista**

##### **2.1.1 La importancia de tener metodologías de aprendizaje**

Para poder transmitir conocimiento no basta con tener claro qué se enseñará, sino también cómo se lo enseñará. Lo primero que se debe saber es que no todo puede ser enseñado en cualquier momento. Se debe establecer una planificación del trabajo a realizar en la que se establezca un orden de ejecución, es por esto que necesito una metodología de enseñanza.

Una vez que se tiene claro qué es lo que se va a enseñar, entra en juego el cómo enseñarlo. Cada institución educativa puede escoger distintos caminos (modo de enseñar) para llegar a un mismo objetivo (adquisición de un determinado fundamento) y aquí entrar en juego un sin fin de cosas. Es por esto que se con el pasar del tiempo se han propuesto varias metodologías de enseñanza a ser adoptadas, cada una con un fundamento y técnica. El software a s elaborado en esta tesis está basada en la “metodología constructivista”.

##### **2.1.2 Raíces de la Pedagogía Constructivista**

La Teoría Constructivista de aprendizaje tiene como antecedente el llamado constructivismo filosófico. De esta forma, el concepto de Constructivismo no es un concepto nuevo y ha sido aplicado en sociología, antropología y en la psicología cognitiva además de la educación. Algunos aspectos de constructivismo pueden ser encontrados entre los trabajos de varios filósofos antiguos. Sin embargo, Jean Piaget (1746-1827) fue responsable por la más importante filosofía del Constructivismo aunque Henrich Pestalozzi (1746-1827), también de Suiza, llegó a muchas conclusiones similares un siglo antes (Nieda & Macedo, 1997).

El constructivismo pedagógico, el cual aparece previamente al constructivismo psicológico de Piaget, surge a través del movimiento conocido como "Escuela Activa" del cual se identifican como algunos de sus representantes más prominentes a Montessori, Dewey, Declory y Claparede. Estos pedagogos enfatizaban el "principio de la actividad", es decir, argumentaban que el niño aprendía en la medida en que hace y experimenta, ambos elementos como parte de su proceso vital de desarrollo.

### 2.1.3 Concepto y Tipos de Constructivismo

Aunque hay una excesiva variedad de matices e interpretaciones, se puede denominarse como teoría constructivista toda aquella que entiende que el conocimiento es el resultado de un proceso de la realidad que tiene su origen en la interacción entre las personas y el mundo. Por tanto, la idea central reside en que la elaboración del conocimiento constituye una modelización más que una descripción de la realidad. Es en base a este concepto que el programa multimedia se basa.

### 2.1.4 Constructivismo aplicado a la educación

Aún teniendo en cuenta la amplia variedad de versiones que coexisten bajo la metodología del constructivismo, pueden destacarse unas pocas ideas fundamentales que caracterizan a esta corriente en la educación. Entre ellas están los siguientes enfoques:

- Considera primordialmente como punto de apoyo la estructura conceptual de cada estudiante, parte de las **ideas previas** que el estudiante tiene respecto al contenido de la clase.
- Prevé el cambio conceptual que surgirá como resultado de la construcción activa del nuevo concepto.
- Confronta los conceptos e ideas previas con los conceptos nuevos, que se busca aprendan.
- Aplica el nuevo concepto, a situaciones concretas y reales con la finalidad de facilitar la transferencia.

Tales características, son importantes en la medida en que el profesor quiera promover en su clase experiencias de aprendizaje constructivistas.

Como consecuencia de esa concepción del aprendizaje, el constructivismo ha aportado metodologías didácticas propias como los mapas y esquemas conceptuales, la idea de actividades didácticas como base de la experiencia educativa, ciertos procedimientos de identificación de ideas previas, la integración de la evaluación en el propio proceso de aprendizaje, los programas entendidos como guías de la enseñanza, etc.

### **2.1.5 Por qué el Constructivismo debe ser adoptado en la enseñanza**

Como se vio, el conductivismo considera al alumno como un agente pasivo, acumulador de información, cuyo aprendizaje depende fundamentalmente de la actividad del profesor. En el caso de la enseñanza de las ciencias, los requisitos que debe cumplir el profesor son el conocimiento de la estructura de la disciplina y la capacidad para transmitirla siguiendo su lógica. El constructivismo se caracteriza por su rechazo a estas formulaciones de la enseñanza, donde se esperaba que el sujeto, en su proceso de aprendizaje, se comportara como un inventor. Por el contrario, el constructivismo rescata, por lo general, la idea de enseñanza transmisiva o guiada, centrando las diferencias de aprendizaje entre lo significativo y lo memorístico.

De esta forma, enseñar se convierte en el establecimiento y mantenimiento de un lenguaje y medios de comunicación entre el profesor y estudiantes así como también entre estudiantes. La perspectiva constructivista procura que los estudiantes puedan analizar, investigar, colaborar, compartir y generar conocimientos basados en lo que ellos ya saben en lugar de procurar que los estudiantes repitan de memoria los hechos, conocimientos y procesos. Para hacer esto eficientemente, el profesor necesita ser un aprendiz e investigador, esforzándose para un mayor conocimiento del medio y de los participantes en una situación de enseñanza dada para que de esta forma puedan ajustar continuamente sus acciones a comprometer a los estudiantes a aprender, usando el constructivismo como referente.

### 2.1.6 El Constructivismo aplicado al programa multimedia de matemáticas.

El presente proyecto, el programa de multimedia de matemáticas para quinto grado, usa una perspectiva constructivista al intentar ayudar al profesor en el proceso de enseñanza de las matemáticas. Saiz y Acuña (2006) mencionan que **Douady** hace una caracterización de este proceso de aprendizaje en una “situación problema”:

- *El alumno ha de poder introducirse en la resolución del problema y ha de poder considerar lo que es una solución posible.*
- *Los conocimientos del alumno han de ser, en principio, insuficientes para resolver el problema.*
- *La “situación problema” ha de permitir al alumno decidir si una solución determinada es correcta o no.*
- *El conocimiento que se desea que el alumno adquiera (“construya”) ha de ser la herramienta más adecuada para resolver el problema al nivel de conocimientos del alumno.*
- *El problema se ha de poder formular en diferentes “cuadros” (por ejemplo, cuadros físico, geométrico, algebraico) entre los que han de poderse establecer correspondencias.*

De esta manera, si se quiere que los alumnos aprendan matemática haciendo matemática, se debe organizar situaciones que enfrenten a los alumnos a genuinos problemas que les permitan utilizar sus conocimientos previos, elaborar conjeturas, ponerlas a prueba, etc.

*Por ejemplo, una editorial envía libros a las librerías de distintas provincias. Los libros se colocan en cajas donde caben 5 libros. El lunes de esta semana armaron 8 cajas. ¿Cuántos libros mandaron?*

Para resolverlo es necesario adaptar el conocimiento aditivo a esta nueva situación, donde los datos en juego no constituyen cantidades homogéneas, como en el caso de la suma. Si no se modificara el conocimiento anterior, se vería aparecer la resolución errónea:

$$5 + 8 = 13.$$

Problemas como este permiten que se empiece a identificar un tipo especial de problemas que luego se llamará multiplicación. Sin embargo, no se lograrán mayores avances en este conocimiento si no se realiza, en forma paralela, un desarrollo de técnicas de cálculo para este tipo especial de situaciones. Es por esto que más una vez se recalca que este software no es un medio de aprendizaje de la matemáticas de quinto año de básica, si no una **herramienta** eficiente para la obtención de esta.

## 2.2 Especificación del Alcance de los Temas de Matemáticas

En el quinto grado se cubren varias áreas de las matemáticas durante un año. El software a ser presentado cubrirá algunas de estas áreas con tutoriales y ejercicios basados en una metodología constructivista (ver sección 2.1.6) . A seguir se explica cuales son las áreas que los niños cubren durante el año y cuales de estos temas el software multimedia tomará en cuenta.

Tabla 2.1 Temas de Matemáticas considerados en el Software

Temas dictados en el quinto año de básica		Temas del S Multimedia Matemáticas
1.Sistema Numérico	1.1 Sistema de numeración romana	X
	1.2 Numeración maya	
	1.3 Sistema de numeración posicional decimal	X
	1.4 Números naturales hasta 9999	
	1.5 El 10000 o a la decena de mil	
	1.6 Números hasta 99999	
	1.7 Adición y sustracción. Propiedades	X
	1.8 Multiplicación simple	X
	1.9 Multiplicación por 10, 100, 1000, 10000, 1 y 0	
	1.10 Propiedades de la multiplicación	
	1.11 Multiplicación con números hasta 99999	
	1.12 Divisiones exacta	X
	1.13 Divisiones inexacta	X
	1.14 División de dividendo de tres cifras y div	

	dos cifras	
2. Sistema de Funciones	2.1 Unión, intersección y diferencia de conjuntos	X
	2.2 Operadores aditivos, sustractivos, multiplicati	X
3. Sistema de Geométrico y Medida	3.1 Orientación en el espacio Dirección y Sentido	
	3.2 Orientación en el plano Dirección y Sentido	
	3.3 Ubicación en una cuadrícula	X
	3.4 Cambios de dirección y giros Ángulos	X
	3.5 Superficies planas y curvas	
	3.6 Medidas convencionales de superficie. cuadrado y centímetro cuadrado.	
	3.7 Cálculo de áreas	X
4. Sistema de Estadística	4.1 Representación en diagrama de barras	X
	4.2 Estadística	X

Estos temas están basados en el currículo de estudio para enseñar Matemáticas para quinto año de básica aprobados por el Ministerio de Educación del Ecuador y se encuentran en el Anexo I.

### 2.3 Especificación de Requerimientos del Software

Ahora bien, ya se ha explicado por qué la metodología constructivista es la escogida para modelar los ejercicios propuestos en el software. Por otro lado ya se ha visto todos los temas que serán abarcados por el software. Es tiempo entonces de ver qué tipo de metodología se va a emplear para implementar el programa. Sin embargo, antes de entrar en este tema se cree necesario mencionar la importancia de la de la Especificación de Requerimientos del Software.

### **2.3.1 Importancia de la Especificación de Requerimientos del Software**

- El análisis y desarrollo de requerimientos tiene como producto final: un acuerdo documentado entre el cliente y el grupo de desarrollo acerca del producto a ser construido.
- El documento ERS establece con precisión las funciones y capacidad del software así como sus restricciones.
- El ERS es la base para toda subsecuente planificación, diseño, y codificación, así como para las pruebas del software y documentación del usuario.
- El ERS debe comprender la totalidad de los requerimientos.
- Los desarrolladores y clientes no deben realizar presunción alguna.
- Si cualquier requerimiento funcional o no funcional no es identificado en el ERS, no es parte del acuerdo y por lo tanto nadie debe esperar que aparezca en el producto final.

### **2.3.2 Características de un ERS**

Un ERS debe ser:

- a) Correcto;
- b) Inequívoco;
- c) Completo;
- d) Consistente;
- e) Delinear que tiene importancia y/o estabilidad;
- f) Comprobable;
- g) Modificable;
- h) Identificable.

### **2.3.3 IEEE 830**

Se ha escogido este estándar para ERS porque ha sido establecido por el “Institute of Electrical and Electronics Engineers,” que es la asociación profesional líder para el avance de la tecnología. Este brinda una colección de buenas prácticas para escribir ERS



y describe los contenidos y las cualidades de una buena especificación de requerimientos, además se muestran ejemplos de especificaciones.

Este estándar tiene la siguiente estructura.

### **Tabla de contenido**

1. Introducción.
  - 1.1. Propósito.
  - 1.2. Ámbito.
  - 1.3. Definiciones,  
Acrónimos y abreviaciones.
  - 1.4. Referencias.
  - 1.5. Panorama General.
2. Descripción global.
  - 2.1. Perspectiva del producto.
  - 2.2. Funciones del producto.
  - 2.3. Perfil del usuario.
  - 2.4. Limitaciones.
  - 2.5. Supuestos y dependencias.
3. Requerimientos específicos.
  - 3.1 Requerimientos Funcionales
  - 3.2 Requerimientos de Interfaz externa
  - 3.3 Requerimientos de desempeño
  - 3.4 Restricciones de diseño
  - 3.5 Atributos
  - 3.6 Otros
4. Apéndices.
5. Índice.

Los requerimientos del software multimedia de esta tesis serán establecidos tan completamente como sea posible y están especificados en el capítulo 3.

## 2.4 Fundamentos Multimedia

### 2.4.1 Concepto de Sistemas Hipermediales

El término HIPERMEDIA es una combinación de los conceptos HIPERtexto y multimedia y hace referencia a una tecnología de construcción de (hiper)documentos que permite a los lectores encontrar fácilmente la información que realmente necesitan, de la manera que ellos decidan, a través de enlaces establecidos por el autor entre los diferentes elementos de información multimedia (texto, sonido, imagen, vídeo, etc.) que conforman el documento.

Un hiperdocumento se compone, en principio, de dos tipos de elementos: por un lado estarían los *nodos*, o "contenedores" de la información multimedia; y por otro los *enlaces*, que interconectan los nodos permitiendo otras alternativas de navegación por la información diferentes del clásico acceso secuencial "desde la primera a la última línea". Existe, no obstante, un tercer componente fundamental denominado *ancla*. Se trata de un mecanismo para señalar puntos incluidos en el interior de los nodos que sirven de origen o de destino a un determinado enlace entre nodos. Por ejemplo, en un nodo con contenido textual puede haber palabras o frases marcadas que al ser seleccionadas por el usuario (p. ej. con el ratón) activen un enlace originando el acceso al contenido de otro nodo (o a otra parte del mismo nodo). Este nodo destino podría contener información multimedia de tipo secuencia de vídeo, con lo que el ancla debería marcar el fotograma o el instante de la secuencia de vídeo que debería comenzar a visualizarse al activarse el enlace.

Los Sistemas Hipermediales son los entornos que ofrecen a los usuarios todos los mecanismos para la creación, manipulación y consulta de hiperdocumentos. Así, respecto a su interface con los usuarios, proporcionan *browsers* o visualizadores para la navegación y, opcionalmente, visiones de mapas generales del hiperespacio del documento para que los usuarios conozcan su situación en cada momento. También incluyen herramientas de autor para el mantenimiento de los contenidos de los nodos y de la estructura de enlaces si es necesario. En este caso se deberán tener en cuenta todas las posibilidades multimedia, por lo que harán falta editores, no solo de texto, sino también de sonido, de imágenes, etc. y mecanismos de conexión entre todos estos tipos de información. Además de las facilidades anteriores que son percibidas por los usuarios del

sistema y, teniendo en cuenta que la información de un documento hipermedia (tanto la de los nodos como la relativa a los enlaces) será registrada físicamente en un sistema de ficheros o en una base de datos, deberán existir también mecanismos que adapten dicha información a las características del sistema de almacenamiento utilizado, así como los correspondientes procedimientos de acceso y recuperación de datos.

#### **2.4.2 Importancia de una Metodología para el Desarrollo de Sistemas Hipermediales.**

Debido a la complejidad evidente de los documentos hipermedia, desde los orígenes de esta tecnología se ha intentado establecer un modelo universal de hiperdocumento que permita su percepción desde diferentes niveles de abstracción para facilitar el desarrollo de estándares de interface entre niveles que garanticen la portabilidad de los documentos generados

Habitualmente el desarrollo de Sistemas Hipermediales suele hacerse utilizando directamente herramientas de autor a nivel de implementación, descuidándose el importante proceso previo de análisis y diseño abstracto de los aspectos estructurales, de navegación y de interface con el usuario que, de esta forma, no quedan recogidos en documentación alguna, haciendo complicado el mantenimiento futuro del sistema. Sin embargo, en los últimos años existe una tendencia a considerar el desarrollo hipermedial con un enfoque de proceso de ingeniería (del software), por lo que ya se han propuesto diferentes metodologías, como HDM o RMM, que establecen la necesidad de considerar un diseño previo a la construcción del sistema y ofrecen una serie de técnicas, más o menos formales, para recoger en diferentes modelos abstractos las especificaciones del sistema hipermedial a desarrollar.

Sólo recientemente han empezado a aparecer metodologías que asumen la orientación a objetos como paradigma de diseño al igual que ha ocurrido en el ámbito de la Ingeniería del Software. En el caso hipermedia, el enfoque OO (Orientado a Objetos) en los modelos es muy útil debido al gran nivel de abstracción que ofrece y a sus mecanismos de composición (generalización, clasificación y agregación) que facilitan el modelado de la estructura hipermedial.

### 2.4.3 Metodologías Existentes para el Desarrollos de Sistemas Hipermediales ?

Las dos metodologías mas conocidas de desarrollo orientado a objetos de sistemas hipermediales son EORM (Enhanced Object-Relationship Model) y OOHDM (*Object Oriented Hypermedia Design Method*). En ambos casos el proceso de desarrollo comienza con una fase de análisis de la información del sistema. La principal diferencia entre EORM y OOHDM radica en que la segunda separa claramente el diseño conceptual del sistema del diseño de las posibilidades de navegación por el hiperdocumento, ofreciendo técnicas diferentes para ambos casos. Es precisamente por esto que se ha escogido usar la metodología OOHDM en lugar de la EORM.

### 2.4.4 Metodología Object Oriented Hypermedia Design Methodology (OOHDM)

El modelo OOHDM u Object Oriented Hypermedia Design Methodology, para diseño de aplicaciones hipermedia y para la Web, fue diseñado por D. Schwabe, G. Rossi, and S. D. J. Barbosa y se está convirtiendo en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web.

OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de 5 etapas:

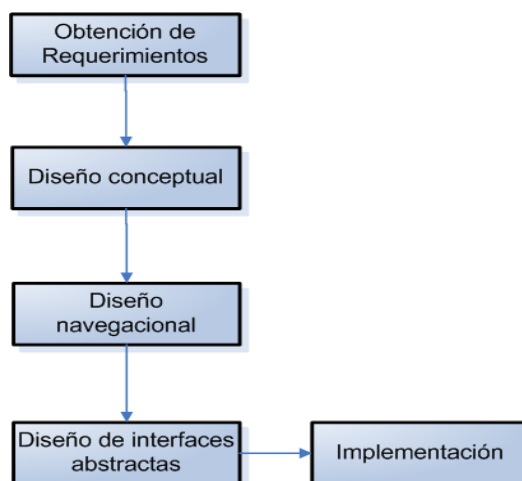


Figura 2.1 Etapas OOHDM

### **2.4.5 Obtención de Requerimientos**

Como en todo proyecto informático la obtención de requerimientos es una de las etapas más importantes. De acuerdo a Soto, Muñoz y de la Horra (2002), la mayoría de estudios realizados acerca de desarrollo de sistemas informáticos demuestran que los errores más caros de ser corregidos son los que se cometen en esta etapa.

Para enfrentar esta dificultad, OOHDM propone dividir esta etapa en cinco subetapas: Identificación de roles y tareas, especificación de escenarios, especificación de casos de uso, especificación de UIs y Validación de casos de uso y UIs. Esta etapa es de valioso apoyo para la Especificación de Requerimientos de Software según el estándar IEEE 830 que será presentada en el siguiente capítulo

#### **2.4.5.1 Identificación de roles y tareas**

En esta subetapa el analista deberá introducirse cuidadosamente en el dominio del sistema, ahora su principal labor será identificar los diferentes roles que podrían cumplir cada uno de los potenciales usuarios de la aplicación.

Los usuarios juegan roles importantes en cada intercambio de información con el sistema. En el ejemplo, una examinación inicial podría revelar los siguientes posibles roles: Alumno, Potencial Alumno, Profesor, Agente de Ventas, Secretaria, Coordinador. Para efectos de validación de los casos de uso es muy importante tener identificado el rol de cada usuario, ya que serán ellos los que entregarán su conformidad con respecto al caso de uso en el que participan.

Luego para cada rol el analista deberá identificar las tareas que deberá soportar la aplicación, como por ejemplo para el rol estudiante: Buscar información acerca de un curso, Buscar información acerca de un profesor u obtener el material para un curso.

### **2.4.5.2 Especificación de escenarios**

Los escenarios son descripciones narrativas de cómo la aplicación será utilizada. En esta subetapa, cada usuario deberá especificar textual o verbalmente los escenarios que describen su tarea.

### **2.4.5.3 Especificación de casos de uso**

Un caso de uso es una forma de utilizar la aplicación. Específicamente representa la interacción entre el usuario y el sistema, agrupando las tareas representadas en los escenarios existentes. Es muy importante que el analista identifique cual es la información relevante en cada uno de ellos, para luego generar un caso de uso coherente.

### **2.4.5.4 Especificación de UIDs**

De acuerdo a UML, los diagramas de secuencia, de colaboración y de estado son capaces de representar un caso de uso. Sin embargo, la especificación de casos de usos usando estas técnicas es un amplio trabajo y puede anticiparse inesperadamente a tomar algunas decisiones de diseño. Para evitar esto OOHDm propone la utilización de una herramienta, llamada UID, que permite representar en forma rápida y sencilla los casos de uso generados en la etapa anterior.

Para obtener un UIDs desde un caso de uso, la secuencia de información intercambiada entre el usuario y el sistema debe ser identificada y organizada en las interacciones. Identificar la información de intercambio es crucial ya que es la base para la definición de los UIDs.

La flecha sin fuente es la primera interacción.

Las flechas representan el procesamiento del sistema antes de la siguiente interacción

En el lugar de “n” se encuentran el número de opciones escogidas de la interacción anterior.

Los rectángulos con línea continua son datos que se deben ingresar obligatoriamente

Los rectángulos con línea no continua son datos que son optativos

Las líneas con la bola negra en un lado representan operaciones que no necesitan el intercambio de interacciones.

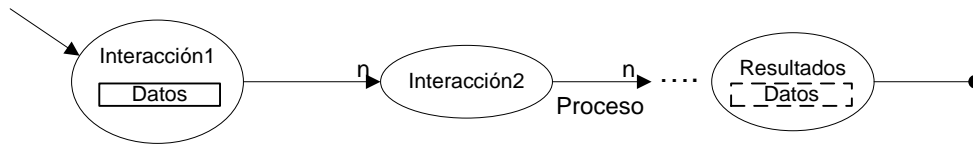


Figura 2.2 Ejemplos de UID

#### 2.4.5.5 Validación de casos de uso y UIDs

En esta etapa, el desarrollador deberá interactuar con cada usuario para validar los casos de uso y UIDs obtenidos, mostrando y explicando cada uno de ellos para ver si el o los usuarios están de acuerdo. El usuario deberá interceder sólo en aquellos casos de uso y UIDs en que participa.

#### 2.4.6 Diseño Conceptual

En la **primera etapa** se construye un esquema conceptual representado por los objetos de dominio o clases y las relaciones entre dichos objetos. Se podría usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones) en el caso en el cual los componentes hipermedia no cambian. Sin embargo, cuando la información es dinámica o cuando se intenta llevar a cabo computaciones complejas o preguntas en los objetos o esquemas, se necesita un modelo orientado a objetos más rico. El modelo OOHDM propone como esquema conceptual basado en **clases**, **relaciones** y **subsistemas**.

La mayor preocupación durante este paso es capturar el **dominio semántico** de una forma neutral, es decir, sin preocuparse mucho con los tipos de usuario y tareas. El resultado de este paso es una clase y una instancia construida de subsistemas, clases y relaciones.

La Gráfica 2-3 muestra una versión simplificada de un esquema conceptual, una revista. En la revista hay historias, las cuales pueden ser ensayos, traducciones o entrevistas; una entrevista es un grupo de preguntas y respuestas. Cada historia tiene un autor, y una entrevista es también relacionada a la persona que permite la entrevista.

Se debe notar que en el ejemplo no hay clase “revista” en el esquema. La revista como tal será tomada en cuenta en la fase del diseño navegacional. Cada tipo posible se lo denomina como perspectiva, ejemplificada en la gráfica 2-3: el atributo “illustration” de un Ensayo puede una foto o un video. Cuando existen múltiples perspectivas, se usa la notación “[p1,p2],” y una “+” puede ser usada delante de una perspectiva para indicar que esta va aparecer por defecto, como es el caso de “Photo” para illustration. Solamente la perspectiva por defecto debe estar presente en todas las instancias, mientras que las otras pueden como también no pueden ser implementadas.

Durante el diseño de la interfaz navegacional y abstracta Se explicará de una mejor manera este especificación definiendo cual va a ser la perspectiva que se mostrará, cuando y como. Claramente, cuando se implementan clases en el modelo conceptual, se necesita especificar una instancia de una variable para cada perspectiva. Decidir si se expresa una relación como atributo, un método o una combinación o ambos es una decisión de diseño que ha sido muy discutido.

En esta tentativa de diseño, se proporciona tres abstracciones para confrontar la complejidad : agregación, generalización/especificación y un concepto de empaquetamiento, subsistemas.

El primero es útil para describir clases complejas como una agregación de clases más simples. La segunda para construir jerarquías de clases usando el mecanismo de herencia. Los subsistemas son un mecanismo de empaquetamiento para abstraer modelos de dominios enteros.

En la gráfica 2-3, una historia puede ser un ensayo o una traducción o una entrevista.



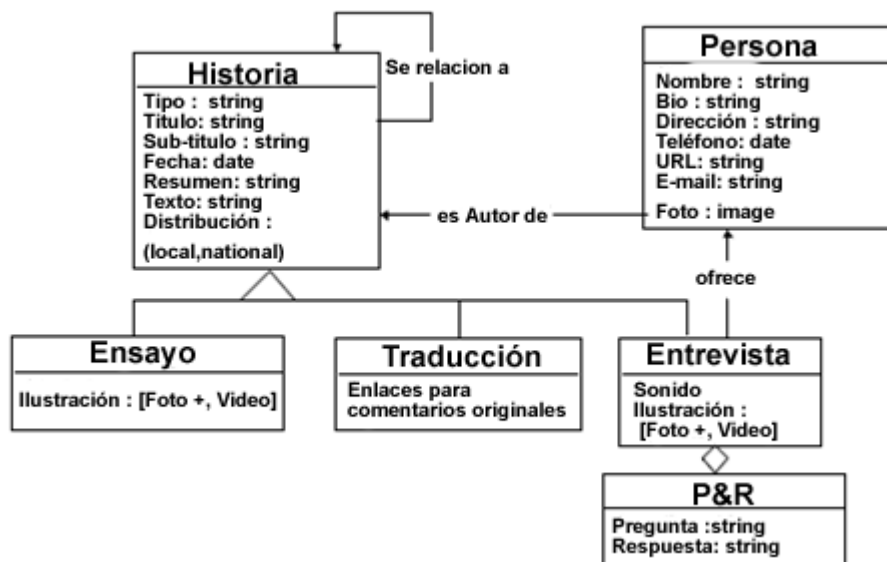


Figura 2.3 Modelo Conceptual para una revista en línea

## 2.4.7 Diseño Navegacional

Generalmente, en la construcción de páginas web, los diseñadores no dan la debida importancia a la navegación, esto provoca en muchas ocasiones problemas en la orientación del usuario. Sin embargo, en OOHDM la navegación es considerada un paso crítico. Cada modelo navegacional proporciona una vista “subjetiva” de un modelo conceptual.

Al diseñar una estructura navegacional de una aplicación, se tomará en consideración varios aspectos tales como:

- Cuales objetos serán navegados, cuales son los atributos que ellos poseen y cuales son las relaciones entre estos objetos y los definidos en el esquema conceptual. Se hará esto al definir nodos y vínculos como vistas orientada a objetos de los objetos conceptuales y sus relaciones.
- Que tipo de estructuras de composición existen entre objetos navegacionales y como estos están relacionados.
- Cual es la estructura fundamental de navegación? En cuales contextos el usuario navegará? Se introducirá el concepto de contextos navegaciones, una arquitectura primitiva para organizar el espacio de navegación.

- Se necesita decidir si los objetos navegados puede verse diferente de acuerdo al contexto en los que son visitados, y se debe especificar claramente esas diferencias. Se debe usar clases en contexto para decorar los objetos navegacionales.
- Cuales conexiones y estructuras de acceso existen entre los objetos que serán navegados (enlaces, caminos, índices, tours guiados, etc)?
- Como la navegación procede cuando el usuario salta de un objeto a otro (el efecto de la navegación).

En OOHDM, hay un grupo de clases navegacionales pre-definidas: nodos, hipervínculos y estructuras de acceso. La semántica de nodos e hipervínculos son las ya conocidos en aplicaciones hipermedia. Las estructuras de acceso, tales como índices y tours guiados representan posibles caminos para acceder a los nodos.

La forma de la semántica navegacional en que OOHDM es tal que cuando un hipervínculo es seguido, el nodo fuente es desactivado y el nodo destino activado, como sucede comúnmente en los exploradores web. No se hará referencia a los casos más complejos.

Para definir los nodos como vistas orientadas a objetos de las clases conceptuales que se definieron durante el diseño conceptual, se usa un lenguaje de consulta (query language). Se define a un nodo al combinar atributos de diferentes clases relacionadas en el esquema conceptual. Los nodos poseen atributos de tipo simple (unico), vínculos de ancla, y pueden ser atómicos o compuestos. Las anclas son instancias de una ancla de clase (o una de sus sub-clases) y son parametrizadas con el tipo de hipervínculo que hospedan. El comportamiento navegacional estándar cuando un nodo recibe el mensaje "ancla seleccionada()" es delegar el mensaje al ancla correspondiente, la que a su vez activa el hipervínculo correspondiente. De esta manera, será posible redefinir el comportamiento estándar del nodo o ancla para que se pueda ejecutar comprobaciones antes que la navegación continúe.

Las estructuras de acceso (tales como índices o tours guiados) son también definidas como clases y caminos alternativos presentes para la navegación en la aplicación. En

términos de arquitectura orientada a objetos, las relaciones entre nodos y objetos conceptuales, y entre hipervínculos y relaciones en el esquema, son expresadas como instancias del diseño padrón del observador. La sintaxis general para definir a los atributos de los nodos se muestra a seguir (el sintaxis para hipervínculos es similar).

```

NODE      Nombre      [FROM NomClase: Nomvar] [INHERITS FROM Clasenodo]
atrib1:   tipo1       [SELECT nombre1] [FROM clase1:Nomvar1, clase] :varName]
          WHERE expresion logica]
atrib2:   tipo2       [SELECT nombre2].....
.....
atribn:   tipon      [Idem]
END
DONDE:
• nombre es el nombre de la clase de nodos que se crea
• NomClase es el nombre de la Clase conceptual (de donde el nodo es mapeado)
• Clasenodo es el nombre de la super clase
• atribi son los nombres de los atributos de esta clase, tipoi el tipo de los atributos
• nombrei son los temas para la expresión de consulta y vari usadas para expresar condiciones lógicas
• expresión lógica permite definir clases cuyas instancias son combinaciones de objetos definidos en el esquema conceptual cuando ciertas condiciones en sus atributos y/o relaciones se asimilan

```

Figura 2.4 Sintaxis para definir a los atributos de los nodos

Como ejemplo, se considera la revista en línea de la figura 2-3. Con este sintaxis simple se puede definir, por ejemplo, un nodo de clase “historia,” incluyendo como uno de sus atributos el “nombre” y la “biografía” de la *Persona* quien lo escribió y un ancla de un hipervínculo para esa “*Persona*.” Notar que, en el modelo conceptual, el *nombre de la Persona* y la biografía son atributos de la clase *Persona*. Al considerar los nodos e hipervínculos como vistas en las clases conceptuales es posible crear objetos navegacionales de una forma oportuna, de acuerdo a los perfiles de los usuarios y tareas, sin modificar los objetos conceptuales.

El ejemplo arriba resultará en la definición del nodo siguiente:

```

NODE Historia [FROM Historia:Hist] [INHERITS FROM Persona]
Autor: String [SELECT Nombre] [FROM Persona :Pr WHERE Pr es Autor de Hist]
Autor_bio :String [SELECT Bio] [FROM Persona:Pr WHERE Pr es Autor de Hist]
..... (otros atributos "preservado" drom la clase conceptual Historia)
paraAutor :Ancla (es Autor de)
END

```

Figura 2.5 Definición de nodo

Notar que el valor del atributo “toAuthor” (*para Autor*) es una ancla que es parametrizada con la clase de hipervínculo *Is Autor of* (es Autor de). Cuando se define la apariencia de la interfaz de la clase nodo “historia,” se puede, por ejemplo, hacer que la ancla aparezca como un botón invisible arriba del nombre del autor (atributo autor). Aunque pueda parecer que ambos atributos tengan el mismo comportamiento, solo el ancla actúa en respuesta a un evento de la interfaz.

La figura 2-6 contiene un esquema navegacional para la revista en línea. Notar que en esta aplicación fue decidido que “*Personas*” no deben aparecer, y por este motivo la información relevante fue incluida en la clase “*Historia*,” como se muestra en el párrafo anterior.

Lo mismo ha sido hecho para la *Persona* que otorgó una entrevista, la cual es una sub clase de “*Historia*.”

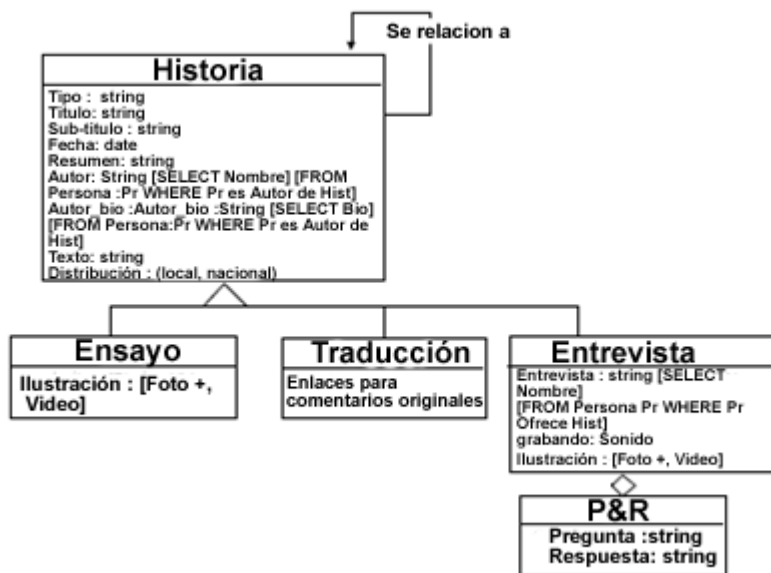


Figura 2.6. Esquema navegacional de la revista en línea.

Notar que la clase persona no aparece; los atributos del autor hacen parte ahora de la historia. Lo mismo para con la persona que otorgó la entrevista.

Aplicaciones hypermedia que son bien diseñadas deben tomar en cuenta la forma en la cual el usuario explora el espacio hypermedia. Información redundante debe ser usada

juiciosamente y se debe ayudar al usuario a navegar controlada y consistentemente. Infelizmente, nodos e hipervínculos no son suficiente para llenar

Por este motivo, después de que las clases navegacionales han sido decididas, es necesario estructurar el espacio de navegación que será presentado al usuario. En OOHDM, esta estructura se define agrupando objetos navegacionales en grupos llamados contextos. Cada definición de un contexto incluye: los elementos que contiene; la especificación de su estructura de navegación interna; un punto de entrada; restricciones de acceso en términos de clases de usuario y operaciones; y estructuras de acceso asociadas.

Hay seis formas para definir contextos:

1. Clase simple derivada: incluye todos los objetos de una clase que satisfacen una propiedad, i.e : “historias de autor=Julian Lopez.” Una variante es el contexto basado en consultas, donde el usuario define la propiedad en el momento de navegación.

**GRAFICAMENTE:**



2. Grupo de clase derivada – es un grupo de contextos de clases simples derivadas, donde la propiedad de definición de cada contexto es parametrizada. i.e “historia por autor ” (el autor puede variar).

**GRAFICAMENTE:**



3. Hipervínculo simple derivado – incluye todos los objetos relacionados con un objeto dado; i.e “historias escritas por Julian Lopez.” Gráficamente es lo mismo que la ítem 1.
4. Grupo de hipervínculo derivado- Un grupo de contextos hipervínculos derivados, siendo obtenidos al variar el elemento fuente del hipervínculo. i.e “historias escritas por periodistas” (los periodistas pueden cambiar). Gráficamente lo mismo que 2.
5. Arbitrario – es un grupo enumerado; i.e, un tour guiado. Gráficamente lo mismo que 1.

6. Dinámico : Es un grupo donde los elementos cambian durante la navegación: i.e historial, canasto de compras. Gráficamente:



La estructura navegacional de la aplicación es definida en un diagrama de contexto, el cual muestra todas las estructuras de acceso y contextos definidos para esta aplicación, y las posibles navegaciones entre ellos. La siguiente figura muestra el diagrama de contexto de la revista en línea.

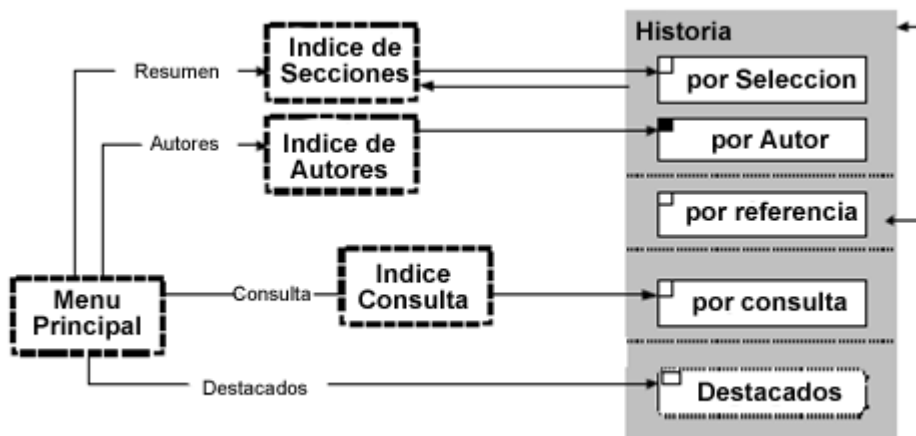


Figura 2.7 Diagrama de Contexto de una revista en línea

El pequeño cuadro de la esquina superior izquierda significa que el grupo posee un índice para sus componentes. La ausencia de una línea punteada entre los grupos indica que es posible cambiarse automáticamente de un contexto en un grupo a un contexto en otro grupo. Este no es el caso para los grupos de contexto “por referencia,” “por consulta,” y para el contexto simple de “Destacados” (highlights), los cuales están separados por líneas punteadas.

Esto significa, por ejemplo, que si el lector está viendo hacia una historia en una sección dada, este podrá navegar para la siguiente historia de la misma sección o a la siguiente historia del mismo autor. Sin embargo, no será permitido navegar a la siguiente historia destacada (highlight), no tendría sentido.

La flecha que va desde “Historia” hacia si misma indica que historias pueden referirse a otras historias relacionadas. Si el lector está mirando hacia una historia, y entonces sigue un hipervínculo hacia una historia relacionada, este llegará al contexto donde todas las historias relacionadas están agrupadas. En este punto, este debe volver para la historia original, como indica la doble flecha, o hacia el índice de resumen, como indica la flecha que va desde “Historia” al “Índice de Secciones.”

Una vez que los contextos hayan sido definidos, es posible extender la definición de las clases de navegación al especificar “decoradores” (decorators), que son atributos que solo pueden ser visibles cuando un objeto es accedido dentro de un contexto dado, tales atributos son definidos en clases denominadas “EnContexto” (InContext). Por ejemplo, cuando se va a “Historias” de Julian Lopez, la biografía del autor puede no estar incluida como un atributo de la historia, mientras que puede ser incluida cuando se va a “destacados.”

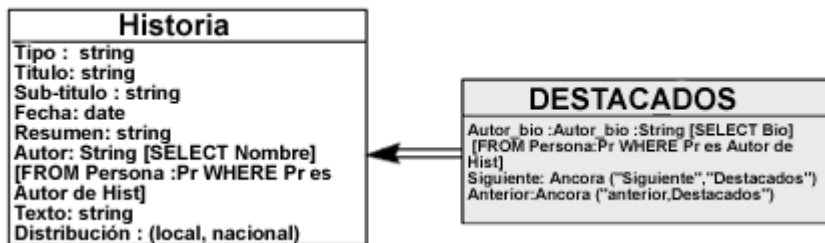


Figura 2.8 Clase “EnContexto” para “Historia” dentro de contextos de “Destacados.”

En este caso, la biografía del autor solo puede ser vista cuando una historia está siendo accedida dentro de un contexto de “Destacados.”

## 2.4.8 Diseño de interfaces abstractas

Después de definir la estructura navegacional, se debe definir el diseño de interfaces abstractas. Esto significa definir la forma en que los diferentes objetos navegacionales van aparecer para el usuario, cuales objetos de interfaz van a activar la navegación entre

otras funcionalidades de la aplicación, y cuales serán las transformaciones en la interfaz y cuando.

OOHDM considera la importancia de la interfaz de usuario y usa un diseño denominado “**Vista de Datos Abstracta**” (Abstract Data View – ADV) para describir el interfaz de usuario de una aplicación hypermedia. ADV son objetos cuando estos tienen un estado y una interfaz, donde la interfaz puede ser ejercitada a través de mensajes (particularmente eventos externos generados por el usuario). ADV son abstractos cuando estos representan solamente la interfaz y el estado, y no la implementación. En general, un ADV especifica la organización y el comportamiento de una interfaz. Sin embargo, la apariencia física actual o los atributos y el estado real del esquema del ADV en la pantalla es hecho en la fase de implementación.

Varias formas de abstracción y composición son usadas en el diseño de ADV. Primero, ADV pueden estar compuestas por agregación o composición de ADV de bajo nivel, permitiendo así la construcción de interfaces de usuario con objetos anidados, como es el caso del ADV de la gráfica 2-10 que está compuesta de tres Adv.

Para este ejemplo y los siguientes vamos a usar diagramas que corresponden al sitio web de Portari, una aplicación hypermedia que contiene parte del trabajo y documentos relacionados con Candido Portinari, un pintor brasilero famoso.

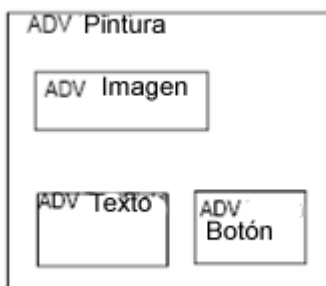


Figura 2.9 ADVs pueden ser agregaciones.

En resumen, los ADVs permiten expresar :

- a) La manera en que los objetos de interfaz están estructurados usando agregación y generalización/especificación como mecanismos de abstracción. Los ADV expresan la estructura del esquema estático que implementa la interfaz. Los ADVs



- permiten definir la apariencia de la interfaz de los objetos navegacionales y otros objetos de interfaz servibles (como barras de menús, botones, etc).
- b) La manera en la cual son estáticamente relacionados con objetos navegacionales. Se usan diagramas de configuración como una herramienta para expresar estas relaciones (ver figura 10).
  - c) Como estos se comportan cuando reaccionan con eventos externos, en particular como estos permiten la navegación y cuales transformaciones en la interfaz ocurren cuando el usuario interactúa con la aplicación. Se usa cuadros-ADV (ADV-Charts), derivados de diagrama de estado, para expresar aspectos de comportamientos y estructura que son típicos en datos multimedia (ver figura 11).

#### **2.4.8.1 Diagrama de Configuración**

A seguir se observa el diagrama de Configuración para el ejemplo del sitio web de Portinari. Este diagrama muestra un diagrama de configuración simple especificando algunas de las interacciones abstractas entre objetos de interfaz y de navegación (llamados “propietarios” del objeto de interfaz). Las líneas punteadas son servicios requeridos mientras que las líneas enteras son servicios proporcionados. Los atributos “tema,” “fecha” y “técnica” actúan como objetos de interfaz estáticos, es decir, no reacciones con eventos externos.

Algunas ADV anidadas tales como “Foto”, “Descripción”, “Interfaz de Contexto”, “Mostrar Descripción” y “Mostrar Referencias” exhiben un comportamiento que es perceptible para el usuario. Por ejemplo, cuando el usuario hace clic en “Mostrar Descripción”, el ADV “Descripción” es desplegado. “Interfaz de Contexto” está compuesto a su vez de ADVs anidados que implementan anclas para contexto de navegación como ya se discutió antes.

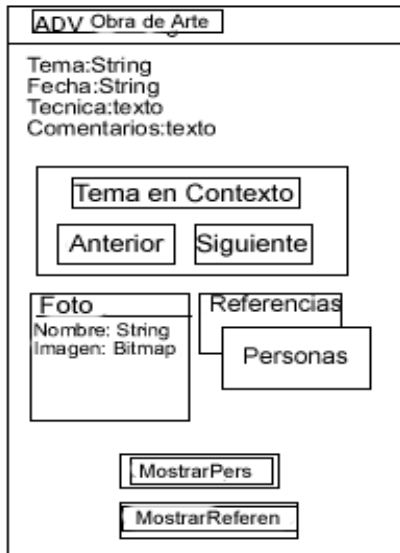


Figura 2.10 Diagrama de Configuración

### 2.4.8.2 ADV

Para terminar con esta sección, se muestra la interfaz real del sitio de Portinari y como los objetos de interfaz abstracta son relacionados con cada uno de sus elementos.

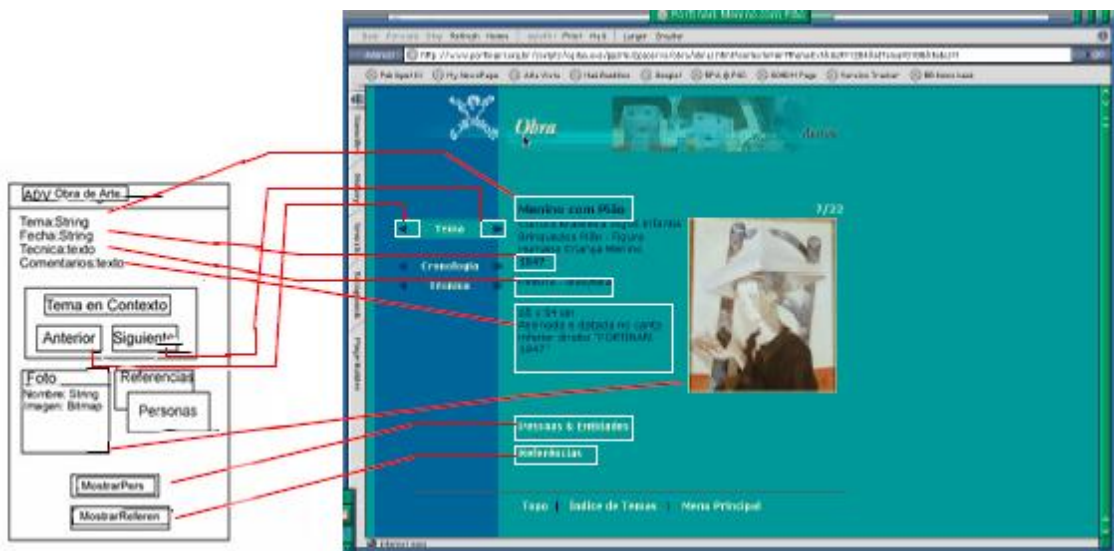


Figura 2.11 ADV y su relación con objetos de interfaz real.

### **2.4.9 Implementación**

En esta fase, el diseñador va a implementar el diseño. Hasta ahora, todos los modelos fueron construidos para ser independientes de la implementación de la plataforma.

Hasta aquí el diseñador ha definido los ítems de la información que hacen parte del dominio del problema. También se ha identificado como estos ítems deben ser organizados de acuerdo al perfil y tareas del usuario. Se ha decidido como la interfaz se va a ver y como se va a comportar. Para la implementación, se debe decidir como los ítems de información (los conceptuales y navegacionales) serán almacenados.

También se debe saber como se va a llevar a cabo (herramientas y extensiones) la apariencia y comportamientos que se quiere.

Para obtener una implementación que sea ejecutable, el diseñador tiene que mapear la interfaz navegacional y abstracta en objetos concretos. El modelo generado después de previamente efectuar actividades puede ser implementado de una manera directa usando varias de las plataformas hypermedia disponibles tales como Flash, Fireworks, photoshop,etc.

## **CAPITULO 3**

### **ANÁLISIS**

#### **3.1 Especificación de Requerimientos de Software según la IEEE 830**

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El documento va dirigido tanto al desarrollador, como a todo aquel personal que está involucrado con el desarrollo del sistema hipermedial así como también a los pedagogos o posibles usuarios finales. Esta especificación servirá de base inicial al desarrollador para el desarrollo del sistema.

##### **3.1.1 Alcance**

El nombre del sistema hipermedia dentro del tema del presente tema de tesis es “Matemáticas para Quinto Grado de Básica.”

Este sistema es requerido a raíz de la creciente necesidad de software educativo en las escuelas, sirviendo así como una herramienta complementaria en la educación de matemáticas para quinto grado de básica.

El sistema ofrece información acerca de diversos temas de la materia de matemáticas de quinto grado y automatiza las evaluaciones de estos temas para reforzar el conocimiento adquirido del alumno.

Por otro lado el Software tiene el objetivo de ayudar en la enseñanza de matemáticas usando una metodología constructivista en los ejercicios propuestos en las evaluaciones. Al mismo tiempo, el sistema intenta ofrecer al estudiante una interfaz que sea familiar a la cultura Ecuatoriana, algo que no se puede mostrar plenamente con un libro.

Finalmente, el software visa ser divertido, agradable e intuitivo intentado motivar al estudiante en el proceso de aprendizaje.

### **3.1.2 Definiciones, Acrónimos y Abreviaciones**

*Hipertexto:* Documento digital o no, que se puede leer de manera no secuencial. En donde tiene estos elementos como secciones, enlaces o hipervínculos y anclajes.

*Hipermedia:* se le designa al conjunto de métodos, o procedimiento para escribir, diseñar, o componer contenidos que tengan texto, video, audio, mapas u otros medios, y que además tenga la posibilidad de interactuar con los usuarios.

*Multimedia:* La multimedia es el uso de diversos medios (e.g. texto, audio, gráficos, animación, vídeo, e interactividad) de transporte de la información. La multimedia también se refiere al uso de la informática de crear, almacenar y contenido de la experiencia multimedia. Mientras que la información se presenta en varios formatos, la multimedia realza la experiencia del usuario y la hace más fácil y más rápida para tomar la información.

*Nodo:* Espacio real o subestracto en el que confluyen parte de las conexiones de otros espacios reales o abstractos que comparten sus mismas características y que a su vez también son nodos (secciones).

*Constructivismo:* Es una corriente de pensamiento surgida en 1980 que afirma que nunca se podrá llegar a conocer la realidad como lo que es ya que, al enfrentarse al objeto de conocimiento, no se hace sino ordenar los datos que el objeto ofrece en el marco teórico del que se dispone. Para el pensamiento constructivista, la realidad es una construcción hasta cierto punto "inventada" por quién la observa.

### **3.1.3 Acrónimos**

OOHDM : Object Oriented Hypermedia Design Method (ver capítulo 2)

SO : Sistema Operativo

ERS : Especificación de Requerimientos del Software

### **3.1.4 Abreviaciones**

No se han definido

#### **3.1.4.1 Referencias**

- 830-1998 “Recommended Practice for Software Requirements Specifications (ANSI/IEEE)”
- Reforma Curricular para la Educación Básica
- Wikipedia , la enciclopedia libre, <http://es.wikipedia.org>

#### **3.1.4.2 Panorama General**

- Este documento consta de tres secciones. Cada sección tiene sub-secciones enumeradas ordenadamente.
- Esta sección es la Introducción y proporciona una visión general de la ERS.
- En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.
- En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema
- El Apéndice y el Índice mencionados en el capítulo 2 en la sección 2.3.3 no serán abordados por ser irrelevantes para el propósito de este trabajo.

### **3.1.5 Descripción Global**

#### **3.1.5.1 Perspectiva del Producto**

El sistema hipermedial forma parte de un proyecto capital denominado “ESPE Comunitaria,” desarrollado por la Escuela politécnica del Ejército, que pretende apoyar a los niños de bajos recursos o con necesidades especiales en el ámbito educativo.

Para la elaboración de este sistema hipermedia, se ha contado con el apoyo de la “Escuela Fiscal Acosta Soberón,” aunque se debe recalcar que el sistema no está direccionado únicamente para esta escuela en particular. Por el contrario, se espera que el sistema pueda ser adaptable para cualquier escuela primaria de la nación.

#### **3.1.5.2 Interfaces del Sistema**

- a) El sistema debe interactuar correctamente con el sistema operativo Windows sobre el que se desarrolló e implanta
- b) Se necesita Acrobat Reader para leer el manual y tener una resolución recomendada de 1024 x 768
- c)

#### **3.1.5.3 Limitaciones de memoria**

Para un mejor funcionamiento del sistema hipermedial, se recomienda tener una memoria RAM de 128 MB para arriba.

#### **3.1.5.4 Interfaces con el usuario**

Hay dos tipos de usuarios : el alumno y el responsable (profesor, administrador, etc). El modo de operación después de instalado el programa es automático, interactivo (multimedia) e intuitivo siempre y cuando se encuentre dentro del sistema hipermedial. Sin

embargo, hay que considerar que cuando el profesor o responsable del sistema en la escuela quiera cambiar los contenidos del ministest y/o ejercicios, este deberá guiarse en las instrucciones del manual de usuario ya que estos cambios no son posibles desde el sistema hipermedial.

### **3.1.5.5 Operaciones**

No se utilizan, cualquier tratamiento ulterior de los datos generados corre por cuenta del usuario.

### **3.1.5.6 Requerimientos para adaptarse a la ubicación**

El sistema debe ser instalado en cada una de las máquinas donde se intenta visualizar la aplicación.

### **3.1.5.7 Funciones del Producto**

- Las interfaces de usuario que puedan requerirse para esta aplicación deberán ser de manejo intuitivo, fácil de aprender y sencillo de manejar.
- El sistema deberá presentar un alto grado de usabilidad.
- Interfaz de navegación con temas y sub-temas de matemáticas: Los temas de matemáticas estarán organizados de acuerdo al cronograma de actividades para el quinto año de la “Escuela Acosta Soberón,” tomando en cuenta los temas más importantes y suponiendo que este cronograma es usado en varias otras escuelas.
- Evaluaciones para cada tema de matemáticas expuesto: Para cada tema macro habrá una evaluación y calificación automática.
- Control de evaluaciones multimedia: el docente podrá administrar las preguntas multimedia de las evaluaciones desde archivos xml.
- Ayuda desde cualquier parte del programa



### **3.1.5.8 Perfil del usuario**

- El sistema multimedia está dirigido para alumnos del quinto grado de básica y personas mayores responsables.
- Los alumnos deben tener un nivel de conocimiento básico de computación y tener un nivel de educación mínimo de quinto grado de básica.
- Los docentes o responsables deben tener dominio de los temas a tratarse y saber como modificar el minitest y los ejercicios.

### **3.1.5.9 Limitaciones**

- Para su ejecución se requiere la instalación individual del sistema en cada pc destinada a ese uso.
- El contenido de cada tema multimedia teórico del sistema no se lo puede modificar.
- La estructura del sistema no se la puede modificar
- La estructura del minitest no se lo puede modificar

### **3.1.5.10 Supuestos y Dependencias**

- Se espera que la institución que desee adoptar este sistema como herramienta de ayuda en el enseño de matemáticas de quinto grado debe poseer computadores capaces de soportar sin problemas el sistema multimedia.
- Se asume que los requisitos descritos en este documento son estables una vez que sea aprobados atendiendo sugerencias de alumnos y profesores de la Escuela Acosta Soberón. Cualquier petición de cambios en la especificación debe ser aprobada por desarrollador del sistema y gestionada por el mismo
- Este trabajo depende de las exigencias del programa de la “ESPE COMUNITARIA.”
- Se asume que el software va a ser guiado por el profesor

### **3.1.6 Requerimientos Específicos**

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son ESENCIALES, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de demostrabilidad: dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

#### **3.1.6.1 Requisitos Funcionales**

- Todo usuario del sistema tendrá los mismos temas y subtemas de matemáticas para consultar
- Todo usuario podrá escoger el minitest para probar su conocimiento.
- Todo usuario podrá obtener una calificación a medida que se realiza la evaluación
- Solo los responsables del sistema podrán cambiar las preguntas del ministes y contenido de ejercicios.

#### **3.1.6.2 Interfaces de usuario**

- El usuario podrá ejecutar el programa en cualquier browser (se recomienda una resolución de pantalla de 1024 x 768 o superior).
- El usuario responsable dispondrá de un banco de preguntas multimedia para administrar las evaluaciones
- El usuario podrá administrar las preguntas de las evaluaciones propuestas en el sistema

#### **3.1.6.3 Interfaz de hardware**

Los computadores requerirán para el optimo desempeño del sistema las siguientes características mínimas :

- Procesador de 800 Mhz o equivalente con Windows XP;
- 128 MB de RAM;
- 500 MB de espacio en disco disponibles en caso de correr el sistema localmente
- Una conexión LAN fase ethernet 10/100 Mbps en caso de ejecutar el programa remotamente;
- Periféricos de audio y se recomienda una tarjeta gráfica de 64 MB o superior, teclado y mouse.

#### **3.1.6.4 Interfaz de software**

El sistema no tiene comunicación con otros módulos, solo con archivos xml.

#### **3.1.6.5 Interfaz de Comunicación**

Las computadoras no se comunicarán entre si en caso del sistema ser instalado localmente en cada una de las máquinas disponibles.

#### **3.1.6.6 Requisitos de Rendimiento**

- Como la ejecución es local, el número de terminales es ilimitado
- Cada Terminal debe cumplir con los requerimientos de hardware y software citados anteriormente para que el sistema pueda funcionar óptimamente
- Se prevé que el 95% de las transacciones dentro de este esquema deberían ser procesadas en menos de un segundo

#### **3.1.6.7 Requisitos de Desarrollo**

- La ejecución se realiza a través de un archivo swf que será convertido en exe al final. A pesar de no usar Internet o un navegador para visualizar este sistema, se debe considerar por precaución el ancho de banda repercute en la descarga de los

archivos (gráficos, sonidos, etc), por lo que se debe intentar que estos tengan el menor peso posible, considerando mecanismos de comprensión.

### **3.1.6.8 Requisitos de Base de Datos**

Solamente el responsable (profesor, director, etc ) puede acceder a modificar el contenido de la preguntas del minitest y de los ejercicios de las unidades de matemáticas. Estos son algunos de los requisitos:

- Las preguntas y contenido de ejercicios son almacenados en archivos xml por simplicidad y para facilitar el manejo de contenido si se pretende ampliar este software en un futuro.
- Para agregar nuevas preguntas, se debe primero crearla (siendo esta multimedia) y agregarla al banco de preguntas haciendo uso del archivo xml. Se debe contar con imágenes.
- Las preguntas, tipo de preguntas y numero de preguntas se pueden modificar de acuerdo al banco de preguntas y temas disponibles.

### **3.1.6.9 Atributos**

- *Interactividad* : El sistema debe permitir la interacción entre usuario-maquina a través de la interfaz de navegación que debe facilitar y hacer posible la comunicación entre el usuario y el objeto hipertextual. La Interfaz debe determinar el diseño original del hipertexto que debe ser atractivo y sugerente, además de tener un estilo homogéneo.
- *Intuitivo*: El sistema debe disponer de un entorno intuitivo que pone el control del sistema en las manos del usuario haciendo las operaciones del sistema muy sencillas, reduciendo así el tiempo de formación e incrementando la productividad de los alumnos.

- *Mantenibilidad*: El sistema es susceptible de ser ampliado. Por tanto deberá diseñarse fácilmente mantenible, aplicando para su desarrollo las metodologías que para ello sean precisas
- *Disponibilidad* : La disponibilidad está acorde a la distribución del software en las escuelas
- *Portabilidad* : Debido a que el sistema está integrado en la plataforma web, se lo puede acceder desde cualquier sistema operativo siempre y cuando las computadoras estén conectadas en una LAN y el servidor principal opere bajo Windows.
- *Multimedial*: El funcionamiento externo del sistema multimedia debe responder a las siguientes funciones: cognoscitiva, comunicativa, motivadora, informativa, integrativa, sistematizadora, y de control.

### 3.2 Obtención de Requerimientos

A seguir se seguirá los cuatro etapas de obtención de requerimientos especificadas en el capítulo anterior. Los gráficos y método de obtención de requerimientos se basó en el documento “*Propuesta de un modelo navegacional para el desarrollo basadas en OOHDM*” (Soto, Muñoz & De La Horra, 2002)

#### 3.2.1 Identificación de Roles y Tarea

Tabla 3.1 Identificación de Roles y Tareas

Actores	TAREAS
Alumno	a. Buscar información teórica acerca de un tema específico de matemáticas y realizar ejercicios b. Realizar la evaluación multimedia propuesta c. Control de la evaluación multimedia

Responsable (Profesor, Administrador)	<input type="checkbox"/> <ul style="list-style-type: none"> <li>a. Explicaciones adicionales</li> <li>b. Manejo de preguntas propuestas en las evaluaciones y de ejercicios.</li> <li>c. Buscar información teórica acerca de un tema específico de matemáticas</li> </ul>
---------------------------------------	--

### 3.2.2 Especificación de Escenarios

#### A. Buscar información teórica acerca de un tema específico de matemáticas y realizar ejercicios.

**Actores:** Alumno, Profesor, Director y Administrador

##### a.1 Buscar Información

El alumno podrá escoger desde el menú principal, el tema de matemáticas de quinto grado que desea consultar. El usuario tendrá información teórica y ejercicios. Por ejemplo, dentro del tema “Sistema Numérico,” el alumno podrá escoger un tema específico a consultar (i.e Sistema de numeración romana), donde podrá aprender acerca de la teoría de ese tema (con la ayuda del profesor) y realizará ejercicios.

##### a.2 Cambiar de tema o volver al Menú Principal

Dentro del tutorial de cada tema específico, el usuario podrá cambiar de tema, volver al menú principal, ir a los ejercicios, etc. Es decir, tendrá libertad de desplazamiento. Por ejemplo, si el usuario se encuentra en “Sistema Numérico,” el alumno podrá ir fácilmente al menú principal o al tema “Divisiones exactas.”

#### B. Realizar la evaluación multimedia propuesta

**Actores:** Alumno

##### b.1 Responder las preguntas planteadas

El usuario entra a ser evaluado acerca de diversos temas. Por ejemplo, ejercicios acerca de “Sistema Numérico” o “Sistema de Funciones.” Estos ejercicios abarcarán los temas

específicos definidos en cada tema macro. El usuario podrá desplazarse a otro lugar de la aplicación si desea, pero se recomienda que la profesora prohíba esto ya que si se desplaza a otro lugar la calificación es perdida. En el ministest, es importante que se responda correctamente la pregunta para seguir a la próxima pregunta.

## **b.2 Parar evaluación**

Se podrá interrumpir la evaluación si es necesario aunque el alumno no haya terminado de responder las preguntas de la evaluación

## **b.3 Pedir resultado de evaluación**

El alumno será evaluado a medida que responde las preguntas. Para esto el número de intentos, fallos y aciertos van a ser contados a medida que el alumno vaya respondiendo las preguntas.

## **C. Dar permisos de acceso**

**Actores:** Profesor, Director y Administrador

### **c.1 Ingresar al Sistema**

El administrador o profesor encargado ingresa al sistema usando una contraseña específica. Si no se tiene esta contraseña no se podrá ingresar al software multimedia.

### **c.2 Cambiar contraseña de seguridad**

El administrador o profesor encargado podrán cambiar o agregar una contraseña de seguridad para acceder al sistema. De esta forma, hay más seguridad en el acceso al sistema. Esta contraseña se encuentra almacenada en un archivo xml.

## **e. Editar preguntas propuestas en las evaluaciones**

**Actores:** Profesor, Director y Administrador

### **e.1 Agregar, modificar o borrar preguntas propuestas en las evaluaciones**

El administrador o profesor accede al archivo xml datos para hacer los cambios necesarios en las preguntas de las evaluaciones propuestas de acuerdo a un tema macro.

### **e.2 Visualizar modificaciones de preguntas.**

El administrador podrá visualizar las modificaciones que ha hecho en las preguntas de evaluación en el mini test.

## **3.2.3 Especificación de casos de uso**

**Caso de Uso:** Buscar un tema y realizar ejercicios

**Roles:** Alumno, Responsable

**Escenarios:** a1, a2

Descripción:

1. El usuario entra al menú principal.
2. El usuario busca un tema macro
3. El usuario busca un tema específico dentro de tema macro
4. El usuario ingresa a ver la teoría del tema específico
5. El usuario realiza ejercicios
6. El usuario cambia de tema o vuelve al índice principal si desea

**Caso de Uso:** Realizar evaluación

**Roles:** Alumno

**Escenarios:** b1, b2, b3

Descripción:

1. El usuario entra a la evaluación
2. El usuario responde a las preguntas de selección múltiple
3. El usuario regresa a menú principal (cancela evaluación) si desea
4. El usuario visualiza calificación si desea
5. El usuario vuelve al menú principal si desea



**Caso de Uso:** Dar permisos de acceso

**Roles:** Responsable

**Escenarios:** c1,c2

Descripción:

1. El usuario ingresa la contraseña de seguridad para ingresar al sistema
2. El usuario ingresa a archivo xml si desea cambiar la contraseña

**Caso de Uso:** Cambiar preguntas de evaluaciones, ayudas y/o ejercicios

**Roles:** Responsable

**Escenarios:** d1,d2

Descripción:

1. El usuario ingresa al xml correspondiente
2. El usuario hace las modificaciones

**Caso de Uso:** Buscar ayuda

**Roles:** Alumno, Responsable

**Escenarios:** e1, e2, e3

Descripción:

1. El usuario consulta ayuda.
2. El usuario ubica ventana de ayuda de acuerdo a su criterio
3. El usuario cierra ayuda

### 3.2.4 Especificación de UIDs

**UID1:** Buscar un tema

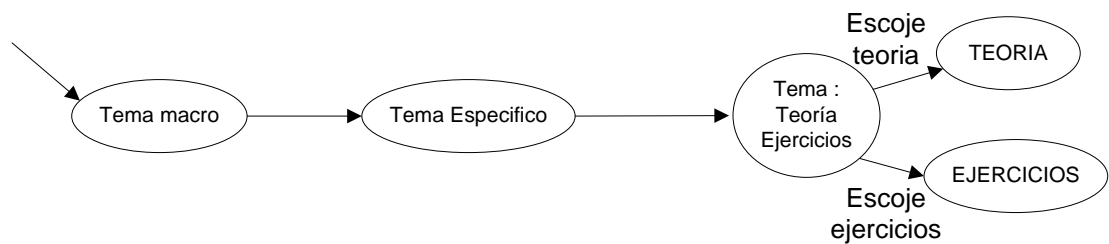


Figura 3.1 UID1

## UID2: Realizar Evaluación

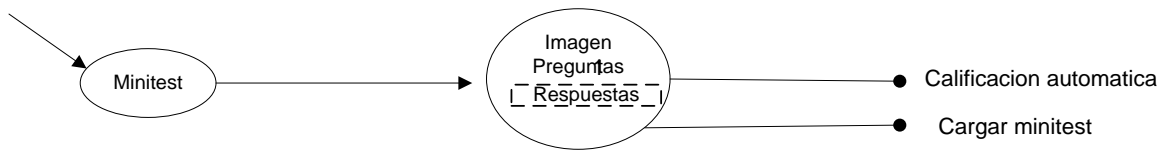


Figura 3.2 UID2

## UID3: Dar permisos de Acceso

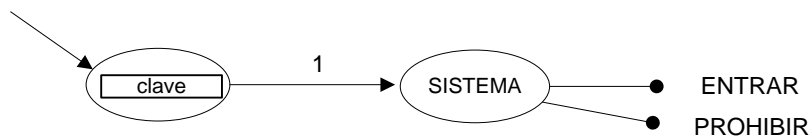


Figura 3.3 UID3

## UID4: Cambiar preguntas de evaluaciones, ayudas y/o ejercicios

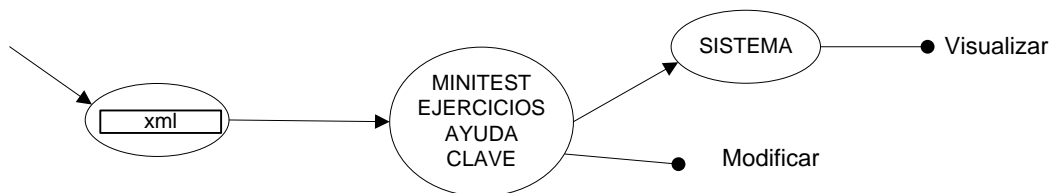


Figura 3.4 UID4

## UID5: Buscar ayuda

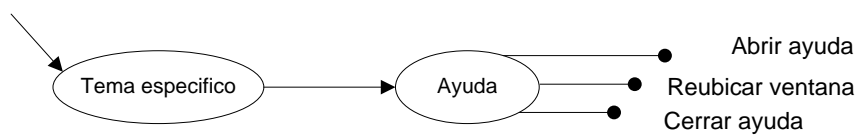


Figura 3.5 UID5

### 3.2.5 Validación de Casos de Uso y UID

Los casos de uso han sido validados por los usuarios.  
Para Esto fueron evaluados 5 niños y 2 profesores.

## 3.3 Diagramas

### Diseño Conceptual

Este diseño conceptual está basado en el ejemplo propuesto en el documento *“An Object Oriented Approach to Web-Based Application Design”* (Schwabe & Rossi, 1998)

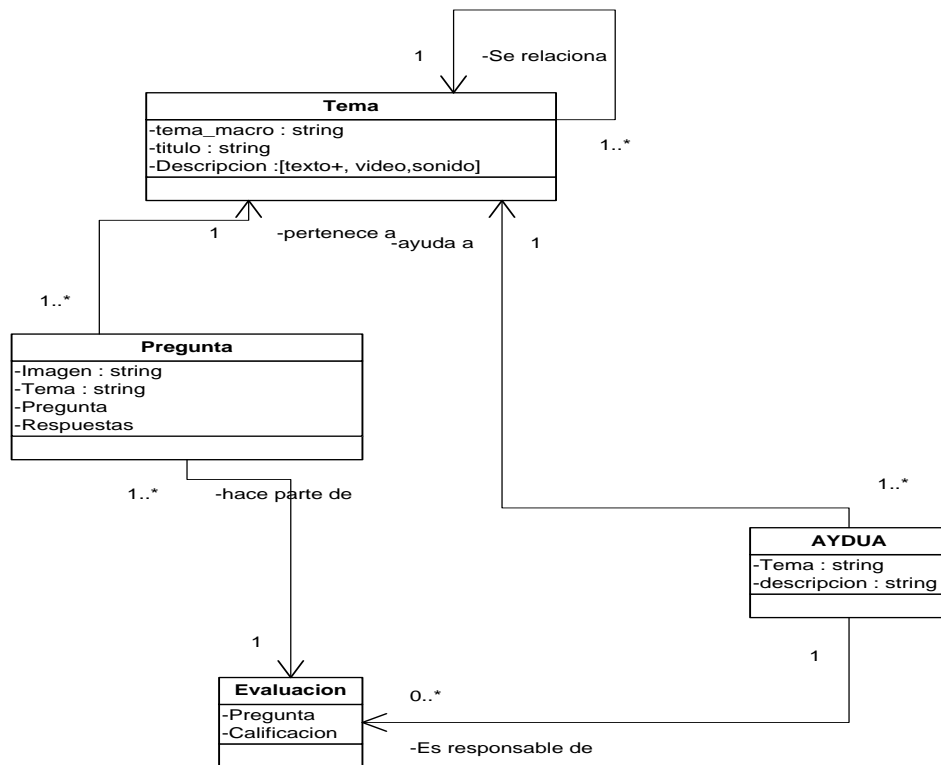


Figura 3.6 Diseño Conceptual

### 3.3.2 Diseño De Navegación

La forma del diseño está basado en el ejemplo propuesto en el documento *“An Object Oriented Approach to Web-Based Application Design”* (Schwabe & Rossi, 1998)

#### 3.3.2.1 Esquema de Navegación de Clases

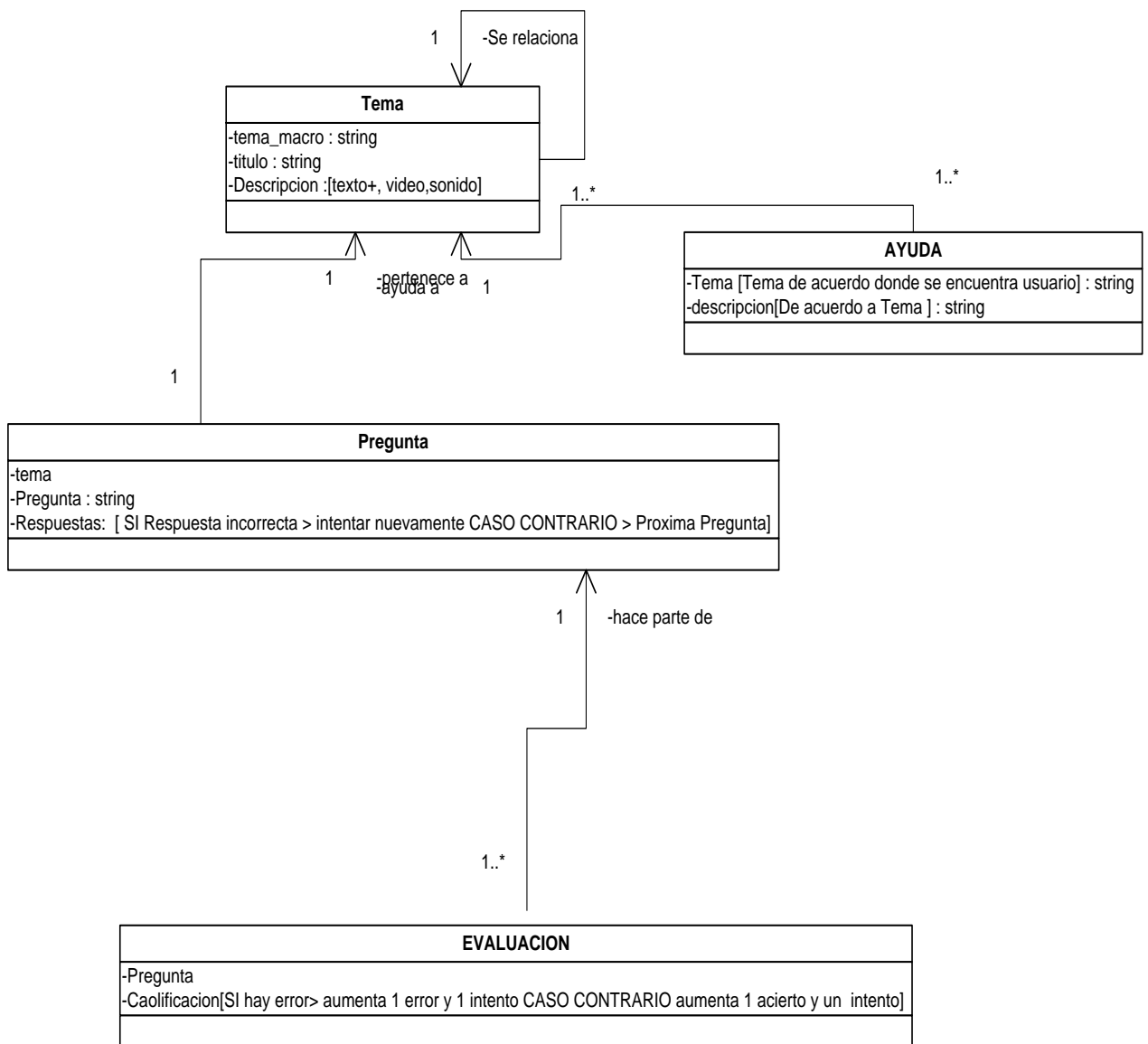


Figura 3.7 Esquema de Navegación de Clases

### 3.3.2.2 Esquema de Navegación de Contextos

La estructura de este esquema está basado en el ejemplo del documento *“Propuesta de un modelo navegacional para el desarrollo basadas en OOHDM”* (Schwabe & Rossi, 1998)

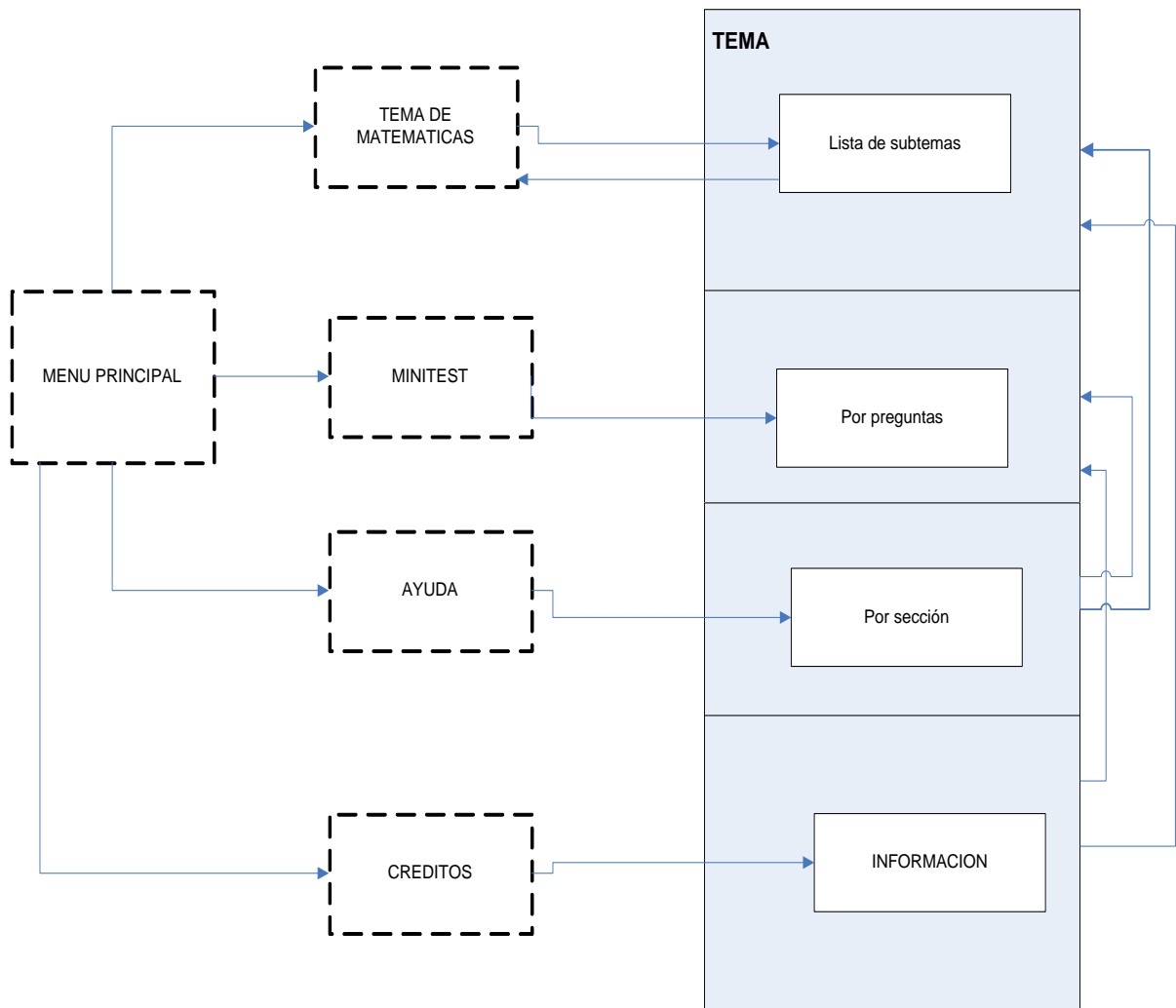


Figura 3.8 Esquema De Navegación De Contextos

### 3.3.3 Diseño De Interfaces Abstractas

Estos diagrama están basados en los ejemplos del documento *“An Object Oriented Approach to Web-Based Application Design”*

### 3.3.3.1 Vista General

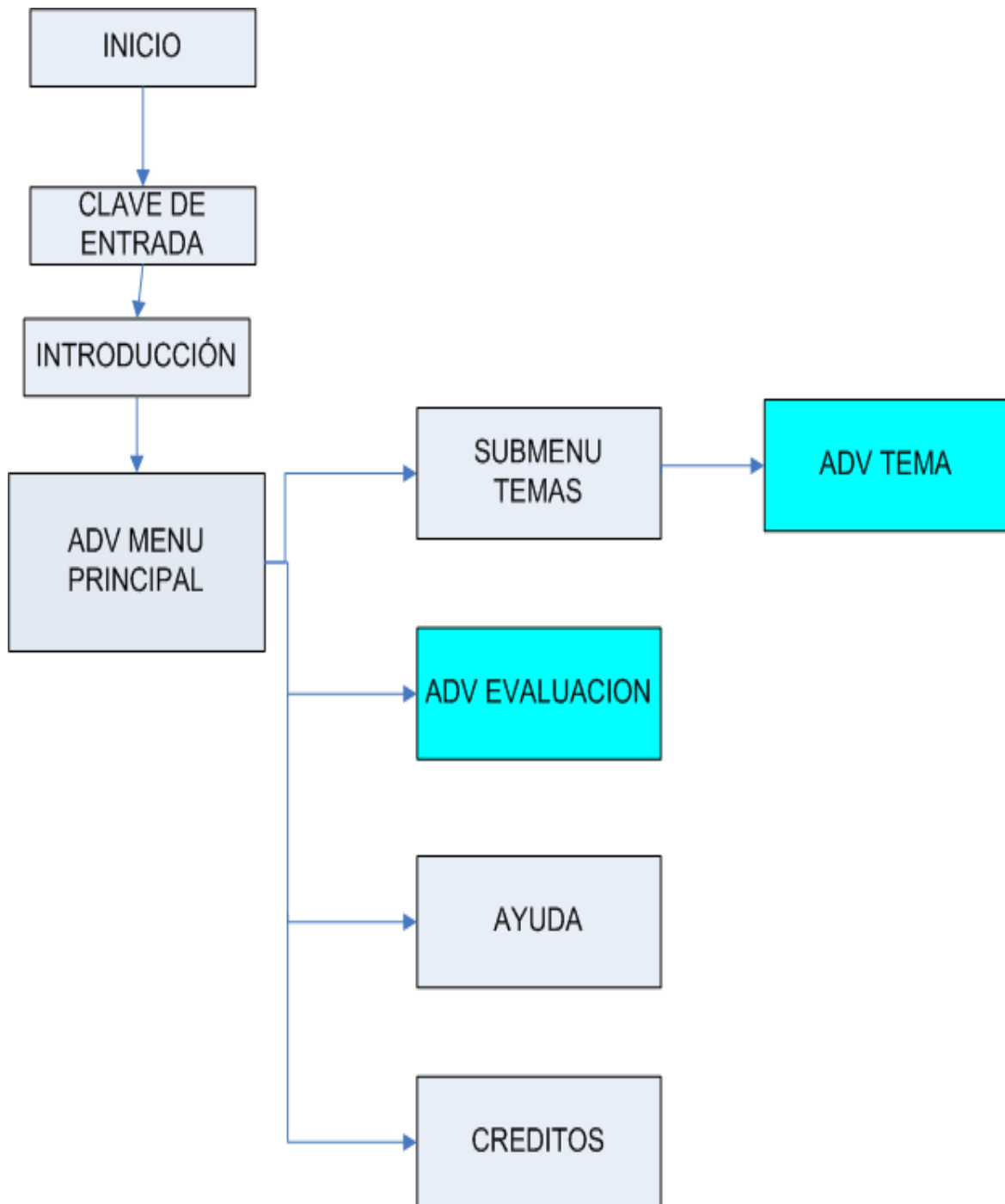


Figura 3.9 Vista General

### 3.3.3.2 Abstract Data View : Menu Principal

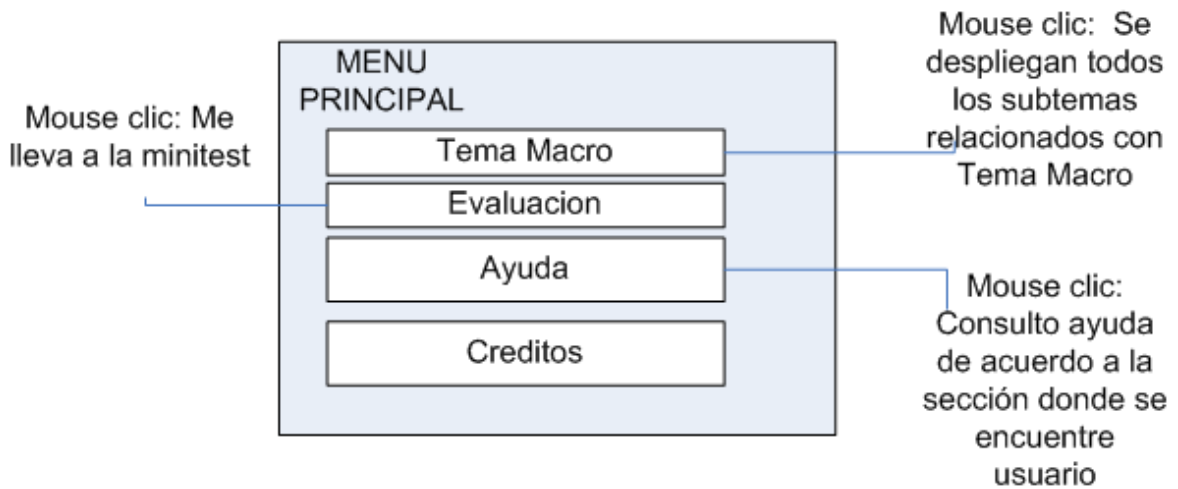


Figura 3.10 Menu Principal

### 3.3.3.3 Abstract Data View : Tema

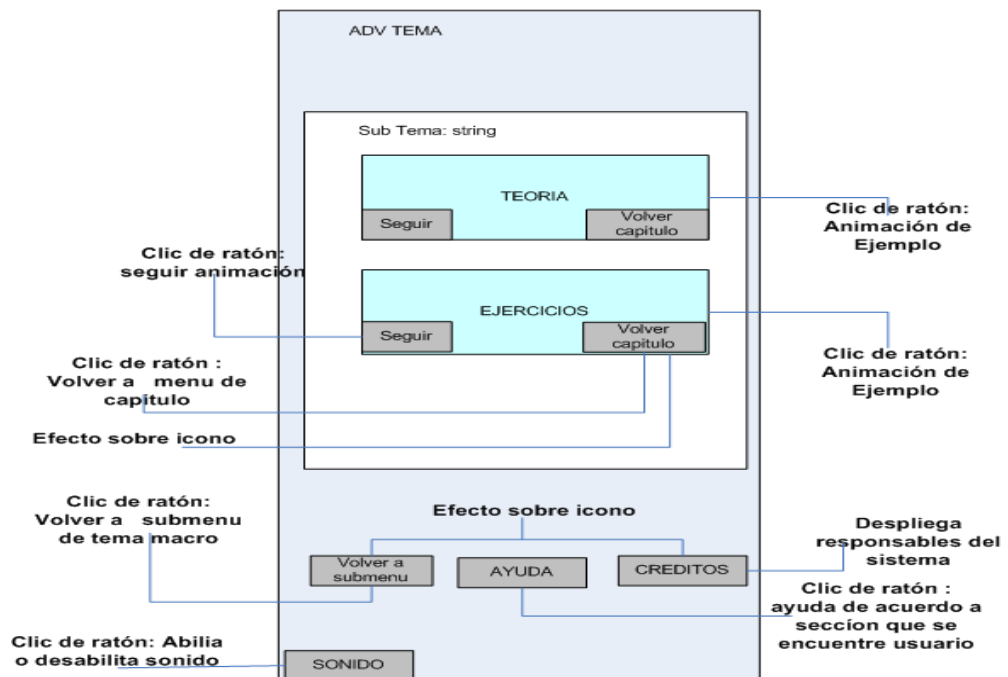


Figura 3.11 ADV Tema

### 3.3.3.4 Abstract Data View : Evaluación

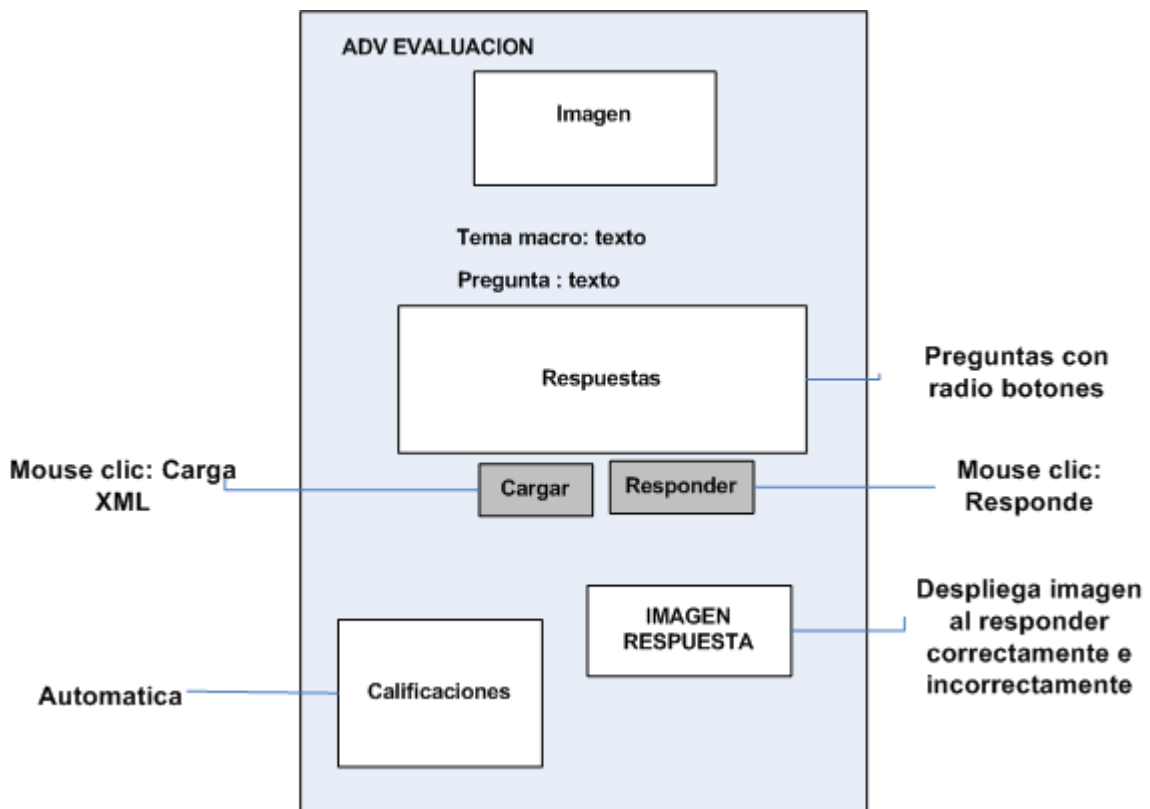


Figura 3.12 ADV Evaluación

### 3.3.3.5 Implementación

Los detalles de la implementación serán relatados en el capítulo 4. Esta sección se limitará a mostrar el resultado de la implementación para que los diagramas presentados puedan ser comparados con el resultado obtenido.





Figura 3.13 ADV Implementación 1



Figura 3.14 ADV Evaluación

## CAPITULO 4

### IMPLANTACIÓN

En esta etapa se mostrará todas las herramientas que se utilizan para la construcción del sistema hipermedial y los equipos que se necesitan para su ejecución.

Por otro lado, en el manual de usuario, se especificará todos los pasos que el usuario debe seguir para hacer uso del sistema sin problemas. Se supone que el profesor debe estar a cargo de este manual y que a través de este, él pueda enseñar a los niños a usar el software cuando sea necesario.

Finalmente se harán pruebas técnicas de la Aplicación para verificar la correcta funcionalidad del sistema.

#### 4.1 PLAN DE IMPLANTACIÓN

##### 4.1.1 Determinación de recursos para la implementación del software

Se detallan los requerimientos del hardware y software a continuación.

Tabla 4.1 – *Especificación de requerimientos de hardware y software*

<b>Requisitos del Software</b>	<b>Requisitos del hardware</b>
Probado en sistema Operativo XP	Procesador 800 Mhz o superior
Adobe Acrobat Reader (para manual)	128 MB de RAM (mínimo)
Lector de xml (Notepade, Wordpad, etc)	500 MB de espacio en disco disponibles
Resolución aconsejable 1024 x 800 píxeles	Periféricos de audio
	Tarjeta de Video de 8 MB (Recomiendo Aceleradora Gráfica (AGP) de 16 superior).

Los programas usados para el desarrollo del sistemas fueron :

- Macromedia Flash 8
- Macromedia Fireworks 8
- Adobe Image 7.0
- Power Audio Recorder versión 1.55
- Setup Factory 7.0
- Microsoft Office Visio 2003
- 

#### 4.1.2 Selección de Recursos

La selección de recursos para la construcción del sistema de aspectos tales como: costo, licencias, portabilidad, interoperabilidad, compatibilidad, requerimientos de hardware.

Para la selección de recursos se consideró:

- a) **Imágenes vectoriales:** La mayoría de las imágenes usadas en el sistema son vectoriales, editadas en Adobe Image e importadas a Flash. La ventaja de las imágenes vectoriales, al contrario de las imágenes basadas en píxeles, es se les puede aplicar zoom de forma ilimitada sin que sus bordes se afecten.
- b) **Archivos de sonido:** A pesar de que el sistema no está basado en la web, se ha tratado de ahorrar espacio en los recursos utilizados. Por eso se usó sonidos en formate MP3, debido a su índice de comprensión. Para esto se ha utilizado el software Power Audio Recorder y otros sonidos han sido bajados de la Internet gratuitamente.

#### 4.1.3 Características del Sistema

Ya se ha discutido varias veces, en los anteriores capítulos, sobre la importancia de una interfaz gráfica de fácil navegabilidad, interactividad e uniformidad para el usuario, entre otros. A seguir se especifica como esto ha sido implementado este sistema y cuales que es lo que este ofrece al usuario.

- a) **Sistema basado en Módulos:** El sistema está basado en módulos (o capítulos) y son independientes uno del otro, es decir, el sistema principal (indec.swf) hace

llamadas individuales a los archivos .swf externos de acuerdo al tema que el usuario escoja. El siguiente diagrama detalla lo explicado:

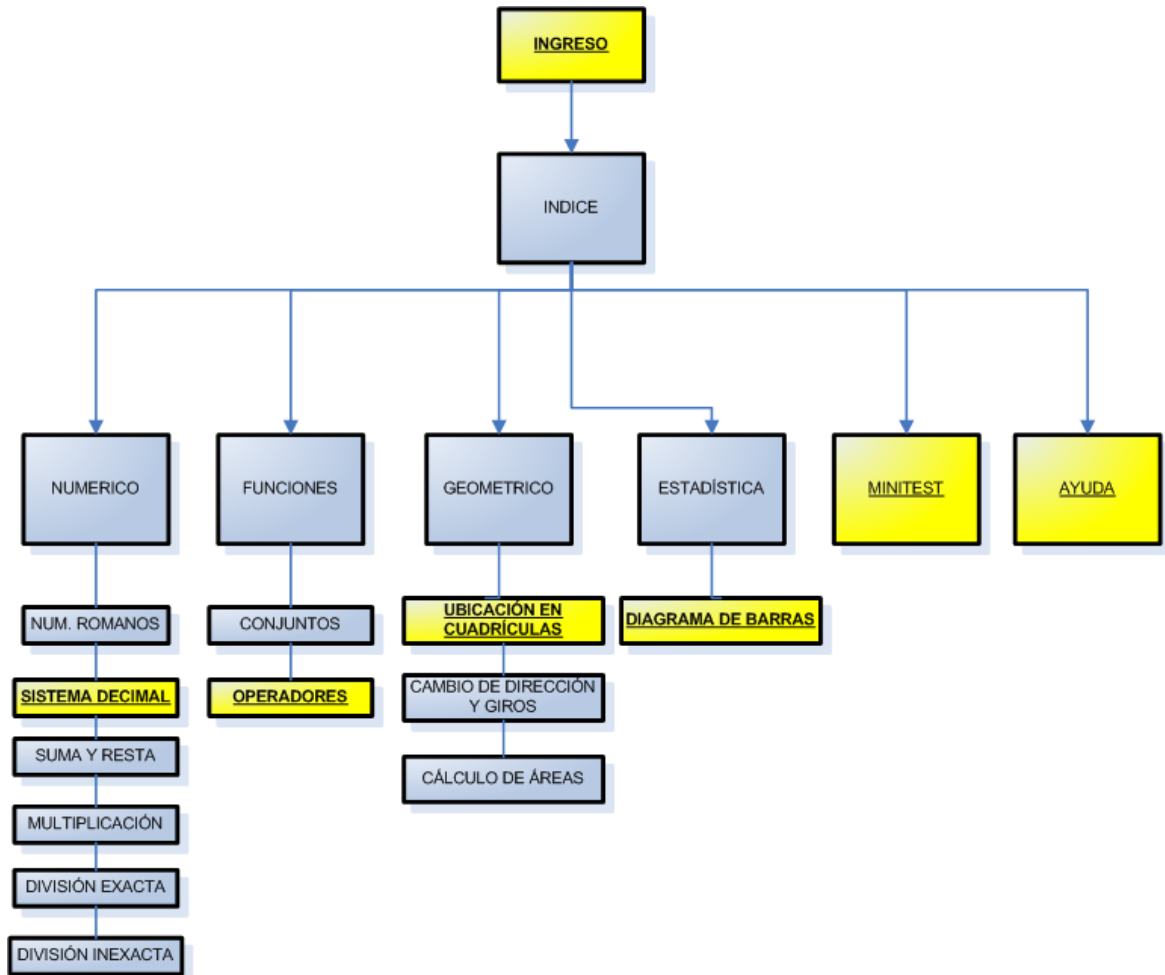


Figura 4.1 Sistema basado en Módulos

b) **Llamadas a xmls:** El sistema, además de tener diseños con un alto grado de interactividad que sofistican la relación con el usuario, permite que algunos de los contenidos de los ejemplos presentados en cada uno de los módulos, el contenido de las ayudas y las preguntas presentadas en el Minitest puedan ser modificados desde archivos xml y de acuerdo a una estructura específica. Esta estructura es detallada en el Manual de Usuario, tomando en cuenta que el contenido es personalizable pero no la estructura. Al usar xml con Flash, se abre la posibilidad de que este sistema pueda ser ampliado en el futuro. En la Figura 4.1, los módulos en amarillo y sub-lineados son los que hacen llamadas a xmls.

- c) **Textos y sonidos:** Para la mayoría de textos se ha usado la fuente Ballons. Cada diálogo de los personajes o narraciones cuentan con su texto correspondiente (en caso de que por algún motivo falte el sonido). Por otro lado, hay varios sonidos en el sistema que identifican acciones, ambiente, animales y diálogos. Cuando los personajes hablan, el sonido va de acuerdo a la boca del personaje, cuando los diálogos terminan, la boca de los personajes callan también.

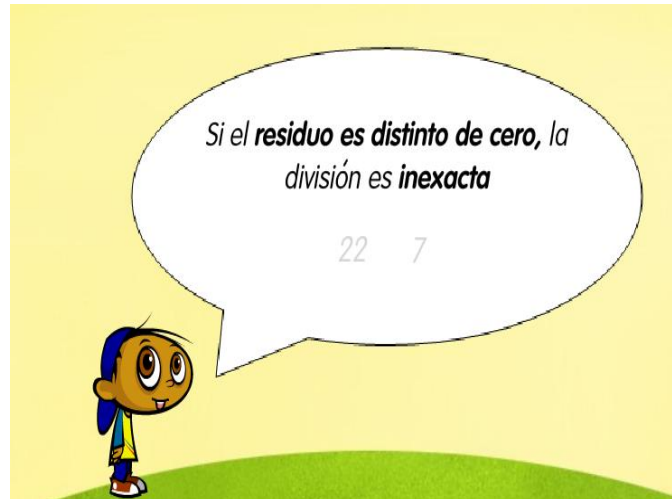


Figura 4.2 Textos y Sonidos

- d) **Colores:** Se precisa una combinación de colores estándar, cuya combinación sea jovial y a su vez sobria pero llamativa para el usuario, ya que no hay que olvidar que el público objetivo son los niños. En este software se usó el celeste, verde y tomate para ambiente y alrededores. También, cuando es oportuno, se hace referencia a colores de la patria como es caso de la camiseta de uno de los personajes (ver figura 4.3).
- e) **Gráficos:** Se ha intentado dar uniformidad en el sistema usando los mismos personajes principales en diferentes ambientes. Los personajes usados para introducir la teoría a los alumnos son: un niño mestizo, una niña mulata y un pescado tomate. Generalmente el niño se limita hacer preguntas, pero también responde pequeñas preguntas que él se hace a sí mismo o que la niña le pregunta. La niña y el pescado responden preguntas y explican teoría. Hay otros personajes secundarios que solo hacen apariciones pero que no realizan diálogos. En su

mayoría los movimientos de los personajes fueron controlados por **programación** teniendo así un movie clip (en flash) para cada personaje con diferentes opciones de movimiento dentro de cada uno de ellos.

Por otro lado, tres tipos de ambientes fueron usados: una colina, un campo y el mar. Se evitó usar el típico ambiente de la sala de aula, Por motivos ya explicados en el capítulo 1.



Figura 4.3 Niño indio y niña mulata



Figura 4.4 Pescado tomate

- f) **Botones:** Se usan varios botones en el sistema para tres funcionalidades específicas: pasar de una sección a otra, proseguir en la línea del tiempo y responder preguntas.

#### 4.1.4 Manual De Usuario

El manual de usuario detalla toda la información que el usuario necesita saber para poder manejar correctamente el software. El manual de usuario se encuentra en el Anexo 3.

Cuando se instala el software, este se encuentra en Inicio->Programas->Sistema Multimedia.

#### **4.1.5 CRONOGRAMA**

El tiempo de Implantación del software va a depender de los siguientes aspectos:

- a) Numero de máquinas donde se instalará el software
- b) Entendimiento del software por parte del profesor
- c) Entendimiento del software por parte del alumno

Asumiendo que todos los requerimientos de la sección 4.1.1 son cumplidos, se requiere **5 minutos de instalación para cada máquina.**

Si el profesor se dedica a leer el manual, se asume que el profesor tendrá un buen conocimiento del software en dos horas y podrá así transmitir estos conocimientos a los estudiantes.

Finalmente los estudiantes entenderán el software a medida que lo usen y de acuerdo a las explicaciones del profesor.

#### **4.1.6 RESPONSABLES**

Los responsables por la implantación del software pueden variar de acuerdo a cada institución pero generalmente los responsables serán los siguientes:

- a) *Director de la ESPE COMUNITARIA*: encargado de donar el software para la institución que lo solicite o necesite.
- b) *Director de la Institución Educativa*: El director que dará la autorización para la instalación del software
- c) *Técnico informático o responsable por computadores*: El que instalará el software en cada una de las máquinas donde se requiera su instalación.

- d) *Instructor de matemáticas*: El/La docente responsable de enseñar matemáticas a los niños de quinto año de básica, quien hará uso del software como herramienta para la educación.



## CAPITULO 5

### PLAN DE PRUEBAS

Como ya se observó este sistema ha seguido la metodología OOHDM, que implica las siguientes fases: Obtención de Requerimientos, Diseño Conceptual, Diseño Navegacional, Diseño de Interfaces abstractas e Implementación. Sin embargo, no se puede asumir decir que el proyecto ha sido terminado ya que se debe hacer las pruebas necesarias para garantizar que el cliente final este satisfecho con este producto. En este caso, el cliente final serán los niños y el profesor de matemáticas. Para hacer las pruebas correspondientes se probó con el quinto año de básica y su profesora de la Escuela Fiscal Acosta Soberón.

#### 5.1 Necesidad de Comprobar la Calidad del Software

La calidad del software va a determinar el tiempo de vida del producto. Evidentemente intervienen múltiples factores, como puede ser el factor económico, es decir, ante de iniciar la creación de un nuevo producto será necesario realizar un estudio económico para conocer la inversión inicial y la rentabilidad del producto. En este caso, este documento no ha puesto mucho énfasis en este estudio económico ya que la implementación de este sistema en cualquier escuela debe ser **GRATIS y** bajo la supervisión de los responsable de la ESPE COMUNITARIA o de los desarrolladores de este Sistema (Ruth García), pues el fin este software es de ayudar a la comunidad, en especial a instituciones de bajos recursos. Sin embargo, se debe considerar inversiones en máquinas, en capacitación, etc. Pero esto ya queda fuera del alcance de esta tesis ya que estos valores van a depender de las instituciones. Es por esto que, para poder medir la calidad de este producto, se ha considerado otro factor ligado al factor económico , que es el factor de ***aceptación por parte del cliente final.***

### **5.1.1 Aceptación por parte del cliente final**

El cliente final será la persona que utilice o se beneficie del nuevo producto por lo que según la percepción del mismo se generará un sentimiento de rechazo o de aceptación hacia el producto, marcando, así, su tiempo de vida en el mercado. Como ya se mencionó, en este caso serán los profesores de matemáticas de quinto año y los alumnos.

Con esta nueva perspectiva, crece la necesidad de asegurar que el producto software cumpla unos requisitos mínimos de funcionalidad, de usabilidad y de compatibilidad, es decir, se espera del producto que haga correctamente lo que tenga que hacer, que sea fácil de utilizar por el cliente y que funcione correctamente en la mayor cantidad de plataformas y sistemas posibles.

En definitiva, se trata de garantizar la calidad del desarrollo software y para ello, y antes de iniciar cualquier tarea, deben establecerse los conceptos de calidad, garantía de calidad y gestión de calidad. A partir de estas ideas se establece una metodología de calidad que deberá aplicarse durante el tiempo de vida del desarrollo software.

### **5.1.2 Pasos a seguir para las pruebas**

Inicialmente se debe escribir el plan de pruebas. Es muy importante que todas las decisiones que se tomen, es decir, estrategia a seguir, orden en la realización de las pruebas, tareas, etc estén justificadas de forma clara en el documento.

Una vez escrito el plan de pruebas, se lo sigue paso a paso, y se rellena la parte del documento relativo a los informes de análisis de cada prueba. Los fallos encontrados deberán corregirse siempre que sea posible, y modificando el código fuente si es necesario.

Una vez corregidos los errores, se volverán a realizar las pruebas que sean pertinentes, hasta que los fallos desaparezcan, y así se irá avanzando hasta completar el plan de pruebas. Hay que tener en cuenta que a medida que se va modificando el software para corregir los errores encontrados, y se vuelve a probar, hay que ir incluyendo toda esta información en el plan de pruebas. Puesto que las primeras pruebas a realizar serán las

de **caja blanca** y debido a la extensión de los módulos del sistema, se prueban solo algunos de estos de forma independiente. Una vez que las pruebas unitarias de **caja blanca** se hayan completado con éxito, se pasará a las pruebas de integración en las que se utilizará técnicas de **caja negra** y finalmente para verificar su validez, se irá a la escuela FISCAL ALBERTO ACOSTA SOBERÓN para probar con varios alumnos y profesores el sistema completo.

### **5.1.3 Definir Pruebas de Caja Blanca**

Para la realización de estas pruebas lo primero es elegir el tipo de cobertura, se dispone de varios tipos de cobertura a la hora de construir la batería de pruebas, pero se ha escogido apostar por la calidad y por tanto se debe elegir una cobertura que asegure la construcción de una batería de pruebas suficientemente exhaustiva, esto implica posiblemente descartar alguna de las posibilidades.

**Cobertura de sentencia:** una batería de pruebas proporciona cobertura de sentencia cuando garantiza que se ejecuta cada sentencia del programa al menos una vez.

**Cobertura de decisión:** una batería de pruebas proporciona cobertura de decisión cuando garantiza que en cada nodo de decisión se toma al menos una vez cada salida.

**Cobertura de condición:** una batería de pruebas proporciona cobertura de condición cuando garantiza que en cada decisión cada condición toma todos los valores posibles.

**Cobertura de decisión/condición:** una batería de pruebas proporciona cobertura de decisión/condición cuando garantiza cobertura de decisión y cobertura de condición simultáneamente.

**Cobertura de condición múltiple:** una batería de pruebas proporciona cobertura de condición múltiple cuando garantiza cobertura de decisión y cobertura de condición simultáneamente, y además se da todas las posibles combinaciones de la condición dentro de cada decisión.

Para realizar las pruebas de caja blanca se ha escogido **Cobertura de decisión**, ya que por tratarse de números es muy extenso hacer pruebas con todos los valores posibles (condición). Sin embargo, debido a la extensión de la documentación, solamente se documentará en este proyecto algunas de estas pruebas.

#### **5.1.4 Definir Pruebas de Caja Negra**

Estas pruebas se centran en los requisitos funcionales del software, para ello se construye conjuntos de datos de entrada que consigan ejercitar todos los requisitos funcionales del sistema y comprobar los datos de salida. Es importante detallar en la documentación los criterios de selección de los conjuntos de datos de entrada, su partición en clases de equivalencia, análisis de valores límite, etc. Además se han de detectar errores o ambigüedades en la interpretación de la especificación.

#### **5.1.5 Pruebas con el cliente (profesor y estudiante)**

Aquí, después de realizar las pruebas de caja blanca y caja negra y que se hayan corregido todos los errores de código encontrados en la aplicación, se irá finalmente a entrevistar al usuario final para que este pruebe el programa completo y exprese su opinión final.

#### **5.1.6 Criterios de Calidad**

Los factores implicados para la catalogación y evaluación objetiva en el momento de realizar las pruebas de caja negra y pruebas con el cliente deben cumplir con una serie de requisitos puntuales que a continuación serán detallados con lo cual se asegurará la calidad del sistema. Se verificarán dichos parámetros de la siguiente manera:

- Correcto : El sistema cumple íntegramente con el parámetro propuesto
- Moderado : El sistema cumple moderadamente con el parámetro propuesto
- Regular: El sistema cumple muy limitadamente con el parámetro propuesto
- No aplicable: El sistema no soporta el parámetro propuesto.

### 5.1.7 Definir Plan de Pruebas

El plan de pruebas planteado es el siguiente:

- a) Revisión por Director y Co-director**
- b) Hacer pruebas de caja blanca**
- c) Hacer pruebas de caja negra**
- d) Hacer pruebas con cliente**

## 5.2 Ejecución De Pruebas

### 5.2.1 Revisión por Director y Codirector

Después de finalizar el software guiado en los diagramas de OOADM y en el cronograma educativo para Matemáticas de Quinto año de básica (Anexo 1) se prosiguió a la entrevista con el director y co-director de tesis para mostrarles el producto desarrollado. La mayoría de correcciones estuvieron enfocadas en la estética y coherencia del software y no en la programación. Esta revisión fue realizada el 23 de Mayo del 2007 y se encuentra en el Anexo 2.

### 5.2.2 Pruebas de Caja Blanca

Vamos a considerar tres casos de pruebas de caja blanca.

#### 5.2.2.1 Entrada de usuario y autenticación

Para entrar al sistema se necesita ingresar un nombre de usuario y contraseña, que es validado de acuerdo al archivo xml correspondiente (ver Gráfico 4.2). Se va a garantizar que en este nodo de decisión se tome al menos una vez cada salida.

##### **a) Código de nodo de decisión:**

```
function cargarDatos(_indice:Number){
    if(clave_xml.firstChild.childNodes[_indice] != null){
        usuario = clave_xml.firstChild.childNodes[_indice].attributes.nombre;
```

```

    clave = clave_xml.firstChild.childNodes[_indice].firstChild.firstChild.nodeValue;
  }
}
indice=0;
clave_xml = new XML();
clave_xml.ignoreWhite = true;

clave_xml.onLoad = function(success) {
  if(success) {
    cargarDatos(indice);
  }
  else {
    face_mc._visible = true;
    face_mc.sad._visible = true;
    ingresar_txt.text = "Error al cargar XML:verifique archivo xml y vuelve a cargar programa";
    txtPasswd._width = 1;
    login._width = 1;
  }
}
clave_xml.load("password/seguridad.xml");

```

### **b) Salidas consideradas:**

Salidas consideradas

i. No se puede cargar XML

ii. Se ingresan datos incorrectos

iii. Se ingresan datos correctos

- i. No se puede cargar XML: La cajas de texto son deshabilitadas y aparece mensaje de error.



Figura 5.1 Entrada De Usuario y autenticación: No Se Puede Cargar XML

- ii. Se ingresa datos incorrectos : Aquí el archivo xml se ha leído exitosamente pero el usuario ingresa datos equivocados y aprieta la tecla de retorno (nombre de usuario equivocado, contraseña equivocada, o ambas). Se despliega una cara triste y sonido de error.

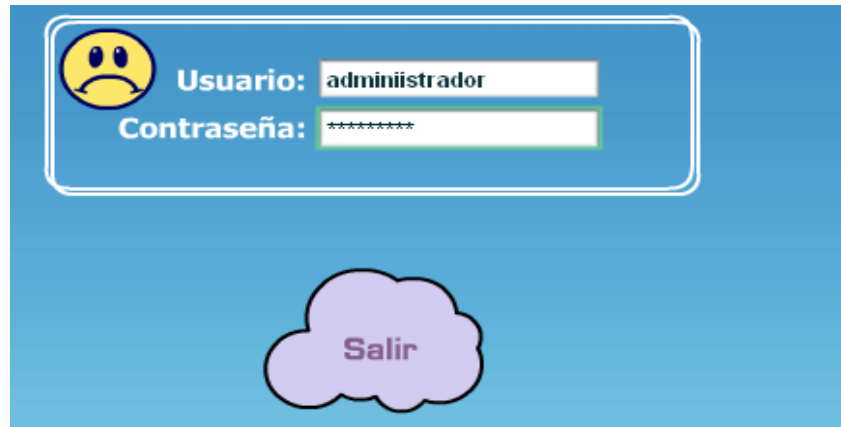


Figura 5.2 Entrada de usuario y autenticación: Datos incorrectos

- iii. Se ingresa datos correctos : Aquí el archivo xml se ha leído exitosamente y el usuario ingresa datos correctos (nombre de usuario y contraseña). Se **prosigue** a siguiente sección.



Figura 5.3 Entrada de usuario y autenticación: datos correctos

### 5.2.2.2 Ejercicios de Sistema Decimal

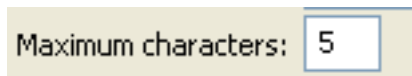
Para poder realizar los ejercicios en el Sistema Decimal, se debe leer los valores de un xml (ver Gráfico 4.2). Cuando se lee el xml solamente 3 ejercicios van a ser ingresados, 3 valores para cada ejercicio. Después de esto, se necesita validar que tipo datos son permitidos ingresar por el usuario. Una vez restringido los valores que se pueden ingresar, estos son validados de acuerdo al archivo xml correspondiente (ver Gráfico 4.2). Se va a garantizar que en este nodo de decisión (validación de datos ingresados) se tome al menos una vez cada salida.

#### a) Código de nodo de decisión:

Primero se restringen el tipo de datos que se van a poder ingresar:

```
answer1_txt.restrict = "0-9";
```

Segundo, se restringe el número de números que se pueden ingresar:



Maximum characters:

Tercero, se valida que las respuestas en el xml no pasen de 5 dígitos, si es así habrá error al cargar xml (for de 4 a 6 corresponde lectura respuestas de xml):

```
for (var i = 4; i <= 6; i++) {  
    if (success){  
        resp[i-3] = my_xml.firstChild.childNodes[i-].firstChild.nodeValue;  
        if (resp[i-3].length > 5)  
        {  
            error();  
        }  
    }  
}
```

Finalmente se agrega un keylistener (que debe ser eliminado al salir de esta sección) y este llama a diferentes funciones de acuerdo al ejercicio que se encuentre el usuario. Estas funciones también son llamadas desde el botón ingresar.

```
keyListener.onKeyUp = function(evt)  
{  
    if (Key.getCode() == 13)
```



```

        {
            trace(String(flag));
            if (flag == 1){
                validate1();
            }
            else if (flag ==2){
                validate2();
            }
            else if (flag ==3){
                validate3();
            }
        }
    }

    Key.addListener(keyListener);

    function validate1()
    {

        if (answer1_txt.text == String(resp[1])){
            mySound.start(0, 1);
            gotoAndPlay(20);
            flag =2;
        }
        else {
            mySound2.start(0, 1);
            gotoAndPlay(11);
        }

    }

    function validate2()
    {

        if (answer1_txt.text ==String(resp[2])){
            mySound.start(0, 1);
            gotoAndPlay(50);
            flag=3;
        }
        else {
            mySound2.start(0, 1);
            gotoAndPlay(40);
        }

    }

```

```
}
```

```
function validate3()
```

```
{
```

```
    if (answer1_txt.text ==String(resp[3])){
```

```
        mySound.start(0, 1);
```

```
        gotoAndPlay(80);
```

```
        flag=4;
```

```
        answer1_txt.type="dynamic";
```

```
    }
```

```
    else {
```

```
        mySound2.start(0, 1);
```

```
        gotoAndPlay(70);
```

```
    }
```

```
}
```

```
function error()
```

```
{
```

```
    ejemplo._visible =false;
```

```
    answer1_txt._visible = false;
```

```
    line_mc._visible = false;
```

```
    ingresar1._width=1;
```

```
    ingresar1._height=1;
```

```
    error_txt.text= "Error en el xml:
```

```
    verificar y cargar ejercicio nuevamente";
```

```
    stop();
```

```
}
```

```
function removekeylistener()
```

```
{
```

```
    Key.removeListener(keyListener);
```

```
}
```

## ***b) Salidas consideradas:***

- i. Se carga números mayores de 5 dígitos en XML

- ii. Se ingresan números mayores de 5 dígitos
  - iii. Se ingresan caracteres que no son números.
  - iv. Se ingresa un número incorrecto
  - v. Se ingresa respuesta correcta
- i. Se carga números mayores de 5 dígitos en XML : Se observa que cajas de texto y botón ingresar son ocultados y mensaje de error se despliega.

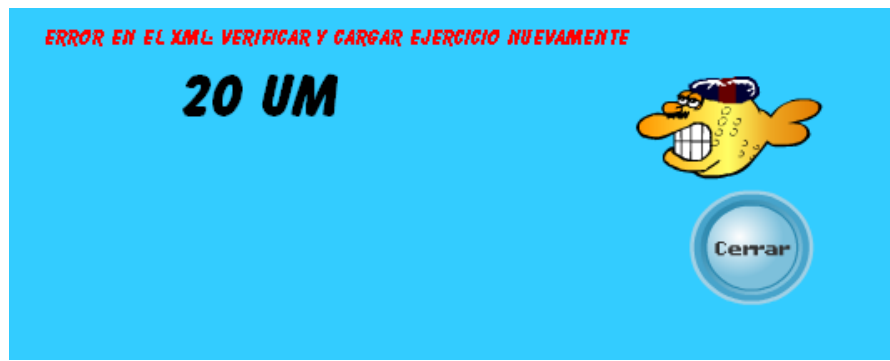


Figura 5.4 *Ejercicios de Sistema Decimal: números cargados de más de 5 dígitos*

- ii. Se ingresan números mayores de 5 dígitos: Aquí el xml ha sido cargado satisfactoriamente, pero no se puede ingresar mas dígitos aunque el usuario siga teclando.



Figura 5.5 *Ejercicios de Sistema Decimal: números ingresados de más de 5 dígitos*

- iii. Se ingresan caracteres que no son números: Aquí el xml ha sido cargado satisfactoriamente, pero no se puede ingresar caracteres que no son números aunque el usuario lo intente.

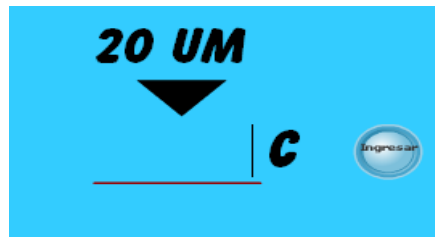


Figura 5.6 *Ejercicios de Sistema Decimal: números ingresados de más de 5 dígitos*

- vi. Se ingresa un número incorrecto: Aquí el xml ha sido cargado satisfactoriamente, pero se ingresa una respuesta incorrecta. Se observa como se despliega una cara triste con sonido de error y enseguida se borra el valor incorrecto.

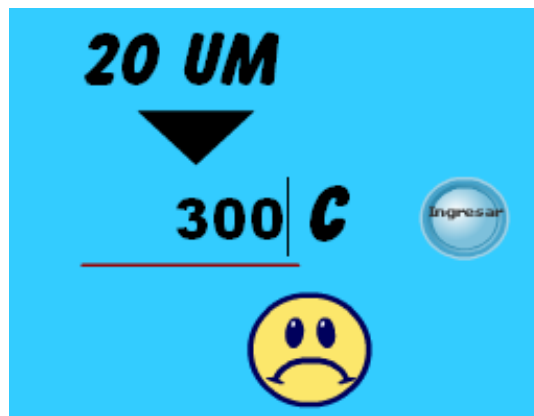


Figura 5.7 *Ejercicios de Sistema Decimal: se ingresa número incorrecto*

- vii. Se ingresa respuesta correcta: Aquí el xml ha sido cargado satisfactoriamente y se ingresa la respuesta correcta. Se observa como se despliega una cara feliz con sonido de acierto y enseguida se prosigue a siguiente ejercicio

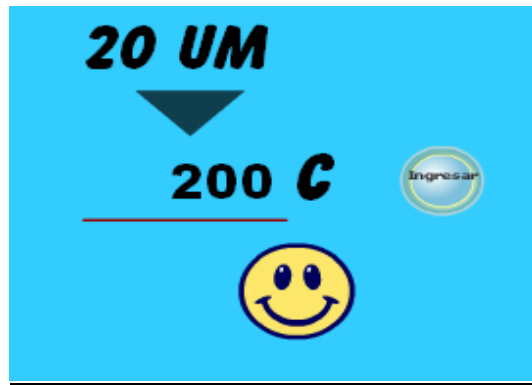


Figura 5.8 *Ejercicios de Sistema Decimal: se ingresa la respuesta correcta*

### 5.2.2.3 Minitest

Para poder responder a las preguntas propuestas por el Minitest, se debe leer los datos desde un xml (ver Gráfico 4.2). Cuando se lee el xml, únicamente 5 respuestas son propuestas para cada pregunta, de las cuáles 1 es correcta. De esta manera, se necesita validar el tipo de datos ingresados en el xml y la respuesta escogida por el usuario en el Minitest. Se va a garantizar que en este nodo de decisión (validación de datos ingresados en xml y respuesta correcta) considere al menos una vez cada salida.

#### a) *Código de nodo de decisión:*

Primero se restringen el tipo de datos que se van a poder ingresar en el xml, esto es, 1 tema, 1 pregunta y 5 respuestas. Además, la respuesta correcta no puede ser menor que 1 o 5 (indica la posición):

```
function cargarDatos(_indice:Number){
if(problema_xml.firstChild.childNodes[_indice] != null){
    problema_xml.firstChild.childNodes[_indice].attributes.tema;
    problema_xml.firstChild.childNodes[_indice].firstChild.firstChild.nodeValue;
    respuesta[1]=problema_xml.firstChild.childNodes[_indice].firstChild.nextSibling.firstChild.nodeValue;
    respuesta[2]=problema_xml.firstChild.childNodes[_indice].firstChild.nextSibling.nextSibling.firstChild.nodeValue;
    respuesta[3]=problema_xml.firstChild.childNodes[_indice].firstChild.nextSibling.nextSibling.nextSibling.firstChild.nodeValue;
    respuesta[4]=problema_xml.firstChild.childNodes[_indice].firstChild.nextSibling
```

```

ibling.nextSibling.nextSibling.nextSibling.firstChild.nodeValue;
respuesta[5]=problema_xml.firstChild.childNodes[_indice].firstChild.nextSibling.nextSibling.nextSibling.nextSibling.nextSibling.firstChild.nodeValue;
correcta=problema_xml.firstChild.childNodes[_indice].firstChild.nextSibling.nextSibling.nextSibling.nextSibling.nextSibling.nextSibling.firstChild.nodeValue;
if (Number (correcta) > 5 || Number (correcta) < 1 || correcta.length>1)
    {
        visibles(false, "Problema con xml en datos ingresados para esta pregunta.");
    }
    imagen=problema_xml.firstChild.childNodes[_indice].lastChild.firstChild.nodeValue
tema_txt.text = tema;
    pregunta_txt.text = pregunta;
    for (i=1; i<=5; i++)
    {
        eval("boton"+i).label = respuesta[i];
    }
pantalla_mc.loadMovie(imagen);
    }
    else
    {
        visibles(false, "No se ha podido cargar los datos, porfavor consulte las ayudas o verifique su archivo XML");
    }
}

siguiente_btn.onPress = function(){
    Siguiente();
}

function Siguiente()
{

    if(problema_xml.firstChild.childNodes[indice+1] != null){
        cargarDatos(indice);
    }
    else {
        visibles(false,"HA FINALIZADO EL MINITEST DESEA VOLVER AL INICIO?");
    }
}

```

```

}

indice=0
problema_xml = new XML();
problema_xml.ignoreWhite = true;
problema_xml.onLoad = function(success) {
if (success) {
        visibles(true,"");
        cargarDatos(indice);
    }
else {
        visibles(false, "No se ha podido cargar los datos, porfavor
        consulte las ayudas o verifique su archivo XML");
    }
}

problema_xml.load("minitest.xml");

boton1.onRelease = function ()
{
    boton1.selected = true;
    Radio = 1;
}

boton2.onRelease = function ()
{
    boton2.selected = true;
    Radio = 2;
}

boton3.onRelease = function ()
{
    boton3.selected = true;
    Radio = 3;
}

boton4.onRelease = function ()
{
    boton4.selected = true;
    Radio = 4;
}

```

```

        boton5.onRelease = function ()
    {
        boton5.selected = true;
        Radio = 5;
    }

function visibles(estado :Boolean, mensaje:String)

{
    titulo_txt._visible = estado;
    tema_txt._visible = estado;
    pregunta_txt._visible = estado;
    boton1._visible = estado;
    boton2._visible = estado;
    boton3._visible = estado;
    boton4._visible = estado;
    boton5._visible = estado;
    responder_btn._visible = estado;
    pantalla_mc._visible = estado;
    msg_txt._visible = not (estado);
    msg_txt.text = mensaje;
    cargar_btn._visible = not(estado);
}

```

**b) Salidas consideradas:**

- i. Respuesta correcta es mal ubicada en XML
  - ii. Se cargan más de 5 respuestas
  - iii. No se lee imagen
  - iv. Se responde incorrectamente.
  - v. Se responde correctamente
  - vi. Se termina evaluación
- 
- i. Respuesta correcta es mal ubicada en XML: La ubicación de la respuesta correcta es menor que 1, mayor que 5 o tiene más de 1 dígito en el XML.





Figura 5.9 Minitest: Respuesta correcta es mal ubicada en XML

- ii. Se cargan más de 5 respuestas: Aquí se va a leer más de 5 respuestas para la pregunta, causando problemas. Se desplegaría entonces nuevamente el gráfico
- iii. No se lee imagen: En caso de que la imagen relacionada a la pregunta no sea encontrada, solamente la pregunta será desplegada. Sin causar problemas con lo que se refiere a los demás contenidos.



Figura 5.10 Minitest: Imagen no puede ser cargada

- iv. Se responde incorrectamente: Aquí el xml ha sido cargado satisfactoriamente, pero se ha respondido mal. Una cara triste con su respectivo sonido aparece por un tiempo y se continúa en la misma pregunta, alterando la calificación.

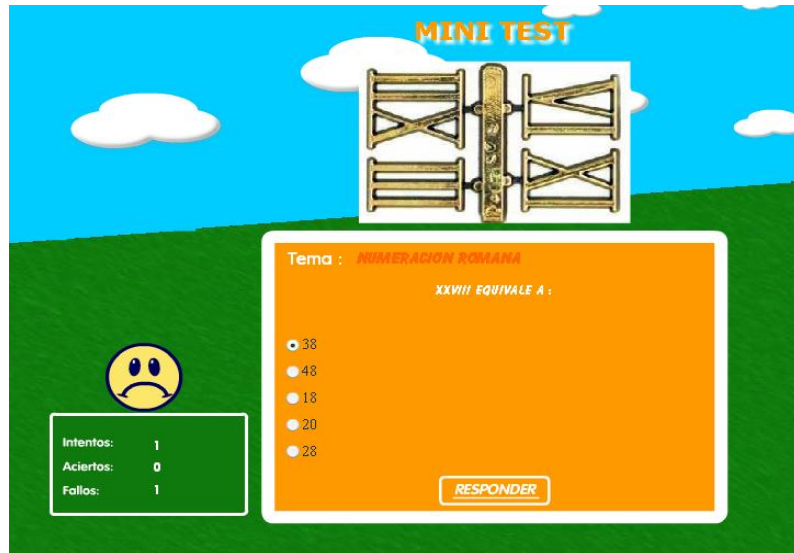


Figura 5.11 *Minitest: respuesta incorrecta*

- vii. Se termina evaluación: Aquí el xml ha sido leído en su totalidad sin presentar problemas y se ha respondido correctamente hasta llegar a la última pregunta. Se despliega un mensaje de final de evaluación, se despliega la calificación y se pregunta si se desea cargar un minitest nuevamente.



Figura 5.12 *Minitest: se llega al final de evaluación.*

### 5.2.3 Pruebas de Caja Negra

Como se mencionó las pruebas de caja negra prueban los aspectos funcionales del producto. Los parámetros considerados en estas pruebas son:

- a) **Facilidad de uso del entorno:** Agradable, fácil de usar, auto-explicativos, se debe conocer donde se encuentra en todo momento, movilidad, ayuda accesible a todo momento.
- b) **Facilidad de acceso e instalación de programas y complementos:** Instalación sencillo, rápida y transparente.
- c) **Consideración de estudiantes con necesidades educativas especiales:** Consideración a estudiantes con problemas tales como visuales, auditivos, motrices, etc.
- d) **Interés y relevancia de los aprendizajes que se ofrecen para los destinatarios:** Relevancia en objetivos educativos, contenidos interesantes y actividades.
- e) **Eficacia didáctica:** facilita cumplir objetivos, material formativo y eficaz.
- f) **Versatilidad didáctica:** ajuste a parámetros (dificultad, tiempo de respuesta, usuarios, idioma, etc), contenidos modificables, registro de la actividad de cada usuario, impresión de contenidos, proporciona informes, permite continuar los trabajos empezados con anterioridad, adaptación a entornos de uso (aula informática, clase, doméstico), adaptación a agrupamientos (individual, parejas, etc), estrategias didácticas, contextos formativos.
- g) **Fuentes de información complementaria:** múltiples enlaces externos, bibliografía, agenda, noticias.
- h) **Canales de comunicación bidireccional:** existencia de foros, consultorias.
- i) **Recursos para gestión de la información:** índices y buscadores de Internet, discos virtuales, recursos para procesar datos.
- j) **Sistema de apoyo docente y tutorial:** consultas puntuales sobre contenidos del material o completo sistema de ayuda.
- k) **Carácter completo:** proporciona todo lo necesario (contenidos temáticos, comentarios, síntesis, ejercicios de autoevaluación, ayudas, soluciones de los mismos) para realizar los aprendizajes previstos.

l) **Creditos:** Los contenidos indican la fecha de la última actualización y los autores.

m) **Editor de contenidos:** facilita a los profesores la modificaciones de contenidos.

Tabla 5.1 Pruebas de Caja Negra: Funcionalidad del Software

<b>ASPECTOS FUNCIONALES</b>	<b>VALORACIÓN</b>
Facilidad de uso del entorno	<i>Correcto</i>
Facilidad de acceso e instalación de programas y complementos	<i>Correcto</i>
Consideración de estudiantes con necesidades educativas especiales	<i>Correcto</i>
Interés y relevancia de los aprendizajes que se ofrecen para los destinatarios	<i>Correcto</i>
Eficacia didáctica	<i>Correcto</i>
Versatilidad didáctica	<i>Moderada</i>
Fuentes de información complementaria	<i>Moderada</i>
Canales de comunicación bidireccional	<i>No aplicable</i>
Recursos para gestión de la información	<i>No aplicable</i>
Sistema de apoyo docente y tutorial	<i>Correcto</i>
Carácter completo	<i>Correcto</i>
Créditos	<i>Correcto</i>
Editor de contenidos	<i>Moderado</i>

#### **5.2.4 Pruebas con el cliente (Profesor y estudiantes)**

Al entrevistar al cliente se consideró tres tipos de encuestas:

Profesores: Encuesta sobre funcionalidad del software (lo mismo la tabla anterior), encuesta sobre aspectos pedagógicos del software

Estudiantes: Encuesta sobre aspectos globales del software

##### **5.2.4.1 Encuesta sobre la funcionalidad del software**

Los resultados obtenidos fueron los siguientes:

Tabla 5.2 Pruebas con el Cliente: Funcionalidad del Software

<b>ASPECTOS FUNCIONALES</b>	<b>VALORACIÓN</b>
Facilidad de uso del entorno	<i>Correcto</i>
Facilidad de acceso e instalación de programas y complementos	<i>Correcto</i>
Consideración de estudiantes con necesidades educativas especiales	<i>Correcto</i>
Interés y relevancia de los aprendizajes que se ofrecen para los destinatarios	<i>Correcto</i>
Eficacia didáctica	<i>Correcto</i>
Versatilidad didáctica	<i>Correcto</i>
Fuentes de información complementaria	<i>Correcto</i>
Canales de comunicación bidireccional	<i>No aplicable</i>
Recursos para gestión de la información	<i>No aplicable</i>
Sistema de apoyo docente y tutorial	<i>Correcto</i>
Carácter completo	<i>Correcto</i>
Créditos	<i>Correcto</i>
Editor de contenidos	<i>Correcto</i>

#### 5.2.4.2 Encuesta sobre aspectos pedagógicos del software

El criterio de calidad es el mismo que para la tabla 4.1 y se consideran los siguientes aspectos:

- a) **Plan docente:** Claridad en la enseñanza de unidades
- b) **Motivación:** Material atractivo y divertido
- c) **Contenidos:** Coherencia, veracidad, profundidad, calidad, organización lógica, buena secuenciación.
- d) **Relevancia de los elementos multimedia:** Multimedia aporta para aprendizaje
- e) **Guías didácticas y ayudas:** Ficha resumen (características básicas), manual de usuario, guía didáctica
- f) **Flexibilización del aprendizaje:** Adaptabilidad a las características específicas de los estudiantes (estilos de aprendizaje, capacidades).

- g) Tutorización de evaluaciones:** en función de las respuestas (acertadas o erróneas) de los usuarios, se sugiere automáticamente determinados contenidos y/o actividades.
- h) Autonomía del estudiante:** Los materiales proporcionan herramientas cognitivas para que los estudiantes hagan el máximo de uso de su potencial
- i) Contenidos constructivistas:** Los contenidos del software van de acuerdo a la metodología constructivista
- j) Múltiples actividades:** se proponen múltiples y diversas actividades.
- k) Enfoque crítico/aplicativo/creativo** de las actividades
- l) Aprendizaje colaborativo:** construcción conjunta del conocimiento entre los estudiantes y recursos para ello (presentación de problemas reales, relación con día a día).
- m) Corrección de las actividades:** Las actividades son corregidas.
- n) Adecuación de los destinatarios:** Software va dirigido a niños de quinto año de básica y profesores del mismo.
- o) Evaluación de los aprendizajes:** Sistema de seguimiento y evaluación de los aprendizajes orientado al usuario.
- p) Sistema de apoyo docente y tutorial:** servicio de consultas, aulas virtuales, tutoría virtual

Los resultados obtenidos al preguntar a la profesora fueron los siguientes:

Tabla 5.3 Pruebas con el Cliente: Aspectos pedagógicos

ASPECTOS PEDAGÓGICOS	VALORACIÓN
Plan docente	<i>Correcto</i>
Motivación	<i>Correcto</i>
Contenidos	<i>Moderado</i>
Relevancia de los elementos multimedia	<i>Correcto</i>
Guías didácticas y ayudas	<i>Correcto</i>
Flexibilización del aprendizaje	<i>Correcto</i>
Tutorización de evaluaciones	<i>Correcto</i>
Autonomía del estudiante	<i>Moderado</i>
Contenidos constructivistas	<i>Correcto</i>
Múltiples actividades	<i>Moderado</i>
Enfoque crítico/aplicativo/creativo	<i>Moderado</i>

<b>Aprendizaje colaborativo</b>	<i>Correcto</i>
<b>Corrección de las actividades</b>	<i>Correcto</i>
<b>Adecuación de los destinatarios</b>	<i>Correcto</i>
<b>Evaluación de los aprendizajes</b>	<i>Correcto</i>
<b>Sistema de apoyo docente y tutorial</b>	<i>Correcto</i>

### 5.2.4.3 Encuesta sobre aspectos globales del software (Alumnos)

Los aspectos considerados son auto explicativos y el criterio de evaluación cambia para que los niños puedan entender mejor como responder.

Los resultados de la encuesta con 27 niños es la siguiente:

Tabla 5.4 Pruebas Globales: Aspectos Globales (niños)

<b>ASPECTOS GLOBALES</b>	<b>SI</b>	<b>MAS O MENOS</b>	<b>NO</b>	<b>NO SE</b>	<b>TOTAL</b>
<b>El software es divertido</b>	25	1	0	1	27
<b>Entendí los contenidos de la unidades</b>	11	10	1	5	27
<b>Pude responder las preguntas del minit</b>	20	1	6	0	27
<b>Me gusta las animaciones</b>	17	4	4	2	27
<b>Los temas se relacionan a mi realidad</b>	18	5	1	3	27
<b>Creo que aprendería mejor matemáticas con este sistema</b>	21	3	0	3	27
<b>Creo que mi escuela debe tener este sis</b>	22	3	2	0	27
<b>Creo que este software ayuda en mi educación</b>	23	2	0	2	27
<b>TOTAL</b>	157	29	14	16	

### 5.3 Resultados de Pruebas

a) **Resultados de pruebas con profesores (funcional y pedagógico) y Caja Negra son:**

Tabla 5.5 Resultados: Caja Negra y Criterios del profesor

CRITERIOS	CANTIDAD
Correcto	31
Moderado	7
Regular	0
No aplicable	4
Total preguntas	42

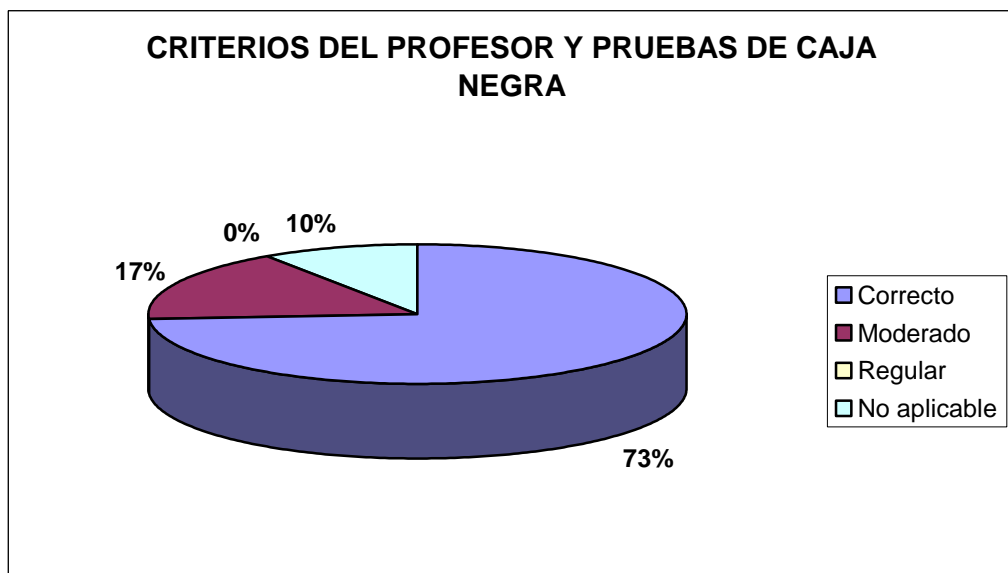


Figura 5.13 Criterios del profesor y pruebas de caja negra

**Conclusiones Resultados con Profesor y Caja Negra:**

Como se puede observar, la mayoría de los parámetros fueron cumplidos satisfactoriamente (Correcto 73%), en ningún momento se presentó una calificación de "Regular" por parte del desarrollador y de la profesora. En las calificaciones de "Moderado," 3 fueron hechas por el desarrollador en asuntos funcionales y 4 por el profesor en asuntos pedagógicos. Los 4 criterios no aplicables se refieren a los dos parámetros funcionales y corresponden a canales de comunicación bidireccional y recursos de gestión de la información. Por lo tanto, el software tiene que mejorar en asuntos pedagógicos tales como contenidos, autonomía del estudiante, múltiples



actividades y enfoque crítico, aplicativo y creativo. El parámetro "autonomía del estudiante" es totalmente entendible que reciba una calificación de "Moderado" ya que desde el inicio, se "dejó claro que este software necesita de la supervisión del profesor.

**b) Resultados de pruebas con los estudiantes**

Como se puede ver en la tabla 4.5, la mayoría de los parámetros fueron cumplidos satisfactoriamente, pues hubo 157 "Sí" contra 59 entre "Más o Menos" , "No" y "No Se." A seguir se presentan los resultados de una forma gráfica (los parámetros fueron resumidos).

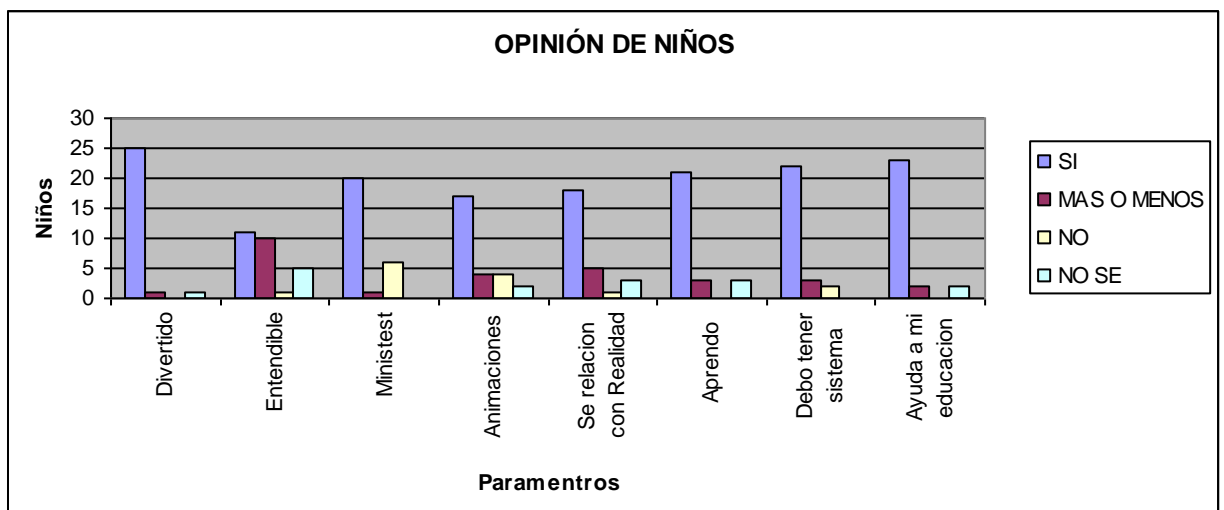


Figura 5.14 Opinión de niños con respecto al Software

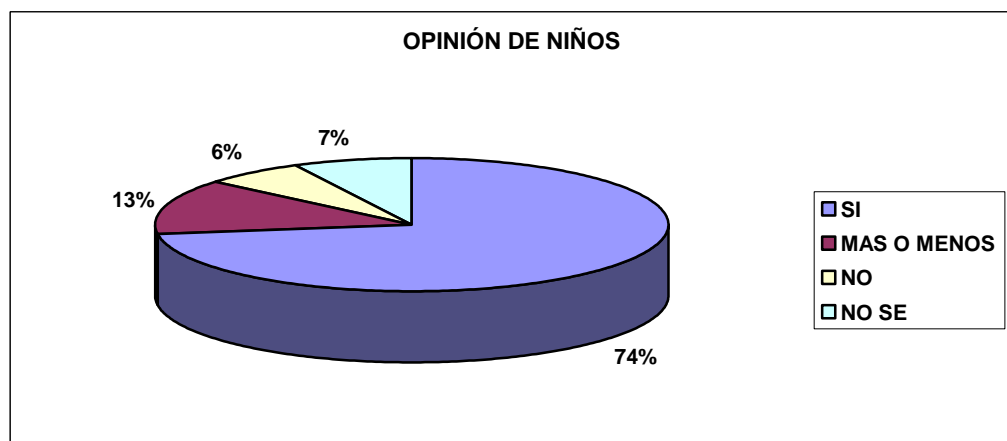


Figura 5.15 *Suma de opiniones de los niños con respecto al Software*

**Conclusiones de las pruebas con los estudiantes:**

Como se puede observar en el gráfico 4.16, la mayoría de niños respondió positivamente al software (74%). El parámetro dos, "Entendí los contenidos de las unidades," es el que presenta más problemas ya que de acuerdo al gráfico 4.15, el "Sí" y el "Más o menos" están muy próximos uno del otro. Sin embargo, esto es entendible ya que desde un principio este software ha sido diseñado para que sea supervisado por el profesor, quien debe explicar los contenidos con más detalles. Se asume que después de haberlo usado unas cuantas veces, los niños podrán manejar el software con más facilidad. También se observa que entre los "Sí" más numerosos están los parámetros "El software es divertido," "Creo que aprendería mejor matemáticas con este sistema," "Creo que mi escuela debe tener este sistema" y "Creo que este software ayuda en mi educación. Se puede decir entonces que se ha logrado cumplir el objetivo general que se propuso al inicio de este proyecto: "Desarrollar un Sistema Multimedia para la Enseñanza de Matemáticas que servirá como herramienta de apoyo para facilitar la enseñanza para los alumnos del quinto año de educación básica."

## CAPITULO 6

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1 CONCLUSIONES

- La perspectiva constructivista procura que los estudiantes puedan analizar, investigar, colaborar, compartir y generar conocimientos basados en lo que ellos ya saben contando con la ayuda del profesor como guía en el proceso de Educación. Se concluye entonces que es muy importante contar con la presencia y conocimiento del profesor aunque la enseñanza cuente con herramientas tales como sistemas multimedia. Se cree que es falso asumir que el estudiante, sobretodo estudiantes en escuela básica, puedan aprender eficientemente solamente con un sistema multimedia. Para que el proceso de aprendizaje sea eficiente el profesor debe ser el guía, sobretodo en la fase inicial.
- Este software necesita de la guía del profesor sobretodo en el momento de introducir el software para el estudiante. Se necesita de su instrucción y respaldo en la teoría de cada unidad, pues siempre cada unidad debe de estar acompañada de una conversación y debate antes de pasar a los ejercicios, para que estos puedan ser entendidos apropiadamente y de acuerdo situaciones reales planteadas.
- Con lo que se refiere al manejo del software, el profesor debe ser como el manual del software para introducir el sistema el niño ya que es muy difícil asumir que los niños lean el manual por sí solos.
- En el transcurso del desarrollo se han encontrado varias dificultades en el diseño, y que se han podido resolver gracias al método de diseño OOHDM ya que las primitivas que provee son bastante útiles. Esta metodología permite hacer modificaciones al software de una forma eficiente evitando confusiones y pérdidas de tiempo. Sin embargo, se necesita un conocimiento profundo de esta metodología para poder entenderla y aplicarla adecuadamente pero debido a que es una metodología reciente la información es un poco limitada y a veces no es muy clara. Por ejemplo, no hay documentos impresos en la biblioteca o en librerías de Quito al respecto, siendo la Internet la única fuente de consulta que, como ya se mencionó, no es muy concisa.

Posiblemente en el futuro se pueda contar con más fuentes de información sobre el OOHDM, ya que de acuerdo a la bibliografía consultada, se está utilizando cada vez más en sistemas multimedia.

- Las diferencias se hacen notorias entre los centros educativos que acceden a las Tecnologías de la Información y de la computación y aquellos que no lo pueden hacer, lo que agranda la brecha educacional y por tanto la brecha entre ricos y pobres, con las inevitables consecuencias sociales que de ello se deriva. Es por esto que esta iniciativa (ESPE COMUNITARIA) junto con pocas otras en el país permiten que esa brecha disminuya al ofrecer herramientas de educación a institutos que no pueden comprar sistemas propietarios debido a falta de recursos. Indudablemente es de gran ayuda para estos centros de educación de bajos recursos.
- La experiencia de usar XML con flash fue muy gratificante ya que XML es una manera simple por la cual representar datos pues la llamada Metadata (Datos que además de proporcionar información se describen así mismos) es útil para que los sistemas que no saben de donde proviene determinada información sepan que tipos de datos están recogiendo y de esa manera manejarlos adecuadamente. De esta forma, al usar XML permite que sin necesidad de usar un programa de administrador de contenidos el profesor pueda administrar los contenidos y modificarlos por si mismo, siendo indispensable una lectura previa del manual acerca de estos archivos. Esto obviamente no quita la posibilidad de desarrollar un programa de administrador de contenidos en el futuro.
- Al desarrollar varios gráficos animados se concluye que es mejor y más eficiente controlar la mayoría de los movimientos por programación en flash que dibujándolos uno por uno, ya que se ahorran recursos, espacio y tiempo.
- Se presentaron varias limitaciones en el momento de diseñar los personajes y paisajes ya el desarrollador de esta tesis es especializado en informática y no en diseño. Sin embargo, en este caso, se enfatizó la gran importancia que tiene una interfaz agradable para el estudiante en el momento de utilizar el software, pues es esta la que va a llamar la atención del estudiante en el momento de instruirse acerca de matemáticas. Si no se tiene un software con los gráficos, animaciones y sonidos apropiados, aunque se cuente con una fuerte programación, los resultados ciertamente no serán muy gratificantes para los niños y por ende será un software poco aplicable en esta área.

- Es importante que después del desarrollo de todo software se mida de la mejor forma posible la aceptación del cliente, pues son ellos los que dirán si el sistema desarrollado ha valido la pena. En este caso, se ha demostrado la aceptación del cliente gráficamente y se ha obtenido resultados alentadores.
- El software ha cumplido con los objetivos establecidos al inicio de este proyecto, teniendo una aceptación del 74%. En las pruebas ha presentado algunas deficiencias con lo que se refiere al contenido de las unidades. Esto es entendible ya que desde un principio el software fue diseñado para que sea una herramienta más en la enseñanza de matemáticas y que la supervisión del profesor sea fundamental, sobretodo al inicio. Como los niños fueron probados inmediatamente después de haber instalado el software, sin un entrenamiento previo, muchos de los contenidos no fueron debidamente entendidos y esto es reflejado en los resultados de las pruebas.

## 6.2 RECOMENDACIONES

- Se recomienda que este software de educación, que necesita de supervisión de un profesor para poder usarlo como un aporte a la enseñanza de matemáticas, sea expuesto desde un *proyector o pantalla* gigante al inicio del año lectivo para que el profesor(a) explique los contenidos y realice algunos ejercicios antes de que los estudiantes puedan entrar al sistema por ellos mismos. Después cuando los estudiantes ya sepan más acerca del entorno del programa, ellos podrán manejarlo con más fluidez.
- Se recomienda que en la teoría de cada unidad haya una explicación y debate al respecto, usando las situaciones reales presentadas y discutiendo otras. Sería ideal por ejemplo que el profesor a medida que vaya avanzado en la materia del año y con una planificación previa, vaya presentando los contenidos del software.
- Debido que el software es muy fácil de entender no se necesita dar un entrenamiento especializado a los profesores. Sin embargo, se recomienda que antes de hacer uso del software se lea detenidamente el manual de usuario.
- Se recomienda que, como una ampliación a este proyecto, se haga un administrador de contenidos usando todos los xmls proporcionados por el sistema.

De esta forma los profesores podrán acceder a los contenidos, modificarlos, agregarlos o eliminarlos en una interfaz gráfica y amigable.

- El proyecto ESPE COMUNITARIA es una iniciativa muy interesante y con objetivos de aportar a la educación en las instituciones de bajos recursos por medio de programas multimedia educativos. Sin embargo se recomienda que este proyecto sea mejor dirigido y coordinado por la facultad. Por ejemplo, cada estudiante(s) responsable por desarrollar una tesis para la ESPE COMUNITARIA debería seguir estándares ya impuestos por la facultad como colores, estructura, preguntas, temas, etc para tener uniformidad (al momento solo hay el loro y el himno).
- Igualmente, se recomienda tener ya los personajes animaciones listas para que los tesisistas no especialistas en diseño, no pierdan el tiempo dibujando gráficos y sí enfocándose en aspectos tales como programación, base de datos, etc. Esto no quitaría la posibilidad de crear nuevos personajes que vayan de acuerdo a los estándares establecidos.
- También se recomienda que cada modulo desarrollado por los tesisistas sea parte de un gran sistema constituido de todas las contribuciones de los mismos. De esta forma, todas las tesis serían parte de un todo funcional que se iría actualizando constantemente, evitando lo que actualmente ocurre, que es tener varios módulos independientes no uniformes mas o menos funcionales.
- De la misma forma, se recomienda que la teoría Constructivista y teoría acerca del OOHDM sea ya presentada al alumno para que estos temas tengan uniformidad. No se gana mucho teniendo varias versiones de esta pedagogía y metodología orientada a objetos en varias tesis. Al tener varias versiones se corre el riesgo de tener una incorrecta interpretación o asimilación de estas metodologías (en especial para OOHDM), causando así diferencias e incompatibilidades con los objetivos de la ESPE Comunitaria. Por otro lado, si se tiene un concepto claro acerca de la pedagogía a ser empleada así como también del OOHDM el alumno se concentra más en entender estos conceptos y realizar planes de ejercicios, de teoría y dibujar gráficos (si es el caso) enfocados a su software sin perder el tiempo en asuntos que ya fueron investigados repetidamente antes.
- Se aconseja que si se va a emplear el OOHDM como metodología de sistemas multimedia para la ESPE COMUNITARIA, se tenga manuales claros, ejemplos y si es posible una tesis al respecto.

- Debido a la falta de base de datos en el sistema y el hecho de no ser enfocado a la web tal vez el OOHDM sea muy complejo para este tipo de sistema, por lo que se recomienda tener una segunda opción para sistemas más simples.
- Se recomienda enseñar esta metodología y emplearla durante los años de estudio de la carrera.
- Se recomienda que la ESPE COMUNITARIA se una o apoye a la iniciativa EDUFUTURO ([www.edufuturo.com](http://www.edufuturo.com)) que es una iniciativa desarrollada a través de la prefectura de Pichincha (Ver Prólogo capítulo 1).

## BIBLIOGRAFIA

1. Campaña Naranjo, Xavier Francisco. **Sistema Hipermedial de Apoyo Educativo con soporte para CMS para el Área de Estudios Sociales de Quinto Año de Educación Básica**. ESPE. Quito, Sangolquí, Junio 2006
2. Chalmers, Alan F. 1988. “¿Que es esa cosa llamada Ciencia? “. [Publicación en línea]. Disponible desde Internet en :  
[http://ar.geocities.com/udesafisica/apuntes/ciencia\\_chalmers.pdf](http://ar.geocities.com/udesafisica/apuntes/ciencia_chalmers.pdf)  
[con acceso el 23-10-2006]
3. EDUFUTURO. 2007-05-25. “**Programa de Educación de la Prefectura de Pichincha**”. Proyectos. [publicación en línea]. Disponible desde Internet:  
<http://www.edufuturo.com/> [con acceso el 23-10-2006]
4. Escalona, M. J. ; Torres, J.; Mejías, M., Jurado, M. C. ; Fillerat, L. L. , “**NDT: Navigational Development Techniques**” [publicación en línea ] . Disponible desde Internet: <http://lsi.ugr.es/~gedes/actividades/Dolmen4/a11.pdf> [con acceso en Octubre 2006]
5. Fundación Chasquinet , “**Productos**” [publicación en línea]. Disponible desde Internet en : <http://www.chasquinet.org/> [con acceso en noviembre 2006]
6. Martínez, José M.; Hilerá, José R.; Martínez, Javier; Gutiérrez, José A., (1996) “**Orientación a Objetos en la Documentación Hipermedial**”. [publicación en línea]. Disponible desde Internet en :  
<http://www.ati.es/qt/LATIGOO/OOp96/Ponen6/atiao6p06.html>, [con acceso en 2007-01-21]
7. Niedo, Juana; Macedo, Beatriz (1997). “**Un Currículo Científico para Estudiantes de 11 a 14 años**”. OEI – UNESCO. Santiago. [publicación en línea]. Disponible en Internet: <http://www.campus-oei.org/oeivirt/curricie/index.html> [con acceso en 2007-02-2]



8. Lamarca Lapuente, María Jesús. "Modelo OOHDM" [publicación en línea]. Disponible en Internet en: <http://www.hipertexto.info/documentos/oohdm.htm> [con acceso 2007-03-21]
9. Lifla. "**Desarrollo de Aplicaciones Internet**" [publicación en línea]. Disponible desde Internet: <http://www.lifla.net> [con acceso 2006-09-12]
10. Martínez, José M.; Hilera, José R.; Martínez, Javier; Gutiérrez, José A.. Agosto 2006. "**Orientación a Objetos en la Documentación Hipermedia**". Universidad de Alcalá de Henares. [publicación en línea]. Disponible desde Internet: <http://www.ati.es/gt/LATIGOO/OOp96/Ponen6/atio6p06.html> [con acceso 2007-04-14]
11. Oña Pardo, Fernando. "**La educación está lejos de ser un 'objetivo de desarrollo del milenio'**". [publicación en línea]. Disponible desde Internet: <http://www.voltairenet.org/article128898.html> [con acceso en 2006-10-22]
12. Pardo Oña, Fernando, "**La Educación en el Ecuador**". Voltairenet.org. [publicación en línea]. Disponible desde Internet: <http://www.voltairenet.org/article128898.html#article128898> [con acceso 2005-10-22]
13. Saiz, Irma Elena ; Acuña, Nelci Noemí (2006) "**Matemáticas: Aportes para la Enseñanza en el Nivel Medio**" [publicación en línea]. Disponible desde Internet: <http://aportes.educ.ar/matematica/nucleo-teorico/recorrido-historico/index.php>
14. Schwabe, Daniel ; Gustavo Rossi (1998). "**An Object Oriented Approach to Web-Based Application Design**". PUC-RIO. Departamento de Informática. Brasil. [publicación en línea]. Disponible desde Internet: <http://www-di.inf.puc-rio.br/schwabe/papers/TAPOSRevised.pdf> [con acceso 2007-02-21]
15. Schwabe, Daniel; Rossi, Gustavo. 1995, Agosto. "**The Object-Oriented Hypermedia Design Model**". [publicación en línea]. Disponible desde Internet en: <http://www.telemidia.puc-rio.br/oohdm/oohdm.html> . [con acceso 2006-11-23]

16. Schwabe, Daniel; Gustavo Rossi, “**Systematic Hypermedia Application Design with OOHDM**”. [publicación en línea]. Disponible desde Internet en: <http://www-di.inf.puc-rio.br/schwabe/HT96WWW/section1.html> [con acceso 2007-02-21]
17. Universidad de Concepción. “**Educación Interactiva**” [publicación en línea]. Disponible desde Internet en: <http://www2.udec.cl/mirmex/proyecto.htm>, [con acceso 2006-08]
18. Schwabe Daniel, Almeida, Pontes; Rita Isabela Moura . “**OOHDM-Web: An Environment for Implementation of Hypermedia Applications in the WWW,**” [web en línea] Dept. of Informatics. PUC-Rio. [publicación en línea]. Disponible desde Internet en:  
<http://www-di.inf.puc-rio.br/schwabe/papers/SigWeb-OOHDMWeb.pdf>  
[con acceso 2007-02-20]
19. Soto De Giorgis, Ricardo; Palma Muñoz, Wenceslao ; Roncagliolo De La Horra, Silvana . 2002, Noviembre “**Propuesta de un modelo navegacional para el desarrollo de aplicaciones basadas en OOHDM**”. Escuela de Ingeniería Informática, Universidad Católica de Valparaíso, Chile. [publicación en línea]. Disponible desde Internet en:  
[http://www.inf.ucv.cl/~rsoto/papersPUCV/Propuesta\\_de\\_un\\_modelo\\_navegacional.pdf](http://www.inf.ucv.cl/~rsoto/papersPUCV/Propuesta_de_un_modelo_navegacional.pdf) [con acceso 2007-05-22]

## **ANEXOS**

### **ANEXO A - PLANIFICACIÓN MICROCURRICULAR PARA EL ÁREA DE MATEMÁTICAS DEL QUINTO AÑO DE BÁSICA**





















## **ANEXO B: MANUAL DE USUARIO**

# MANUAL DE USUARIO

## **A. ACERCA DEL SISTEMA**

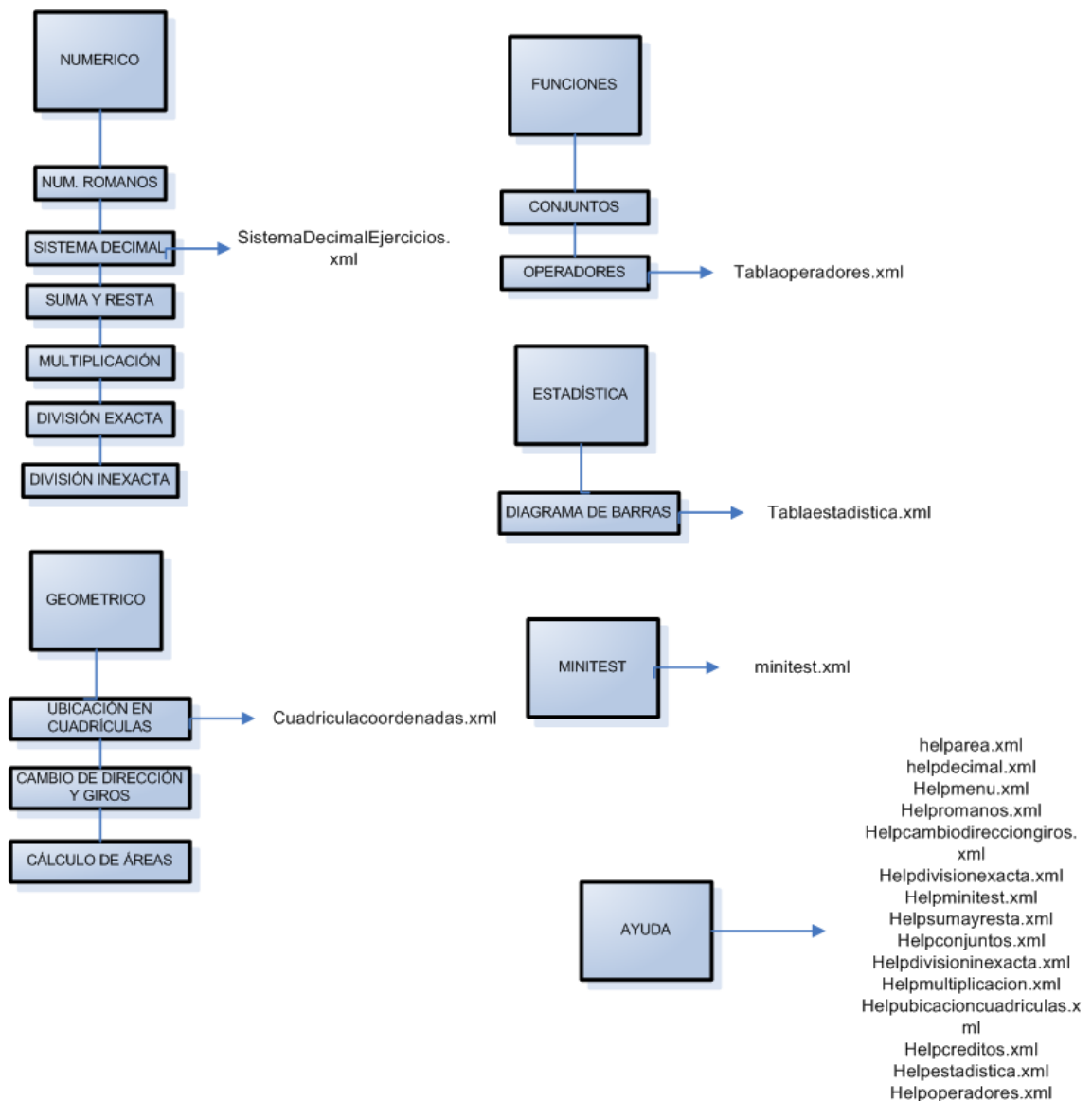
Este sistema intenta ser una herramienta de ayuda para los profesores del área de matemáticas del quinto año de básica en el momento de enseñar a los alumnos esta materia. Para lograr este objetivo, en cada capítulo se ha presentado una introducción a cada tema y se realiza ejercicios relacionados a cada uno de los temas. En ciertas ocasiones, el contenido de los ejercicios puede ser cambiado por los profesores o responsables de la clase. Además se proporciona un Minitest, cuyas preguntas pueden ser modificadas por los profesores y la ayuda es accesible desde cualquier parte del sistema.

Por otro lado, se ha intentado usar personajes que representan a la mayoría de la población Ecuatoriana, en este caso una mujer mulata y un niño mestizo además de usar un pescado en una ocasión. Estos personajes aparecen en ambientes diferentes a la típica sala de aula de una escuela, tales como una colina, un campo y el mar. Se ha usado ambientes diferentes a la de una escuela intencionalmente, intentando relacionar problemas de matemáticas con situaciones del día a día , siguiendo así la filosofía de enseñanza constructivista.

## **B. ARCHIVOS XML**

### **B.1. Archivos XML Existentes**

XML significan en inglés de eXtensible Markup Language («lenguaje de marcas extensible»), este formato de archivos son generalmente usados para internet, sin embargo no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas, como es el caso de este sistema. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. En el sistema, estos archivos fueron usados:



## Anexo B. 1 Archivos XML

### B.2. Para recuperar archivos XML

En caso de que algún conflicto ocurra con algún archivo XML, sustituirlo con el archivo correspondiente que se encuentra en la carpeta Respaldo(Archivos de Programa/Sistema Multimedia/Respaldo). Los archivos volverán a ser los mismos de la instalación inicial.

## C. INSTALAR SISTEMA

Es muy fácil instalar este software. Tan solo hay que seguir las instrucciones del instalador y el software estará instalado.

Al inicio tendremos una pantalla de Bienvenida, aplastamos siguiente. Después leemos el “Acuerdo de Licencia”. Aceptamos si estamos de acuerdo para proseguir con la instalación. Si no, no la aceptamos y la instalación se cancela.

En la siguiente pantalla, escribimos la información del usuario para “Nombre” y “Compañía.”

A seguir, escogemos la carpeta de instalación. Se aconseja que dejemos la carpeta que el instalador sugiere por defecto: C:\Archivos de Programa\Sistema Multimedia. Es ahí donde el sistema va a ser instalado.

Finalmente, aparece la ventana con todas las opciones que hemos escogido e información que hemos dado y si todo está correcto proseguimos con la instalación.

Si todo ha salido bien, obtendremos la siguiente pantalla. Debemos aplastar entonces, “Terminar.”



## Anexo B. 2 Instalación Completada

### D. INGRESO AL SISTEMA

#### D.1. Ingresar Nombre de usuario y Contraseña

Al iniciar el sistema, es importante ingresar un nombre de usuario y contraseña para poder tener acceso a los contenidos de matemáticas del software. Si no se ingresa los datos correctos, entonces no se podrá acceder al resto del sistema. El nombre de usuario y contraseña se encuentran en el



archivo seguridad.xml dentro de la carpeta password y es desde ahí que el sistema lee los datos para hacer la validación correspondiente.

Cuando se ingresa el nombre de usuario y la contraseña, se debe aplastar la tecla de retorno (tecla enter) y el sistema hará la validación automática.



#### Anexo B. 3 Ingreso de Nombre de usuario y contraseña

Si son datos incorrectos aparecerá una cara triste y el sonido de error.



#### Anexo B. 4 Ingreso de Nombre de usuario y contraseña equivocados

Si es correcto se proseguirá a la siguiente sección donde se despliega el video de ESPE COMUNITARIA. Para seguir al Menu principal se debe aplastar el boton "Siguiente" antes o después que este video se termine.



Anexo B. 5 Ingreso de Nombre de usuario y contraseña validos

## D.2. Modificar Nombre de usuario y Contraseña

Si se quiere modificar el nombre de usuario y contraseña, ir a la carpeta donde el sistema fue Instalado (Generalmente se encuentra en la carpeta de Archivos de Programa de Windows dentro de la carpeta Sistema Multimedia), abrir la carpeta password y editar el archivo seguridad.xml. Hay varias formas de abrirlo, la forma más facil es dar un clic derecho en el archivo , selección “Abrir con” y escojer la aplicación Notepad. Enseguida se abrirá el siguiente contenido:

```
seguridad.xml - Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0" encoding="iso-8859-1"?>
<SEGURIDAD>
  <usuario nombre="administrador">
    <clave>matematicas</clave>
  </usuario>
</SEGURIDAD>
```

Anexo B. 6 Ingreso de Nombre de usuario y contraseña validos

De esta forma, si se quiere cambiar el nombre de usuario se tendrá que cambiar el nombre adentro de las comillas *administrador*. Si se quiere cambiar la clave se debe cambiar lo que se encuentra entre <clave> y </clave>. *Se debe tomra cuidado para no hacer ningún otro cambio*. Cuando se halla finalizado los cambios, guardar y salir.

## E. MENU PRINCIPAL

### E.1. Identificando los elementos del Menú Principal

El menú principal contiene 4 secciones referentes a los temas de matemáticas del quinto año, donde al hacer click en cada sección se desplegarán sus contenidos (capítulos o módulso) en la nube grande.

El menú principal también contiene botones para acceder al Minitest, Creditos y Ayuda.

Además se puede optar por suspender el sonido del sistema y volverlo a habilitar cuando se crea necesario.

La siguiente figura demuestra el Menú Principal.



## E.2. Identificando Botones básicos comunes fuera del Menú Principal

Fuera del Menú Principal, cuando se entre en un capítulo cualquiera, siempre las siguientes opciones van a estar disponibles:



: **Botón de ayuda** que proporciona información en un ventana desplazable acerca de a la sección donde se encuentre el usuario.



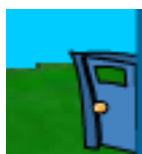
: **Botón de regreso** a Menu Principal.



: **Botón de Creditos** se despliega las personas responsables del desarrollo del sistema.



: **Botón de Sonido** donde de acuerdo a la sección habilitará o suspenderá el sonido.



: **Botón regreso a Submenú.** Se regresa al submenú del capítulo correspondiente.



: **Botón salida total.** Se cierra el sistema completamente.

## F. SECCIÓN NUMÉRICO

### F.1. Ingresar a Capítulo Numérico

Si se quiere ingresar a los contenidos de la sección “Numerico”, se debe hacer click sobre la nube correspondiente y todos los capítulos de esta sección aparecerán en la nube principal.



Anexo B. 8 Sección Numérico

### F.2. Ingresar a Números Romanos

Se hace click en el primer capítulo, **Números Romanos**, y se obtiene el menú de este capítulo donde se encuentran: botón Concepto y botón Ejercicio.

#### F.2.1. Concepto:

Aquí se hace una pequeña introducción a cómo y donde se usaban los números romanos en la antigüedad. Enseguida, se despliegan los números romanos básicos y sus correspondientes números decimales.

#### F.2.2. Ejercicio:

Aquí se proponen tres reglas con sus respectivos ejercicios al final de cada una. Si al realizar el ejercicio, el usuario escoje la opción correcta una cara feliz aparecerá con el sonido de aprobación. Si se escoje la opción incorrecta, una cara triste aparecerá con su respectivo sonido de error.

### **F.3. Ingresar a Numeración Decimal**

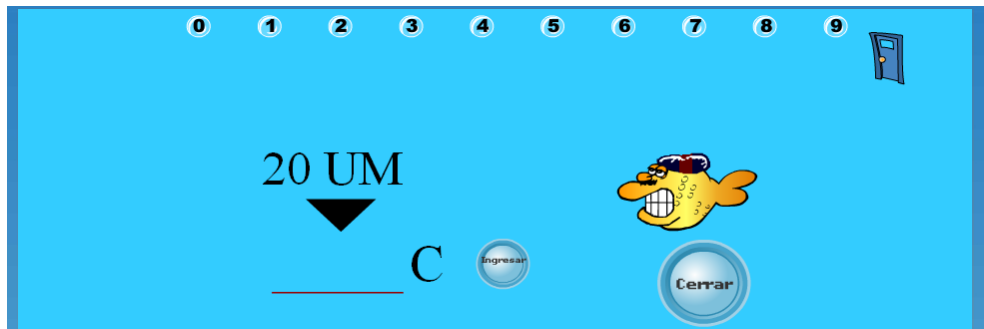
Se hace click en el segundo capítulo de esta sección, ***Numeración Decimal***, y se obtiene el menu de este capítulo donde se encuentran: botón Concepto y botón Ejercicio.

#### **F.3.1. Concepto:**

Aquí se hace una pequeña introducción acerca de cuales son los símbolos de la Numeración Decimal, enseguida se consideran dos reglas que son desplegadas automáticamente. Sin embargo, después de que estas dos reglas han sido presentadas el usuario tiene la opción de volverlas a escucharlas.

#### **F.3.2. Ejercicio:**

Aquí se proponen unicamente **tres** ejercicios donde se pregunta por equivalencias numéricas. Por ejemplo, cuantas DECENAS (D) tienen 20 UNIDADES DE MIL (UM). Cualquiera de estas tres preguntas pueden ser modificadas por el usuario mientras se está en la aplicación, pero tendrán efecto una vez que el archivo xml correspondiente (ver Gráfico 1) haya sido guardado y se aplaste nuevamente el botón de Ejercicios desde el capítulo o se ingrese directamente a los Ejercicios desde el submenú.



### Anexo B.5 Ejercicios de Sistema Decimal

#### F.3.3. Modificar el archivo xml :

Para modificar el archivo xml, se ingresa a la carpeta donde se guardó el Sistema (generalmente esta carpeta está en Archivos de Programa/Sistema Multimedia), y se accede al archivo ***SistemaDecimalEjercicios.xml*** (por ejemplo, haciendo clic derecho en el archivo y escogiendo Notepad para editarlo).

```

<Cuadrícula>
  <Ejemplo1>20 UM</Ejemplo1> 1
  <Ejemplo2>4000 U</Ejemplo2>
  <Ejemplo3>25 DM</Ejemplo3>
  <Respuesta1>200</Respuesta1> 2
  <Respuesta2>400</Respuesta2>
  <Respuesta3>2500</Respuesta3>
  <Unidad1>C</Unidad1> 3
  <Unidad2>D</Unidad2>
  <Unidad3>D</Unidad3>
</Cuadrícula>
  
```

### Anexo B.10 Modificando SistemaDecimalEjercicios.xml/

Los contenidos entre <Ejemplo1> y </Ejemplo1> encerrado en el círculo 1 corresponden a la 1 Pregunta del ejemplo. En este caso serían 20 UM (Unidades de Mil). Cabe recalcar que las unidades deben ser escritas con sus letras iniciales (UM) y no en palabras (Unidades de Mil) y se debe dar un **espacio** después del

número. Lo mismo se aplica para <Ejemplo2> y </Ejemplo2> y <Ejemplo3> y </Ejemplo3>.

Las respuestas, es decir, lo que se espera que el estudiante escriba (circulo 2) están encerradas entre <Respuesta#> y </Respuesta#> y las unidades para cada una de estas Respuestas (circulo 3) están entre <Unidad#> y </Unidad#>.

De esta forma, por ejemplo, al hacer click en Ejercicios se va a desplegar el Gráfico 8. 20 UM corresponde a <Ejemplo1>**20 UM** </Ejemplo1> y C (Centenas) corresponde a <Unidad1>**C**</Unidad1> . Cuando el usuario ingrese el equivalente, el sistema comprobará que este valor sea igual a **200**. Si es correcto se desplegará al siguiente ejercicio que, en este caso, es **4000 U** para la pregunta y **D** para el espacio de la respuesta que validará con **400**, etc.

Recordar que los equivalentes para las Unidades posibles son :

U : Unidades

D : Decenas

C : Centenas

UM : Unidad de Mil

DM : Centena de Mil

#### **F.4. Ingresar a Suma y Resta**

Se hace clic en el tercer capítulo de esta sección, **Adición y Sustracción**, y se obtiene el menu de este capítulo donde se encuentran botones para :

- a) Suma : Regla 1, Regla 2, Regla 3 y Ejercicio
- b) Resta : Regla 1, Regla2 y Ejercicio.

##### **F.4.1. Concepto:**

Los Concepto de Suma y Resta están adentro de las Reglas. En la suma se revisa el concepto acerca de la agrupación de números, parentesis y



orden de sumandos. En la resta se establece la regla de comprobación de una sustracción y se plantea un caso de suma y resta por agrupación.

#### **F.4.2. Ejercicio :**

En el caso de los ejercicios de la Suma y Resta, se plantean sumas y restas randomicas contando el número de intentos, aciertos y fallos.

### **F.5.Ingresar Multiplicación Simple**

Se hace clic en el cuarto capítulo de esta sección, ***Multiplicación Simple***, y se obtiene el menu de este capítulo donde se encuentran botones para : Concepto y Ejercicio.

#### **F.5.1. Concepto :**

Los Concepto de Multiplicación son presentados con varios ejemplos sobre cómo se realiza una multiplicación, el triple y cuádruple. Además, cuando los Concepto se terminen, se puede acceder a los ejercicios sin necesidad de volver al menú.

#### **F.5.2. Ejercicio :**

En este caso, los ejercicios también ayudan a establecer Concepto pidiendo respuestas a los usuarios. Los ejercicios hacen que el usuario aprenda como se deben multiplicar los numeros por 10, 100, 1000 y 10000 dando un ejemplo con floreros y espacio para responder. Después de esto se realiza un ejemplo de como una multiplicación puede ser agrupada de diferentes maneras y tener el mismo resultado.

### **F.6.Ingresar División Exacta**

Se hace clic en el quinto capítulo de esta sección, ***División Exacta***, y se obtiene el menu de este capítulo donde se encuentran los siguientes Botones: Regla 1, Regla 2, Regla 3 y Ejercicio.

### **F.6.1. Concepto :**

Los Concepto de la División Exacta están reflejados en las tres Reglas. Se identifica el nombre de los terminos de la División exacta, se realiza un ejemplo de una división exacta y se establece que una multiplicación simple tiene dos divisiones exactas correspondientes.

### **F.6.2. Ejercicios :**

En este caso, se plantean divisiones exactas randómicas contando el número de intentos, aciertos y fallos.

## **F.7.Ingresar División Inexacta**

Se hace clic en el sexto capítulo de esta sección, ***División Exacta***, y se obtiene el menu de este capítulo donde se encuentran los siguientes Botones: Regla 1, Regla 2 y Ejercicio.

### **F.7.1. Concepto :**

Los Concepto de la División Inexacta están reflejados en las dos Reglas. Se establece cuando una división es inexacta y se hace un ejemplo al respecto.

### **F.7.2. Ejercicios :**

En este caso, se plentean divisiones inexactas y se observa como estas son resueltas , encerrando sus residuos.

## **G. SECCIÓN FUNCIONES**

### **G.1. Ingresar a Capítulo Conjuntos**

Se hace clic en el primer capítulo de esta sección, ***Conjuntos***, y se obtiene el menu de este capítulo donde se encuentran los siguientes Botones: Concepto y Ejercicio.

### G.1.1. Concepto :

Aquí se definen los conjuntos realizando varios ejemplos con pelotas.

### G.1.2. Ejercicios :

Aquí se realizan operaciones con Conjuntos, esto es, unión, intersección, resta y suma. Se hace la pregunta, se demuestran los conjuntos y se da espacio para responder.

## G.2. Ingresar a Capítulo Operadores

Se hace clic en el primer capítulo de esta sección, **Operadores**, y se obtiene el menú de este capítulo donde se encuentran los siguientes Botones: Concepto y Ejercicio.

### G.2.1. Concepto :

Aquí se define el concepto de operadores y operadores numéricos a través de ejemplos. Se recalca que los operadores son acciones que causan una consecuencia.

### G.2.2. Ejercicios :

Se plantea una tabla donde se pide al usuario ingresar los factores desconocidos para los diferentes operadores numéricos dados (multiplicativo, aditivo y sustractivo). Esta tabla puede ser modificada después de haber guardado el archivo xml correspondiente (ver Gráfico 1) y al entrar nuevamente a este Ejercicio.



INGRESAR						
Antes	Operador Multiplicativo	Después	Operador Aditivo	Después	Operador Sustractivo	Después
3	<input type="text" value="6"/>	18	<input type="text" value="9"/>	19	<input type="text" value="17"/>	17
1	<input type="text" value="3"/>	3	<input type="text" value="6"/>	6	<input type="text" value="0"/>	0
5	<input type="text" value="2"/>	10	<input type="text" value="14"/>	14	<input type="text" value="10"/>	10
6	<input type="text" value="4"/>	24	<input type="text" value="26"/>	26	<input type="text" value="19"/>	19

Anexo B.6. Ejercicio de Operadores

### G.2.3. Modificar el archivo xml :

Para modificar el archivo xml, se ingresa a la carpeta donde se guardó el Sistema (generalmente esta carpeta está en Archivos de Programa/Sistema Multimedia), y se accede al archivo **tablaoperadores.xml** (por ejemplo, haciendo clic derecho en el archivo y escogiendo Notepad para editarlo).

```

<tabla>
<antes1>3</antes1>
<antes2>1</antes2>
<antes3>5</antes3>
<antes4>6</antes4>

<despuesmultiplicacion1>18</despuesmultiplicacion1>
<despuesmultiplicacion2>3</despuesmultiplicacion2>
<despuesmultiplicacion3>10</despuesmultiplicacion3>
<despuesmultiplicacion4>24</despuesmultiplicacion4>

<despuessuma1>19</despuessuma1>
<despuessuma2>6</despuessuma2>
<despuessuma3>14</despuessuma3>
<despuessuma4>26</despuessuma4>

<despuesresta1>17</despuesresta1>
<despuesresta2>0</despuesresta2>
<despuesresta3>10</despuesresta3>
<despuesresta4>19</despuesresta4>
</tabla>

```

1	Operador Multiplicativo	2	Operador Aditivos	3	Operador Sustractivo	4
Antes		Después		Después		Después
3	x 6 ☺	18	+ ☐	19	- ☐	17
1	x 3 ☺	3	+ ☐	6	- ☐	0
5	x 2 ☺	10	+ ☐	14	- ☐	10
6	x ☐	24	+ ☐	26	- ☐	19

Anexo B. 7. Modificando tablaoperadores.xml

El Gráfico 11 indica como los valores del archivo xml son representados en la tabla. Por ejemplo, el primer número de la columna antes corresponde a `<antes1>3</antes1>`. Hay que tener cuidado para que los valores ingresados sean coherentes. Por ejemplo, los valores en la columna 4 deben ser siempre menores que la columna 3 (porque operadores sustractivos son aplicados). Similarmente, los valores de la columna 3 deben ser mayores o iguales que la columna 2 ya que se aplica operadores aditivos, etc. Es muy importante que únicamente se modifiquen los *números* en este archivo xml.

## H. SECCIÓN GEOMÉTRICO

### H.1. Ingresar a Capítulo Ubicación Cuadrícula

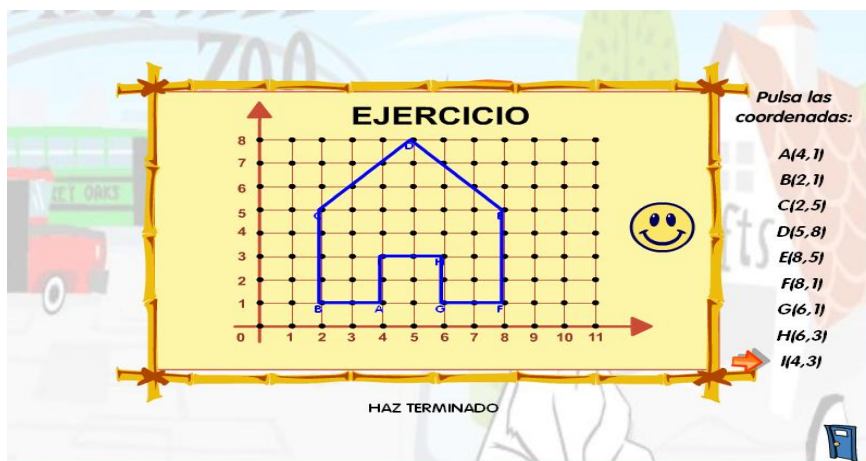
Se hace clic en el primer capítulo de esta sección, **Ubicación Cuadrícula**, y se obtiene el menú de este capítulo donde se encuentran los siguientes Botones: Concepto y Ejercicio.

#### H.1.1. Concepto:

Aquí, a través de un paseo al zoológico, se indica direcciones (Norte, Sur, Este y Oeste) y como se encuentran puntos en una cuadrícula (coordenada horizontal es primero que la coordenada vertical).

#### H.1.2. Ejercicio:

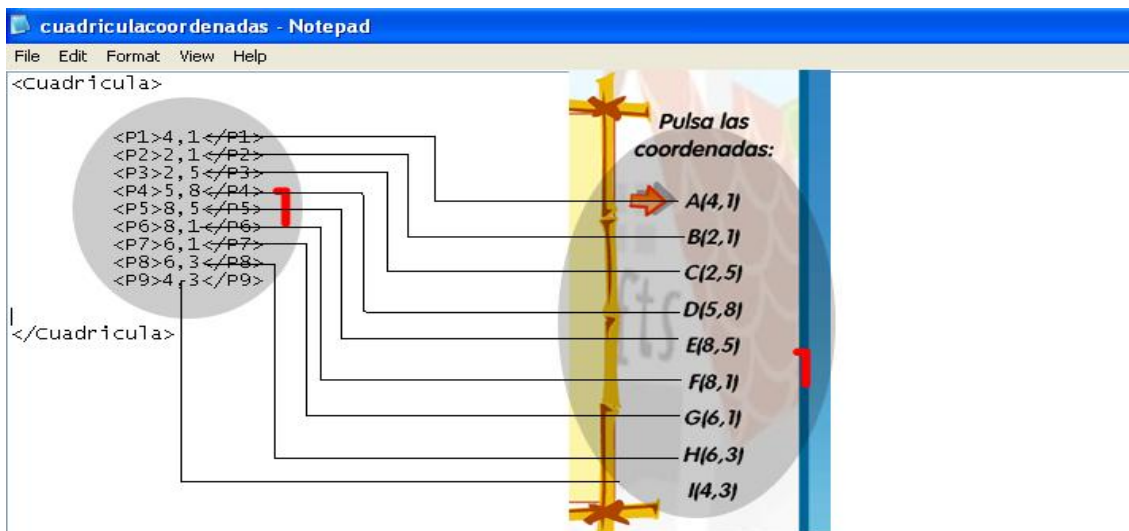
En una cuadrícula, se pide al usuario que ubique 9 puntos de una forma ordenada, es decir, no se prosigue al siguiente punto hasta que se haya acertado el anterior. Se aconseja que para que sea más divertido para el niño, estos puntos formen una imagen. Por ejemplo, en Gráfico 12 los puntos forman una casita. Considerar que después de acertados los 9 puntos, automáticamente se liga el último punto con el primero. Estos puntos pueden ser modificados después de haber guardado el archivo xml correspondiente (ver Gráfico 1) y al entrar nuevamente a este Ejercicio.



Anexo B.8 Ejercicio Ubicación Cuadrículas

### H.1.3. Modificar el archivo xml :

Para modificar el archivo xml, se ingresa a la carpeta donde se guardó el Sistema (generalmente esta carpeta está en Archivos de Programa/Sistema Multimedia), y se accede al archivo **cuadriculacoordenadas.xml** (por ejemplo, haciendo clic derecho en el archivo y escogiendo Notepad para editarlo).



#### Anexo B.9 Modificando cuadriculacoordenadas.xml

El Gráfico 13 indica como los valores del archivo xml son representados en los puntos de la cuadrícula. Por ejemplo, el primer punto A(4,1) corresponde a <P1>4,1</P1>. Hay que tener cuidado para que los valores ingresados sean coherentes. Por ejemplo, los valores en la línea horizontal van desde 0 a 11 mientras que en la vertical van de 0 a 11 (Ver Gráfico 12). Valores mayores o menores a estos crearían conflictos. Es muy importante que únicamente se modifiquen los *números* en este archivo xml, conservando las comas.

## H.2. Ingresar a Capítulo Cambio de Dirección y Giros

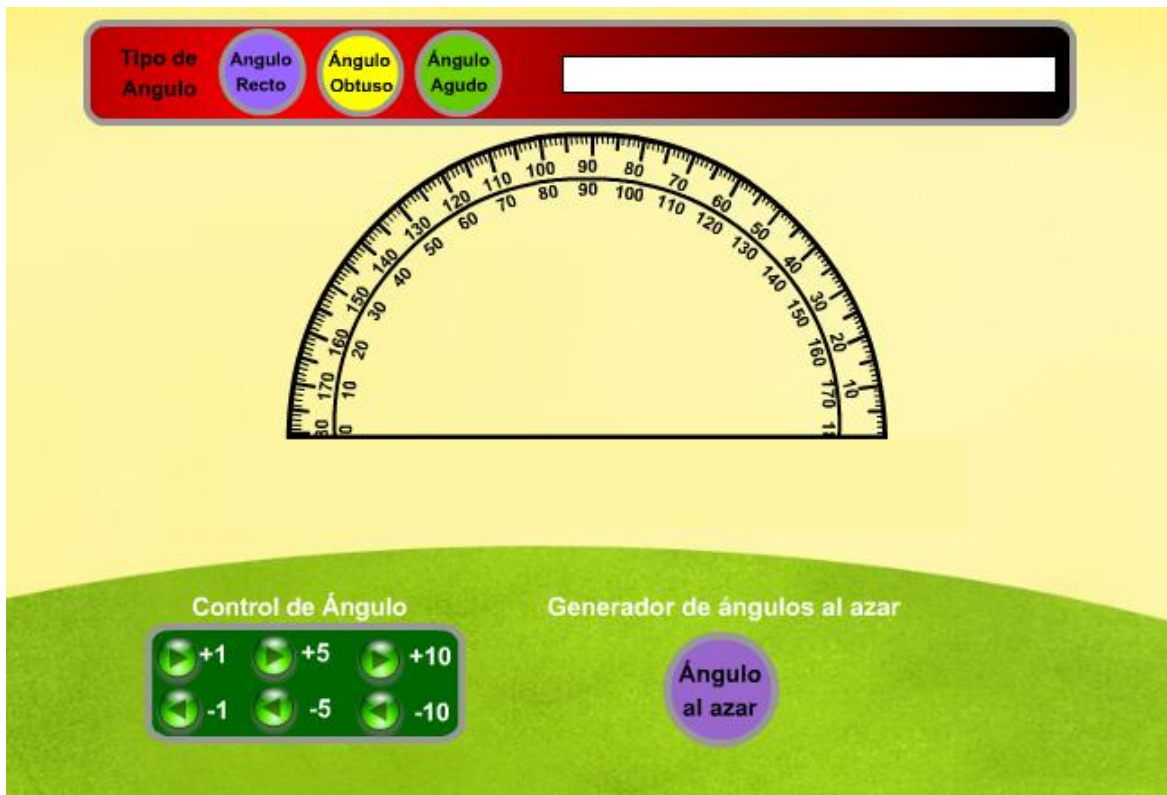
Se hace clic en el segundo capítulo de esta sección, **Cambio de Dirección y Giros**, y se obtiene el menú de este capítulo donde se encuentran los siguientes Botones: Concepto y Ejercicio.

### H.2.1. Concepto:

Aquí, a través de un gallo que se mueve de acuerdo a la dirección del viento, se muestra las direcciones Norte, Sur, Este y Oeste. Además se muestran los ángulos equivalentes a los giros del gallo para las diferentes direcciones (Norte-Sur, Sur-Norte, Este-Oeste, Oeste-Este). Finalmente con un graduador se muestra el concepto de ángulos rectos, agudos y obtusos.

### H.2.2. Ejercicio:

Este ejercicio permite que el estudiante pueda formar ángulos usando botones (+ 1, -1, +5, - 5, +10, -10) y aplastando los botones en el graduador, que son aquellos números múltiplos de 10. Se inicia con un ángulo 0. Finalmente, el ejercicio también permite que el estudiante determine que tipo de ángulo es el que se ha formado y un mensaje aparece de acuerdo al botón aplastado.



Anexo B.10. *Ejercicio Cambio de Dirección y Giros*

### **H.3. Ingresar a Capítulo Cálculo de Área**

#### **H.3.1. Concepto:**

#### **H.3.2.**

Se demuestra con ejemplos como se crea un cuadrado, un rectángulo y un triángulo y sus respectivas Áreas.

#### **H.3.3. Concepto:**

#### **H.3.4. Ejercicio:**

Se despliegan valores al azar para los lados del cuadrado, la base y altura del rectángulo y triángulo para que el estudiante encuentre las áreas correspondientes.

## **I. SECCIÓN ESTADISTICA**

### **I.1. Ingresar a Capítulo Estadística (Diagramad e Barras)**

#### **I.1.1. Concepto:**

Aquí, a través de una conversación entre los personajes, se explica para qué es importante usar los diagramas de barras. Enseguida se pasa a una tabla donde se pide ingresar ciertos valores para que sea diseñado automáticamente el diagrama de barras correspondiente. El usuario tiene la opción de volver a ingresar los valores en la tabla las veces que quiera y hacer que el sistema dibuje nuevamente el diagrama de barras correspondiente.

#### **I.1.2. Ejercicio:**

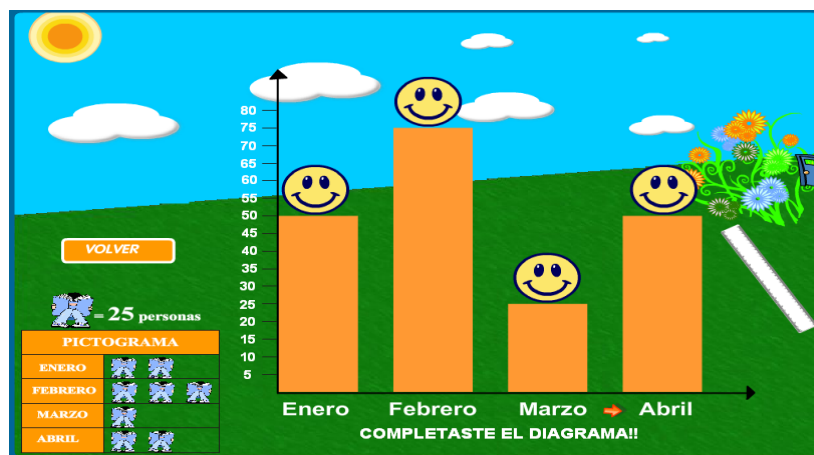
Este ejercicio se basa en el número de visitas que se ha efectuado a una biblioteca durante el mes de Enero, Febrero, Marzo y Abril. Esto se demuestra en un pictograma donde cada símbolo (un niño leyendo un libro) equivale a 25 personas. El usuario después de analizar el pictograma tiene la opción de dibujar el diagrama de barras correspondiente al hacer clic en los valores desplegados en la recta vertical. Esto se ejecuta de una



forma ordenada, es decir, que la barra correspondiente a cada mes es dibujada una a su vez. No se puede pasar a la barra del siguiente mes si la barra del mes anterior no ha sido dibujada correctamente. Los valores en el pictograma pueden ser modificados a través de un archivo xml, pero primero las modificaciones deben ser guardadas y el pictograma cargado nuevamente.



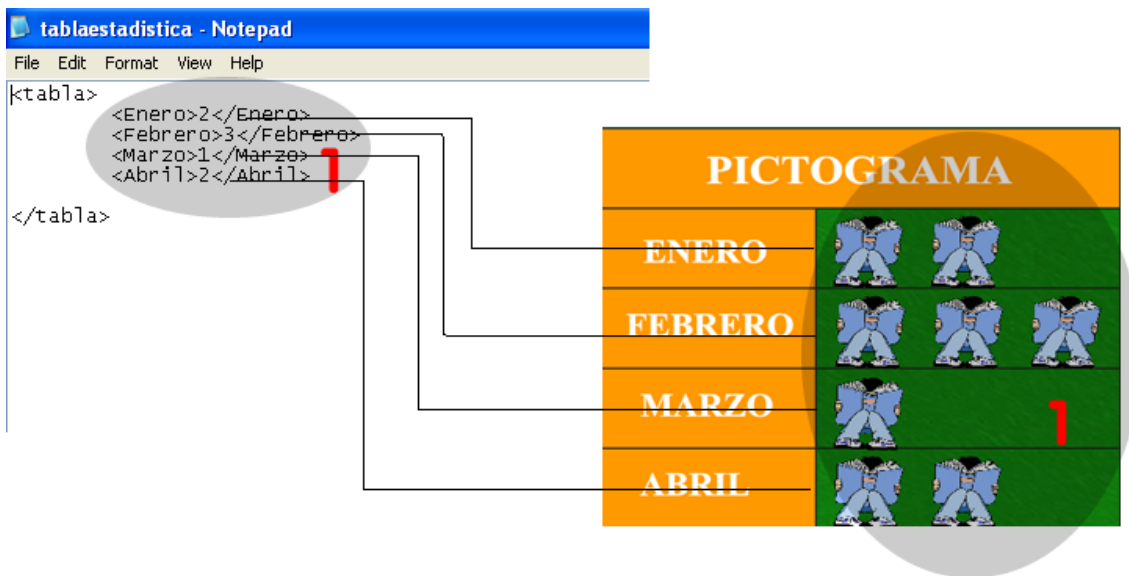
Anexo B.11. Ejercicio Estadística: Pictograma



Anexo B. 12. Ejercicio Estadística: Diagrama de Barras

### I.1.3. Modificar el archivo xml :

Para modificar el archivo xml, se ingresa a la carpeta donde se guardó el Sistema (generalmente esta carpeta está en Archivos de Programa/Sistema Multimedia), y se accede al archivo **tablaestadistica.xml** (por ejemplo, haciendo clic derecho en el archivo y escogiendo Notepad para editarlo).

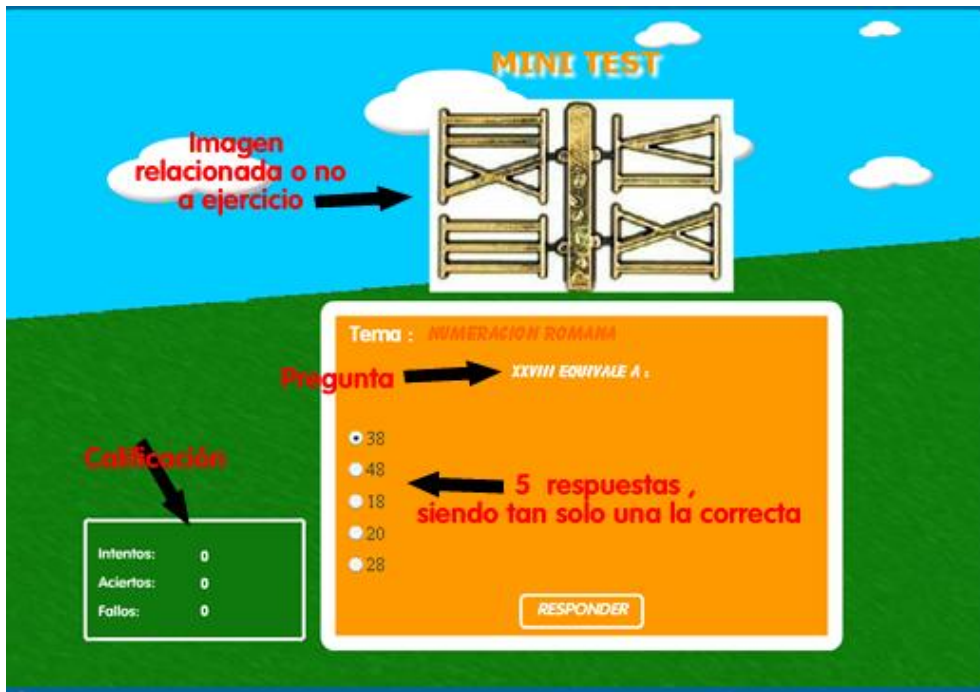


Anexo B. 13. Modificando tablaestadistica.xml

El Gráfico 17 indica como los valores del archivo xml son representados en el pictograma. Por ejemplo, hay dos símbolos de niños estudiando para el mes de Enero, representando 50 visitas. Esto corresponde a `<Enero>2</Enero>` en el archivo xml. Hay que tener cuidado para que los valores ingresados sean coherentes. Por ejemplo, los valores en la línea vertical del diagrama de barras van desde 0 a 80 (Ver Gráfico 16). Valores mayores o menores a estos crearían conflictos. Es muy importante que únicamente se modifiquen los *números* en este archivo xml.

## J. MINITEST

El minitest permite evaluar los conocimientos aprendidos a través de preguntas con 5 respuestas de elección múltiple. La estructura para cada preguntas es la siguiente:



Anexo B.14. Estructura del Minitest

### J.1. Responder Preguntas

Las preguntas son desplegadas una de cada vez, es decir, que no se puede pasar a la siguiente pregunta sin antes haber contestado correctamente la anterior. Para contestar, las respuestas se encuentran en 5 radio botones, de las cuales, tan solo una es la correcta. Se debe escoger la respuesta y hacer clic en el botón responder. Si la pregunta es correcta se despliega una cara feliz con su respectivo sonido y se prosigue a la siguiente pregunta. Si la respuesta es incorrecta, se despliega una cara triste con su sonido respectivo y **no** se prosigue a la siguiente pregunta. Cada vez que se presiona el botón *Responder*, el número de

intentos se va sumando uno y se suma el número de aciertos o fallos según corresponda.

## J.2. Finalizar el Minitest

Cuando se finaliza el Minitest se tiene la opción de cambiar el archivo xml y volver a cargar otro minitest. No se recomienda cambiar el xml antes de llegar al final del minitest porque pueden aparecer conflictos. Al cargar un nuevo minitest, la calificación (intentos, aciertos y fallos) se ponen en cero nuevamente.



Anexo B.20. Fin de Ministest

## J.3. Modificar el archivo xml

Para modificar el archivo xml, se ingresa a la carpeta donde se guardó el Sistema (generalmente esta carpeta está en Archivos de Programa/Sistema Multimedia), y se accede al archivo ***minitest.xml*** (por ejemplo, haciendo clic derecho en el archivo y escogiendo Notepad para editarlo).

```

minitest - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="iso-8859-1"?>
<MINITEST>

  <problema tema="Numeracion Romana">
    <pregunta>>XVIII equivale a :</pregunta>
    <respuesta1>38</respuesta1>
    <respuesta2>48</respuesta2>
    <respuesta3>18</respuesta3>
    <respuesta4>20</respuesta4>
    <respuesta5>28</respuesta5>
    <correcta>5</correcta>
    <image>Minitestimg/roman1.jpg</image>
  </problema>
  PREGUNTA 1

  <problema tema="Sistema Numerico Decimal">
    <pregunta>"cuantas decenas tienen 40000 unidades?"</pregunta>
    <respuesta1>30000</respuesta1>
    <respuesta2>4000</respuesta2>
    <respuesta3>400</respuesta3>
    <respuesta4>40</respuesta4>
    <respuesta5>4</respuesta5>
    <correcta>2</correcta>
    <image>Minitestimg/decimal1.gif</image>
  </problema>
  PREGUNTA 2

</MINITEST>

```

### Anexo B. 15. minitest.xml: estructura

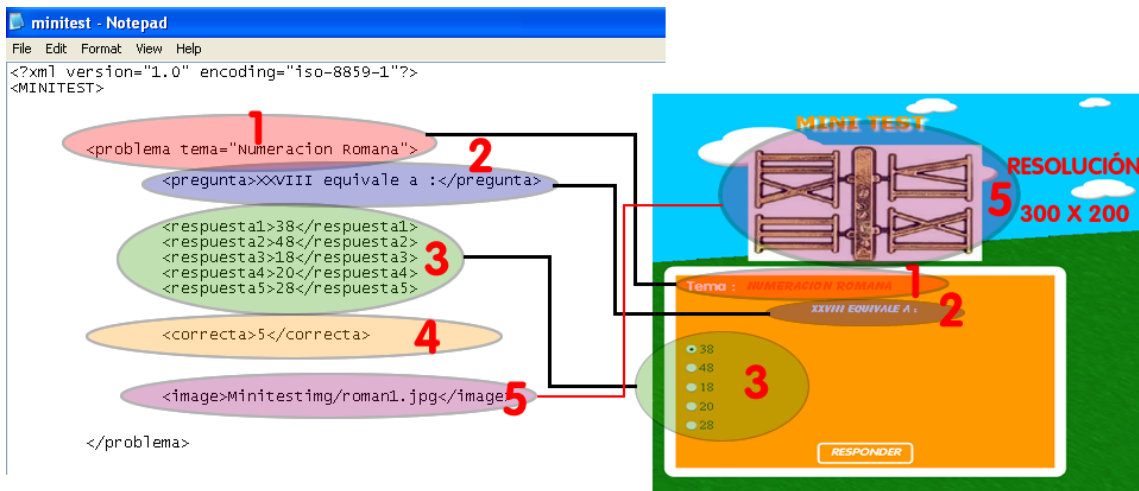
El Gráfico 20 muestra que cara una de esas secciones representa una pregunta, si se quiere agregar más preguntas se copia y pega en el lugar que se desee (por ejemplo, acabo del circulo correspondiente a la pregunta 2) una sección (por ejemplo la Pregunta 1) y se modifican sus contenidos. Por ejemplo, se podría agregar una tercera pregunta:

```

<problema tema="SUMA Y RESTA">
  <pregunta>" 549 + 0 = 549, qué sucede cuando uno de los sumandos es cero (0)?"</pregunta>
  <respuesta1>Nada</respuesta1>
  <respuesta2>No existe la suma</respuesta2>
  <respuesta3>Se resta cero</respuesta3>
  <respuesta4>El resultado es el otro sumando</respuesta4>
  <respuesta5>Ninguna de las anteriores</respuesta5>
  <correcta>4</correcta>
  <image>Minitestimg/suma y resta.gif</image>
</problema>

```

A seguir se analiza la estructura de cada pregunta:



Anexo B. 16. minitest.xml : estructura de la pregunta

Como se observa, cada sección del xml es reflejada en el minitest. Por ejemplo : El Tema: Numeración Romanos en el gráfico del minitest corresponde a <problema tema="Numeración Romana">. La imagen a ser desplegada en la pregunta debe estar guardada en la carpeta **Minitesting** y debe tener una resolución **300 x 200**. Si no se tiene esa resolución, la imagen se desplegará fuera de puesto, dañando como se ve el Minitest o causando conflictos. Por otro lado, el circulo 4 muestra cual de las respuestas es la correcta, en este caso <correcta>5</correcta> indica que la **quinta** pregunta es la correcta. Se debe considerar que solo hay preguntas de 1 al 5. Si se despliega valores menores o mayores a estos, habrá conflictos y nunca se podrá proseguir a la siguiente pregunta.

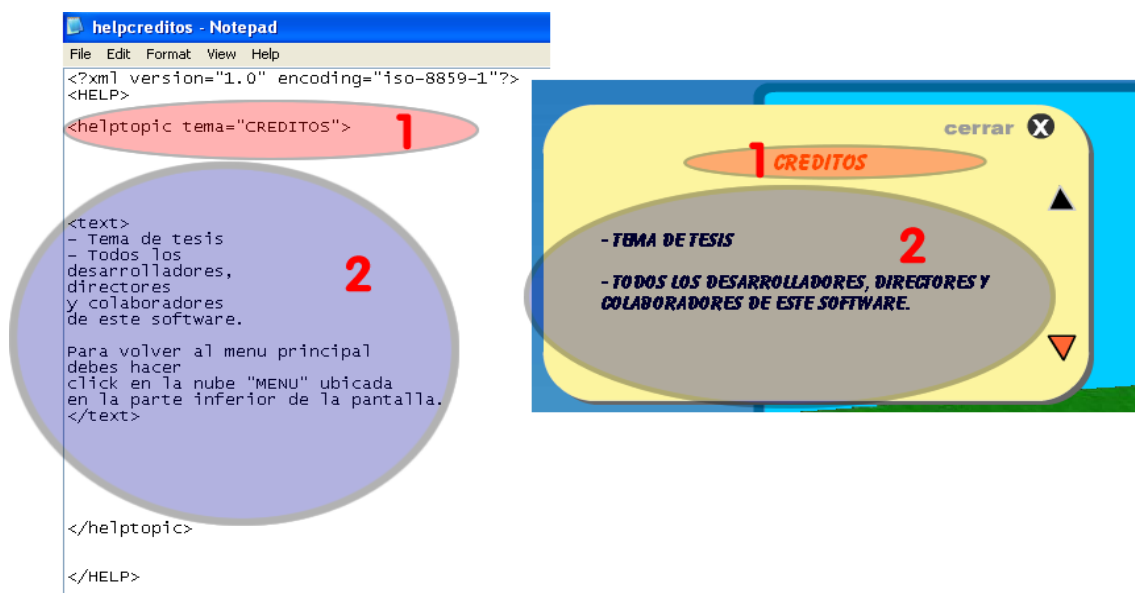
## K. AYUDA

El sistema ofrece ayuda en cualquier sección en la que se encuentre el usuario una vez que este ha ingresado al Menú Principal. La ayuda dará información acerca de la sección que se encuentre el usuario.

La ayuda aparece en una ventana extra adentro del sistema y puede ser desplazada a cualquier lugar que el usuario desee.

## K.1. Modificar los archivos xmls

El contenido de las ayudas están guardadas en varios archivos xmls de igual estructura (ver Gráfico 1). Para modificar los archivo xmls, se ingresa a la carpeta donde se guardó el Sistema y enseguida a la carpeta help (generalmente esta carpeta está en Archivos de Programa/Sistema Multimedia/help).Acceder a un archivo xml en esta carpeta. (por ejemplo, haciendo clic derecho en el archivo y escogiendo Notepad para editarlo). A seguir se muestra cuál es la estructura.



Anexo B. 17. helpcreditos.xml : estructura

Como el Gráfico 21 demuestra, `<helptopic tema="CREDITOS">` representa el tema de la ayuda, que en este caso es "Creditos". Mientras el texto entre `<text>` y `</text>` (circulo 2), representa el cuerpo de la ayuda. La ventana de ayuda presenta una barra de desplazamiento (scroll bar) para poder visualizar el resto del texto que se muestra en el xml.

Hay 15 archivos xml de ayuda, cada nombre representa la sección a que corresponde. Por ejemplo, helpcreditos.xml es ayuda acerca de la sección créditos, helpdivisionexacta.xml acerca de la sección división exacta, etc.

## **L. PROBLEMAS**

Si se siguen cuidadosamente las reglas de este Manual, el sistema no tiene que presentar ningún problema. En caso de problemas al cargar el xml, verificar que el nombre del archivo esté de acuerdo a los de LA Figura 1. Como última opción substituir todos los xml por los archivos iniciales xml que se encuentra en la carpeta respaldo.

Sin embargo, si se necesita información adicional o ha ocurrido un problema que no se puede solucionar con este Manual, por favor escribir a [ruthy\\_gg@hotmail.com](mailto:ruthy_gg@hotmail.com) con la pregunta específica y tema "PROBLEMAS CON SISTEMA MULTIMEDIA".



## BIOGRAFÍA

Ruth García Gavilanes nació en Quito el 16 de Enero de 1982. Su educación pre-escolar fue en el Jardín de Infantes ABC y su educación primaria fue en el Colegio Ecuatoriano Suizo de la ciudad de Quito. El final de sexto grado hasta el final de cuarto curso lo realizó en Brasilia-Brasil en el *Colegio Nossa Senhora do Perpetuo Socorro* (sexto grado a segundo curso) y *Colegio Educacional do Lago* (tercero y cuarto curso). Fue en Brasilia también donde ella aprendió más sobre la teoría de la música y piano en la “*Escola de Música de Brasilia.*” Además tuvo la oportunidad de practicar Gimnasia Olímpica e Inglés en otros institutos de esa ciudad.

De regreso al Ecuador en Noviembre de 1997, ella entró al Colegio Americano de Quito con un Inglés muy básico pero pudo culminar sus estudios con notas sobresalientes y con una media beca para estudiar Ciencias en la Computación en el College of Mount Saint Vincent en New York-USA. En esta ciudad ella pudo consolidar sus conocimientos del Inglés y hacer una pasantía en la ONU. Como pasatiempos en su tiempo libre, ella se interesó en aprender la salsa, que es muy conocida en New York, y hasta la fecha lo sigue practicando.

Sin embargo, a pesar de sus notas excelentes en el extranjero, Ruth decidió regresar al Ecuador en el 2000 para quedarse con su familia y continuar sus estudios universitarios. Fue entonces que ella entró a la ESPE a seguir la carrera de Ingeniería en Sistemas y en paralelo estudiar Francés en la Alianza Francesa de Quito, la misma que terminó en el 2005. Ruth tuvo algunas dificultades para adaptarse al sistema de esta universidad del Ejército, ya que era muy diferente al que estaba acostumbrada antes. En el 2004 decidió ir a los Estados Unidos por seis meses en búsqueda de nuevos horizontes pero regresó a los seis meses para concluir su carrera, que la terminó en Junio del 2007.

**HOJA DE LEGALIZACIÓN DE FIRMAS**

**ELABORADA POR**

---

Ruth García Gavilanes

**COORDINADOR DE LA CARRERA**

---

Ing. Ramiro Delgado

Lugar y fecha: \_\_\_\_\_