



INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación en Computación

"SISTEMA AUXILIAR PARA LA ELABORACIÓN  
DE MATERIALES DIDÁCTICOS PARA LA  
ESTIMULACIÓN DEL RECONOCIMIENTO  
VISUAL EN NIÑOS USANDO EL PARADIGMA  
DE EDUCACIÓN BASADO EN WEB"

T E S I S

QUE PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

JEANETT FIGUEROA MARTÍNEZ

DIRECTORES DE TESIS:

M. EN C. SERGIO SANDOVAL REYES.  
M. EN C. RUBÉN PEREDO VALDERRAMA



México D.F.

Junio 2009.

## RESUMEN

El proceso de alfabetización requiere de un trabajo profundo en la enseñanza de la lecto-comprensión. Este proceso demanda el desarrollo de habilidades de parte de los docentes y un trabajo esmerado, ya que el niño no sólo comienza el proceso de aprendizaje de la lecto-escritura, como una herramienta fundamental para la adquisición de conocimientos, transmisión de ideas y sentimientos, sino también por la situación en la que se halla inmerso al inicio de su escolarización. Este proceso de adquisición del saber es fundamental, partiendo de las primeras letras, números e incluso los primeros textos; motivo por el cual la tesis se centra en la necesidad de desarrollar una herramienta de software para auxiliar a los educadores en la realización de materiales educativos que estimule el reconocimiento visual en los niños y los ayude a madurar dentro del proceso de aprendizaje de la lectura.

Los libros, los juegos de bloques de construcción, los materiales de desecho, los juegos de pensar, y el cuerpo mismo, conforman ambientes de aprendizaje lúdicos, creativos, colaborativos e interactivos que favorecen a los niños, dentro de los contextos que son significativos para ellos, donde pueden vivir experiencias interesantes, excitantes, divertidas, sorprendentes, graciosas, curiosas, algunas permiten vivir micromundos entretenidos y amigables, fantasiosos con el fin de lograr una meta, esto le permite al niño desarrollar la creatividad, las reglas, dar un orden a sus ideas, que se auto critique en su desempeño, generando cambios de mentalidad y mejorando los procesos que están bajo su responsabilidad de ejecución, por este motivo los materiales que elaborará el terapeuta dentro de este sistema serán orientados a juegos.

La propuesta de esta tesis se apoya en el paradigma de Educación Basada en Web (*Web Based Education*, WBE por sus siglas en inglés), como un paradigma de enseñanza / aprendizaje cuyo objetivo es hacer posible la entrega de contenido educativo por medio de la computadora hacia el estudiante a través de la Web mediante un navegador que funciona como cliente para el estudiante, utilizando la arquitectura Cliente / Servidor, el protocolo HTTP, y un servidor que atiende las peticiones de los clientes.

Esta tesis esta enfocada hacia la construcción de una herramienta de Ingeniería de Software Asistida por Computadora (*Computed Aided Software Engineering*, CASE por sus siglas en inglés). La elaboración se basará en la arquitectura Cliente / Servidor y haciendo uso de componentes multimedia orientados a objetos que permitan, a los profesores o instructores, desarrollar materiales educativos basados en componentes y proporcionar este material a sus alumnos usando el paradigma de Educación Basada en Web.

## ABSTRACT

The process of literacy needs from a deep work in the lecto-comprehension teaching. The development of skills is demanded by this process on behalf of the teachers and a careful work, since the child not only begins the learning process of the reading and writing, like a fundamental tool for the knowledge acquisition, ideas and feelings transmission, but also for the situation in which he/she is immersed to the beginning of his schooling. This knowing acquisition process is fundamental, departing from the first lettering, numbers and even the first texts; these motives are the reason for which the thesis is centered on the need to develop a software tool to help the educators in the achievement of educational materials that stimulates the visual recognition in the children and help them to mature inside the reading learning process.

The books, construction blocks games, recyclable materials, games to think, and the body itself, form playful , creatives, collaboratives and interactives learning environments that favor to the children, inside the contexts which are significant for them, where they can live through interesting, exciting, entertaining, surprising, graceful and curious experiences. Where someone allows living through entertaining and friendly micro worlds, which they inspire the fantasy in order to achieve a goal, allowing the child to develop the creativity, the rules, give an order to his ideas, autocriticizing his performance, doing changes of mentality and improving the processes that are under his responsibility of execution. For these motives the materials that the therapist will prepare inside this system will be oriented to games.

The proposal of this thesis relies on the paradigm of Web-Based Education (WBE), as an education/learning paradigm which aim is making possible the delivery of educational content by the computer towards the student through of a Web browser which works as client for the student, using the Client/Server architecture, HTTP protocol, and a server which attends to the clients requests.

This thesis has an approach towards the construction of a *Computed Aided Software Engineering* (CASE) tool, under the Client/Server architecture and using use of object oriented multimedia components that allow to the teachers develop educational materials based on components and provide this material to his students using the WBE paradigm.

## GLOSARIO

**ActiveX:** grupo de tecnologías para Internet / intranet desarrolladas por Microsoft a finales de la década del 90. Algunas de estas tecnologías siguen siendo muy usadas, y otras han pasado a la historia. Por lo general, cuando se habla de un "ActiveX", se hace referencia directa a la tecnología ActiveX Control, que se utiliza en páginas *Web* [40].

**Actor:** símbolo del diagrama de caso de uso del UML que indica el rol que un usuario juega con respecto al sistema, es importante destacar que la palabra rol ya que no necesariamente un actor es representado por una persona, si no más bien la labor que se realiza frente al sistema [37], su representación gráfica es:



**Aferencias:** las aferencias sensoriales son las neuronas encargadas de la recepción de sensaciones para transmitir las al cerebro así como las neuronas eferentes son las que se encargan de conducir la información del cerebro a los músculos implicados [45].

**Agregación:** símbolo utilizado en el diagrama de clases del UML que representa la integración de varias clases en una sola. El símbolo de agregación es un diamante colocado en el extremo de la asociación, como se muestra en la Figura 3.10 [37].

**Apache:** es un servidor http, es un software de código abierto (libre) para plataformas Unix (GNU/Linux, etc), Windows, Macintosh, que implementa el protocolo HTTP [73].

**Apache Tomcat:** software de código abierto (libre) para plataformas Unix (GNU/Linux, etc), Windows, Macintosh; es un servidor *Web* que implementa Java *servlets* y Java JSPs [79].

**Arquitectura Cliente / Servidor:** esta arquitectura define servicios especializados con funciones específicas. Se denomina proceso cliente al que solicita un servicio y un proceso servidor al que es capaz de proporcionar el servicio. Los clientes proporcionan al usuario las interfaces adecuadas para utilizar estos servidores y llevar a cabo el procesamiento local. Los servidores pueden proporcionar a los clientes servicios tales como impresión, acceso a archivos, o acceso a Bases de Datos [34].

**Armar:** concertar y juntar entre sí las varias piezas de que se compone un mueble, un artefacto, etc., o sentar, fundar algo sobre otra cosa [75].

**Asociación:** símbolo de los diagramas UML, representa el tipo de relación más básica para mostrar la invocación desde un actor o de un caso de uso a otro caso de uso; dicha operación se denota con la flecha simple [37], su representación gráfica es:



**Caso de uso.** símbolo del diagrama de caso de uso del UML que indica una operación o tarea específica que se realiza tras la orden de un actor o desde la invocación de otro caso de uso [37], su representación gráfica es:



**CBT o CBI:** son las siglas en inglés de *Computer-Based Training o Instruction*, Capacitación por medio de la computadora; Consiste en un programa de computadora que presenta información utilizando recursos multimedia. Se refiere a cualquier tipo de entrenamiento cuyo principal mecanismo de apoyo es la computadora, sus objetivos son: mejorar el aprendizaje de nuevos conceptos, desarrollar habilidades y destrezas para lograr un mejor desempeño. En este contexto también se asocia el término CBT a la palabra *courseware* [46].

**Cinestésico:** el sentido cinestésico es conocido también con el nombre de sentido muscular, su función consiste en coordinar de manera automática músculos, tendones y huesos. Siempre que realizamos cualquier movimiento, gracias a este sentido sabemos los movimientos que realizamos y la posición de miembros y cuerpo [47].

**CCM:** son las siglas en inglés de CORBA *Component Model* es un componente básico en CORBA donde los tipos son especificados en un lenguaje definido de interfaces y puesto en repositorios. Un componente es utilizado por la referencia desde otro objeto. Un componente definido es una especificación y extensión de una definición de interfase [44].

**Clase:** es un molde que comparte una estructura y una conducta para la creación de objetos; el diagrama de una clase en el UML se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones, un ejemplo se puede observar en la Figura 3.12.

**CMS:** son las siglas en inglés de *Content Management System*. Sistema Manejador de Contenido es un ambiente virtual de aprendizaje que permite separar el contenido del diseño para crear cursos en línea, no posee herramientas elaboradas para el trabajo colaborativo (foros, *chats*, etc) [48].

**COM:** son las siglas en inglés de *Component Object Model*. Es una plataforma de Microsoft para componentes de software introducida en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología [40].

**Coordinación óculo manual:** es la coordinación mano-ojo, sirve para determinar la lateralidad en el niño [49].

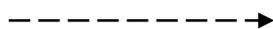
**Courseware:** software de cómputo para el uso en la enseñanza. Se refiere a cualquier material instruccional, a través de la computadora, para apoyar el proceso de aprendizaje [70].

**CORBA:** son las siglas en inglés de *Common Object Request Broker Architecture*. Es un estándar definido por el Grupo Director de Objetos (*Object Management Group*, OMG por sus siglas en inglés) que permite escribir componentes de software en múltiples lenguajes y ejecutarlos en múltiples computadoras para trabajar juntos, en otras palabras de forma distribuida [43].

**DCOM:** son las siglas en inglés de *Distributed Component Object Model* en español Modelo de Objetos de Componentes Distribuidos, la cual es una tecnología de Microsoft que permite desarrollar componentes de software distribuidos sobre múltiples computadoras que se intercomunican [43].

**DDE:** son las siglas en inglés de *Dynamic Data Exchange*, el cual es el protocolo de intercambio de datos de Microsoft para aplicaciones de Windows [36].

**Dependencia o Instanciación:** símbolo del UML que indica la relación de un caso de uso A hacia un caso de uso B, indicando que el caso de uso B implementa la funcionalidad del caso de uso A se denota con una flecha punteada [37] como se ve en la siguiente figura



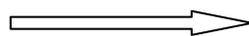
**Elaborar:** transformar una cosa u obtener un producto por medio de un trabajo adecuado [75].

***e-Learning***: es un nuevo concepto de educación a distancia en el que se integra el uso de las tecnologías de la información y otros elementos didácticos para la capacitación y enseñanza. El *e-learning* utiliza herramientas y medios diversos como Internet, intranets, CD-ROM, presentaciones multimedia, etc. [71].

**Esolanovismo o cognoscitivismo**: el cognoscitivismo es el proceso independiente de decodificación de significados que conduce a la adquisición de conocimientos a largo plazo y al desarrollo de estrategias que permitan la libertad de pensamiento, la investigación y el aprendizaje continuo en cada individuo, lo cual da un valor real a cualquier cosa que se desee aprender. En el paradigma constructivista, el alumno es quien aprende involucrándose con otros alumnos durante el proceso de construcción del conocimiento (construcción social), tomando la retroalimentación como un factor fundamental en la adquisición final de contenidos [50].

***Framework***: plataforma, entorno, marco de trabajo; desde el punto de vista de desarrollo de software, un *framework* es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Los *frameworks* suelen incluir: soporte de programas, bibliotecas, lenguaje de scripting, software para desarrollar y unir diferentes componentes de un proyecto de desarrollo de programas. Los *frameworks* permiten facilitar el desarrollo de software, evitar los detalles de bajo nivel, permitiendo concentrar más esfuerzo y tiempo en identificar los requerimientos de software [42].

**Generalización**: símbolo del UML que indica si cumple, dependiendo de su estereotipo, la función de uso <<uses>> o de herencia <<Extends>> y solo se emplea para los casos de uso [37], su representación gráfica es la siguiente:



**GLC**: son las siglas en inglés de *Global Learning Consortium*. El Consorcio de Aprendizaje Global es un estándar internacional para los sistemas *e-Learning*, desarrollado en Estados Unidos. El objetivo del proyecto es la amplia adopción de especificaciones que permitirán que contenidos y entorno de aprendizaje distribuido de múltiples autores puedan trabajar juntos [51].

**Gnosia**: proceso de conocimiento a partir de las impresiones suministradas por los órganos sensoriales. Implica la percepción, reconocimiento y denominación de los estímulos provenientes de un objeto. Requiere la participación de los centros corticales superiores del cerebro [50].

**GUI**: son las siglas en inglés de *Graphical User Interface*, la interfaz gráfica de usuario y es el medio tecnológico de un sistema interactivo que posibilita, a

través del uso y la representación de un lenguaje visual una interacción amigable con un sistema informático [53].

**Herencia:** es un concepto de la POO que indica cuando una clase hereda las operaciones y atributos específicos a una subclase, además de tener sus propias operaciones y atributos [37].

**HTML:** son las siglas en inglés de *HyperText Markup Language*. Lenguaje de marcado de hipertexto o lenguaje de marcación (una forma de codificar el documento a través de etiquetas que al dar clic sobre estas llevan a otra información referente al texto). Es el lenguaje predominante para la construcción de las páginas *Web* [53].

**IEEE:** son las siglas en inglés de *Institute of Electrical and Electronics Engineers*. Instituto de ingenieros eléctricos y electrónicos, uno de sus objetivos es organizar y hacer normas estándares para las ingenierías [56].

**Lateralidad:** lo que hace a las personas tomar las cosas con una mano o con otra, con lo cual ejecuta ciertos movimientos, por ejemplo, la escritura. Dada la composición del cerebro este determina características de mando sobre cada hemisferio del cuerpo; nuestro cerebro maneja de manera cruzada la lateralidad, es decir que los actos realizados por el hemisferio derecho del cuerpo son procesados por el hemisferio izquierdo del cerebro. Claro está que esta preferencia está demarcada inicialmente por los determinantes genéticos y en segundo lugar por el medio y la cultura [57].

**LCMS:** son las siglas en inglés de *Learning Content Management System*. El Sistema Manejador de Contenido y Aprendizaje es un ambiente virtual de aprendizaje que incorpora la gestión de contenidos para personalizar los recursos a cada alumno, así como la implementación de cursos, materiales y contenidos en línea [54].

**LMS:** son las siglas en inglés de *Learning Management System*. El Sistema Manejador de Aprendizaje es un ambiente virtual de aprendizaje que se utiliza para la creación, gestión y distribución de cursos a través de la *Internet*. Sirve de contenedor de cursos e incorpora herramientas para facilitar el trabajo colaborativo [54].

**LOM:** son las siglas en inglés de *Learning-Object Metadata*. Especificación de Meta datos de Objetos de Aprendizaje. Primera acreditación estándar para la tecnología de aprendizaje [55].

**LTSC:** son las siglas en inglés de *Learning Technology Standards Comité*. Comité de Estándares de Tecnologías de Aprendizaje, es uno de los estándares de objetos de aprendizaje [55].

**Metodología:** conjunto de procedimientos, técnicas, herramientas y un soporte documental basado en una teoría del conocimiento que ayude a los desarrolladores a realizar nuevo software [58].

**Middleware:** es un software que conecta componentes de software o aplicaciones para que puedan intercambiar datos entre sí. Es utilizado a menudo para soportar aplicaciones distribuidas. Esto incluye Servidores Web, Servidores de aplicaciones, y herramientas similares. *Middleware* es especialmente esencial para tecnologías como XML, SOAP, servicios *Web* y arquitecturas orientadas a servicios [76].

**Multimedia:** es un término empleado para describir diversos tipos de medios que se utilizan para transportar información (texto, audio, gráficos, animación, video, e interactividad) [76].

**MVC:** son las siglas en inglés de *Model View Control* que indican el patrón Modelo Vista Controlador; Es un patrón de arquitectura para desarrollo de software, que separa los datos de la aplicación, la interfaz del usuario y la lógica de control [28].

**Navegador Web:** también llamado *browser* es un software de aplicación usado para localizar y desplegar páginas *Web*. Los navegadores más populares son Microsoft *Internet Explorer*, *Mozilla*, *Safari*, *Netscape Navigator* entre otros; son navegadores gráficos lo que significa que pueden desplegar gráficos y texto. Los navegadores más modernos pueden presentar información multimedia incluyendo sonido y video.

**Objetos:** es la unidad individual que en tiempo de ejecución realiza una tarea; es la instancia de una clase [78].

**OLE:** son las siglas en inglés de *Object Linking and Embedding*. Es una tecnología de Microsoft para compartir información entre distintas aplicaciones de Windows. Permite la creación de documentos o programas incorporando elementos de otros. Para la ejecución de dicho programa o documento deben existir los archivos necesarios para utilizarlos. Por ejemplo, si en un documento se utilizan gráficos Excel, para visualizarse debe estar instalado el programa Excel o un visualizador de Excel [38].

**Patrón:** pueden ser uno o más modelos a seguir [58].

**Percepción visoespacial:** funciones mentales implicadas en distinguir por medio de la vista, la posición relativa de los objetos en el entorno o en relación a uno mismo [59].

**Perceptual:** percepción de estímulos a través de los órganos de los sentidos. “La sensación es la aprehensión sensorial de un objeto, que contiene tres características a destacar: configuración, actualidad y carácter objetivo”. El producto de la percepción es la suma de las sensaciones [60].

**Praxia:** (de praxis o práctica). Es el proceso neurológico que nos permite organizar, planear y ejecutar, de una forma eficiente, habilidades de todos los tipos. Algunos componentes de la praxia ocurren involuntariamente, automáticamente o inconscientemente, mientras otros requieren de pensar. Los diferentes componentes de la praxia requieren una información precisa proveniente del cuerpo, particularmente de los sistemas táctiles, propioceptivos y los sentidos visuales y auditivos que completan, refinan y producen continuamente el proceso de la praxia [61].

**Procedimiento:** definición de la forma en que se ejecuta la tarea [58].

**Propioceptivo:** sistema propioceptivo es el que nos proporciona información sobre el funcionamiento armónico de músculos, tendones y articulaciones, participa regulando la dirección y rango de movimiento, permite reacciones y respuestas automáticas, importantes para la supervivencia, interviene en el desarrollo del esquema corporal y en la relación de éste, con el espacio, sustentando la acción motora planificada [62].

**Protocolos TCP / IP:** son las siglas en inglés de *Transmission Control Protocol / Internet Protocol*, es uno de los principales protocolos para la transmisión de datos entre dos dispositivos en redes. El protocolo IP especifica el formato de los paquetes o también llamado datagramas y la dirección del esquema. TCP permite a los *hosts* establecer una conexión e intercambiar datos. TCP garantiza la entrega de datos y también garantiza que los paquetes serán entregados en el mismo orden en el cual se envían [68].

**Psicométrico(a):** del sustantivo psicometría que se refiere al conjunto de los métodos de medición indicativa utilizados dentro del campo de la psicología [63].

**Streaming:** es un término que se refiere a ver u oír un archivo directamente en una página *Web* sin necesidad de bajarlo completamente a la computadora para reproducirlo. Por lo que es posible reproducirlo mientras se está bajando. En términos más complejos podría decirse que describe una estrategia sobre demanda para la distribución de contenido multimedia a través de la *Internet* [64].

**UML:** son las siglas en inglés de *Unified Modeling Language*. Lenguaje para el modelado, es un lenguaje para la especificación, visualización, construcción y documentación de los componentes de un sistema. UML es la notación

(principalmente gráfica) de que se valen los métodos para expresar los diseños orientados a objetos [58].

**Visomotora:** se refiere a la coordinación visual la cual permite ciertos movimientos ajustados por el control de la visión; tiene gran importancia en el aprendizaje de la escritura [67].

**VBX:** son las siglas en inglés de *Visual Basic Extension*. En programación, VBX era el modelo de componente usado en Microsoft Visual Basic entre las versiones 1.0 y 3.0. Siguió siendo soportado en la versión 16-bit del VB 4.0, pero se volvió obsoleto con la aparición del OCX [39].

**WBE:** son las siglas en inglés de *Web-Based Education*. La educación basada en *Web* es un nuevo paradigma de enseñanza cuyo objetivo es hacer posible la entrega de contenido educativo a la computadora personal del estudiante utilizando de la *Web* mediante un navegador que funciona como cliente para el estudiante a través de la arquitectura Cliente / Servidor, protocolo HTTP, y un servidor que atiende las peticiones de los clientes [69].

**WWW:** son las siglas en inglés de *World Wide Web*. La *Web* o WWW, es un sistema de hipertexto que funciona sobre la Internet. Para ver la información se utiliza una aplicación llamada navegador *Web* para extraer elementos de información (llamados “documentos” o “páginas *Web*”) de los servidores *Web* o “sitios” y presentar los resultados en la pantalla del usuario [65].

**XML:** son las siglas en inglés de *Extensible Markup Language*. Es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web* (W3C) [29].

# ÍNDICE

Resumen	i
Abstract	ii
Glosario	iii
Índice	xi
Lista de figuras	xiv
Lista de tablas	xix
 CAPÍTULO 1	
	1
Marco de referencia	
1.1 Introducción	2
1.2 Planteamiento del problema y justificación	3
1.3 Objetivos	3
1.3.1 Objetivo general	3
1.3.2 Objetivos específicos	3
1.4 Límites y alcances	4
1.5 Beneficios esperados	4
1.6 Organización de la tesis	5
 CAPÍTULO 2	
	6
Estado del arte	
2.1 Introducción	7
2.2 Software educativo - WIN-ABC	9
2.3 Software educativo - EL MONO COCO	11
2.4 Software educativo – JUGAR CON	12
2.5 Software educativo – ACTIVIDADES DE LECTO-ESCRITURA	13
2.6 Software educativo - LOS COLORES	14
2.7 Software educativo - EN SERIE	15
2.8 Software educativo - LOS TAMAÑOS	16
2.9 Software educativo – LAS LETRAS SUENAN	17
2.10 Software educativo - APRENDE LOS ANIMALES	19
2.11 Software educativo – FIGURAS	20
2.12 Software educativo - LOS PRIMEROS NÚMEROS POR VÍA VISUAL DIRECTA	22
2.13 Software educativo – PACO EL CHATO	23
2.14 Software educativo – DESBERDINTASUNAK	23
2.15 Software educativo - ADENTRO AFUERA	24
2.16 Software educativo - JUGANDO ACTIVO MI INTELIGENCIA	24
2.17 Software educativo - ¿CUÁL ES CUAL?	25
2.18 Software educativo - ANICLIC MÉTODO DE LECTURA	26

2.19 Software educativo - A GAIVOTA DE 'A COVIÑA'	27
2.20 Software educativo - JUGANDO APRENDO	28
2.21 Software – JCLIC AUTOR	29
2.23 Propuesta de solución	34
<b>CAPÍTULO 3</b>	<b>39</b>
<b>Análisis y diseño del sistema</b>	
3.1 Metodología para el desarrollo del sistema	40
3.2 Diagramas de UML	41
3.2.1 Diagrama de casos de uso	41
3.2.2 Diagramas de clases	48
3.2.2.1 Diagrama de la clase del subsistema Login	49
3.2.2.2 Diagrama de la clase de creación de archivos XML	50
3.2.2.3 Diagrama de la clase de subida de archivos	51
3.2.3 Diagramas de secuencia	53
3.2.4 Diagramas de actividades	54
3.3 Patrón de diseño Modelo-Vista-Controlador	55
3.4 Modelo Objeto Documento	57
3.5 Componentes de Software	59
3.5.1 Diseño de la interfaz de los componentes de software	61
<b>CAPÍTULO 4</b>	<b>65</b>
<b>Implementación del sistema</b>	
4.1 Descripción del funcionamiento del sistema AMDERV-WBE	66
4.1.1 Arquitectura Cliente / Servidor	66
4.1.2 Servlets	68
4.1.3 Java Server Page (JSP)	68
4.1.4 Java Bean	69
4.1.5 Struts	69
4.1.6 Modelos para el desarrollo de aplicaciones <i>Web</i>	70
4.2 Implementación del controlador	71
4.3 Implementación del sistema	72
4.3.1 Subsistema Login	72
4.3.1.1 Base de datos	76
4.3.2 Implementación de los componentes	77
4.3.3 Implementación del subsistema de subida de archivos	84
<b>CAPÍTULO 5</b>	<b>86</b>
<b>Pruebas y resultados del sistema</b>	
5.1 Pruebas	87
5.2 Resultados	90

CAPÍTULO 6	92
<b>Conclusiones sistema AMDRV-WBE.</b>	
6.1 Logros alcanzados	93
6.2 Trabajos a Futuro	94
Referencias bibliográficas y electrónicas	95
 ANEXO A	
Manual de usuario del sistema AMDERV-WBE	101
 ANEXO B	
Especificaciones de hardware y software	113
 ANEXO C	
Encuestas realizadas a profesores	115

## LISTA DE FIGURAS

FIGURA	DESCRIPCIÓN	PÁGINA
<b>CAPÍTULO 2</b>		
2.1	Presentación del software educativo: WIN-ABC	10
2.2	Pantalla del ejercicio rotación-inversión	11
2.3	Menú de selección de actividades del software educativo “El mono Coco”	12
2.4	Portada del Software educativo “Jugar con”	13
2.5	Portada del software “Actividades de lecto-escritura”	13
2.6	Uniendo sonido escuchado con la silaba correspondiente	14
2.7	Descripción del software educativo “LOS COLORES”	14
2.8	Actividad de elegir el color escuchado	15
2.9	Descripción del Software educativo “En serie”	15
2.10	Actividad de dar un clic en la figura que sigue	16
2.11	Descripción del Software educativo “LOS TAMAÑOS”	17
2.12	Actividad de dar un clic en la figura grande o pequeña	17
2.13	Descripción del Software educativo “Las letras que suenan”	18
2.14	Actividad de dar un clic en la letra que escuchó o vió en el patrón	18
2.15	Descripción del Software educativo “Aprende los animales”	19
2.16	Actividad de reconocimiento del animal	19
2.17	Actividad de rompecabezas	19
2.18	Actividad de memorama	20
2.19	Actividad de señala al animal	20
2.20	Actividad de busca la pareja	20
2.21	Descripción del Software educativo “Figuras”	20
2.22	Reconocimiento de los colores	21
2.23	Reconocimiento de las figuras	21
2.24	Descripción del Software educativo “Los primeros números por vía visual directa”	22
2.25	Actividad uniendo las dos cantarinas con el dos indicado con los dedos	22
2.26	Descripción del Software educativo “Paco el Chato”	23
2.27	Descripción del Software educativo “Desberdintasunak”	23
2.28	Descripción del Software educativo “ADENTRO AFUERA”	24
2.29	Descripción del Software educativo “Jugando activo mi inteligencia”	25
2.30	Descripción del Software educativo ¿Cuál es cual?	25

2.31	Descripción del Software educativo “ANICLIC MÉTODO DE LECTURA”	26
2.32	Actividad que une la imagen con la letra que comienza	26
2.33	Actividad donde se escribe el patrón visto y escuchado	27
2.34	Descripción del Software educativo “A GAIVOTA DE ‘A COVIÑA”	27
2.35	Actividad donde hay que seleccionar, la respuesta a lo que se pregunta	28
2.36	Descripción del Software educativo “Jugando aprendo”	28
2.37	Actividades: Busca las diferencia, orientación izquierda-derecha, memorama y rompecabezas	29
2.38	Pantalla para desarrollo de actividades	30
2.39	Vista de los Puzzles o rompecabezas	30
2.40	Vista de una pantalla con la actividad de Asociación	31
2.41	Vista de una pantalla con la actividad sopa de letras	31
2.42	Vista de una pantalla con la actividad sopa de crucigrama	32
2.43	Vista de una pantalla con la actividad de texto: respuesta escrita a casillas	32
2.44	instalación del paquete jugando activo mi inteligencia en la computadora	33
2.45	Programa “jugando activo mi inteligencia” al empezar el programa	33

### CAPÍTULO 3

3.1	Diagrama de caso de usos para el acceso al profesor	41
3.2	Diagrama de caso de uso para el acceso al alumno	42
3.3	Diagrama de caso de uso para el acceso al administrador	42
3.4	Diagrama de caso de usos de el ambiente de trabajo cuando el profesor ingresa al sistema.	42
3.5	Diagrama de casos de uso de el ambiente de trabajo cuando el alumno ingresa al sistema.	43
3.6	Diagrama de casos de uso de el ambiente de trabajo cuando el administrador ingresa al sistema.	43
3.7	Diagrama de clases que modela el sistema	46
3.8	Representación de un objeto de la clase Administrador	46
3.9	Representación de una asociación directa Usuario que accede al Portal	47
3.10	Representación de una agregación en la clase GUI multimedia	48
3.11	Representación de una súper clase Usuario que hereda a Alumno, Profesor y Administrador	48
3.12	Clase LoginForm	49
3.13	Clase LoginBean	49

3.14	Clase XmlActionForm	50
3.15	Clase XmlFile	51
3.16	Clase StrutsUploadForm	52
3.17	Clase del StrutsUploadAction	52
3.18	Diagrama de Secuencias del sistema	53
3.19	Diagrama de Actividades del sistema	54
3.20	Modelo-Vista-Controlador	56
3.21	Arquitectura JavaBeans-JSPs-Servlets= <i>Model-View-Controller</i>	56
3.22	Fragmento de un código HTML	58
3.23	Representación del DOM para el código HTML	58
3.24	Modelo general de los componentes de software y archivos XML que utilizan.	61
3.25	Diseño de la interfaz del componente memorama	61
3.26	Diseño de la interfaz de obtención de datos para el componente memorama	62
3.27	Diseño de la interfaz del componente relación de columna	62
3.28	Diseño de la interfaz de obtención de datos para el componente relación de columnas	63
3.29	Diseño de la interfaz del componente relación de imagen con sonido	63
3.30	Diseño de la interfaz de obtención de datos para el componente relación de imagen con sonido	64

#### CAPÍTULO 4

4.1	Arquitectura Cliente / Servidor	67
4.2	Ciclo de vida del servlet	69
4.3	Diagrama de la arquitectura Java modelo 1	70
4.4	Diagrama de la arquitectura java modelo 2	71
4.5	Arquitectura del sistema AMDERV-WBE	71
4.6	Implementación del patrón MVC con <i>struts</i>	72
4.7	Vista de la página entrada.jsp	73
4.8	Diagrama de clase del ActionForm entradaActionForm	73
4.9	Página de salida cuando el cliente no existe en la base de datos	74
4.10	Página de salida cuando el cliente es un Alumno	74
4.11	Página de salida cuando el cliente es un Profesor	75
4.12	Página de salida cuando el cliente es un Administrador	75
4.13	Diagrama del Subsistema de verificación del cliente con <i>struts</i>	76
4.14	Atributos de la tabla usuario	77

4.15	Código en <i>ActionScript</i> que carga la información de un XML al componente	78
4.16	Componente de relación de columnas	79
4.17	Componente de memorama	80
4.18	Componente de relación de imagen con sonido	80
4.19	Código Fuente que utiliza JAXP para generar el archivo slides.xml para el componente memorama	81
4.20	Árbol de objetos DOM para el archivo Slides.XML para el componente memorama	82
4.21	Archivo slides.XML resultado del árbol DOM	82
4.22	Diagrama de <i>Strutt-config.xml</i> del Sistema que genera los archivos XML de los distintos tipos de componentes	83
4.23	Diagrama de <i>strut</i> que genera el Archivo XML para el componente memorama	83
4.24	Diagrama de <i>Strut</i> que genera el Archivo XML para el componente relación columna	83
4.25	Diagrama de <i>strut</i> que genera el Archivo XML para el componente relación de sonido con imagen	84
4.26	Diagrama de <i>strut</i> que genera el Archivo XML que integra los componentes de memoramas o relación de columnas o relación de sonido con imagen	84
4.27	Modelo relacional del Sistema de base de datos utilizado para la autenticación y la creación del material didáctico	85

#### Anexo A

A1	Página de entrada para acceso al sistema	102
A2	Página de acciones para el administrador	103
A3	Página que muestra el formato de altas	104
A4	Página que muestra el formato de bajas de los usuarios	104
A5	Página que muestra el formato para modificar la información del usuario	105
A6	Página que muestra el menú de opciones para el usuario profesor	106
A7	Página que muestra la opción para introducir el nombre al material didáctico	106
A8	Menú para la creación de los diferentes materiales didácticos	107
A9	Muestra el formato para elegir las imágenes para el memorama	107
A10	Página que muestra la vista preliminar del memorama con las figura elegidas	108
A11	página que muestra el formato para llenar la pregunta, las imágenes, su descripción para el componente relación	109

	columna	
A12	Página que muestra la vista previa del componente relación columna	109
A13	Página que muestra el formato para llenar los datos del componente relación de sonido con imagen	110
A14	Página que muestra la vista previa del componente relación de sonido con imagen	111
A15	Página que despliega en forma de lista todos los materiales	112
A16	Página que despliega la ejecución de un material didáctico	112

## LISTA DE TABLAS

<b>TABLA</b>	<b>DESCRIPCIÓN</b>	<b>PÁGINA</b>
<b>CAPÍTULO 2</b>		
2.1	Comparación de software a fin y el Sistema a desarrollar	37
<b>CAPÍTULO 3</b>		
3.1	Descripción de los casos de uso del sistema	46
<b>CAPÍTULO 5</b>		
5.1	Pruebas de funcionamiento del sistema AMDERV-WBE	88

---

---

# CAPÍTULO 1

## MARCO DE REFERENCIA

---

---

En este capítulo se presenta una reseña del paradigma de Educación Basada en la *Web*, se señalan algunas de sus ventajas y se proporciona una idea de la orientación del sistema. A partir de ello se formulan los objetivos, límites, alcances y los beneficios esperados, así como la organización de la tesis.

## 1.1 Introducción

Las computadoras en los hogares, lugares de trabajo y las escuelas, en conjunto con las nuevas tecnologías de la información y las telecomunicaciones, se han transformado en herramientas indispensables, en mecanismos para obtener información de forma instantánea.

Hasta hace algunos años, el problema de la información se centraba en la posibilidad de acceder a la misma. Hoy, el problema de la información se ha desplazado hacia la capacidad de hacer uso oportuno, efectivo y eficiente de ella, pues está disponible en una inmensa fuente pública, cuyo tamaño está creciendo de manera vertiginosa todos los días, y no se puede procesar toda esta información de manera individual. La fuente a la que nos referimos es la *Web* (o *World Wide Web*, *WWW* por sus siglas en inglés), por lo que el nuevo contexto mundial requiere nuevos paradigmas en los entornos de enseñanza / aprendizaje. Uno de estos nuevos paradigmas es la Educación Basada en *Web* (*Web Based Education*, *WBE* por sus siglas en inglés). Convirtiéndose en un excelente complemento de casi todos los tipos de educación; el modelo básico es la entrega de materiales educativos vía *Web* del lado del cliente utilizando un navegador y como infraestructura la Internet para la comunicación y un servidor *Web* que permita utilizar el protocolo HTTP para atender en las peticiones del cliente proporcionando herramientas para la comunicación, con una capacidad enorme para aprovechar la mayor ventaja de la Internet / *Web*, que es la bidireccionalidad.

Así, comienza a ser necesario el uso de herramientas que aprovechen las ventajas de la Internet / *Web*:

- Desarrollar nuevos modelos que dirijan un aprendizaje significativo basado en redes de información, con nuevas secuencias en la ejecución de actividades.
- Las tecnologías multimedia, como un medio poderoso de comunicación en muchos ámbitos de nuestra sociedad. Aprovechando su capacidad para integrar diversos medios en la transmisión de un mensaje o idea, esta ha sido de diferentes maneras como: los medios de publicidad, el entretenimiento y por supuesto en la educación.
- El uso de estas tecnologías dan como resultado nuevas formas de interacción por lo que se hace necesario determinar nuevas estructuras y actividades, donde las nuevas tecnologías se refuercen a los métodos tradicionales y faciliten el trabajo cooperativo, incentivando a poner en juego tácticas socio-afectivas que apoyen el proceso de enseñanza / aprendizaje valiéndose de diversas herramientas de comunicación como

el correo electrónico, los foros de discusión, el chat, la teleconferencia, etc.

En base a lo anterior los diferentes sectores del sistema educativo tienen una ardua labor para lograr una sinergia. En esta tesis se quiere contribuir con una herramienta para elaborar materiales educativos basados en WBE para el reconocimiento visual, que este orientada hacia los educadores y niños, que ayuden a estos últimos en la estimulación visual en sus primeras etapas del aprendizaje de la lectura.

## **1.2 Planteamiento del problema y justificación**

Esta tesis propone una herramienta que auxilie a los educadores con mínimos conocimientos en informática, para armar sus materiales didácticos, basados en WBE, con facilidad y pueda emplearlos para lograr la estimulación visual en los alumnos. El primer paso es diseñar el escenario, plantear los objetivos por parte del educador y posteriormente proporcionar, bajo su supervisión y evaluación, a los alumnos para su uso bajo el paradigma del WBE.

## **1.3 Objetivos**

El presente trabajo tiene como objetivos los siguientes:

### **1.3.1 Objetivo general**

Desarrollar una herramienta apoyada de la arquitectura Cliente / Servidor, basada en el paradigma WBE para que, a través de componentes multimedia orientados a objetos permita armar material didáctico que estimule el reconocimiento visual en niños que estén en la etapa del aprendizaje de la lectura.

### **1.3.2 Objetivos específicos**

- 1) Desarrollar un subsistema basado en *Web*, que permita al profesor armar materiales didácticos dirigidos a la estimulación visual.
- 2) Desarrollar un subsistema basado en *Web* que permita desplegar componentes multimedia a través de imágenes, sonidos, animaciones y videos que ayuden a estimular el reconocimiento visual en niños durante la etapa de aprendizaje de la lectura a través de la computadora.
- 3) Desarrollar un subsistema basado en *Web*, para la administración de los usuarios que accedan al sistema en apoyo al profesor, por medio de un administrador, para dar altas, cambios y bajas de usuarios.

## 1.4 Límites y alcances

En el presente trabajo de tesis se describe el desarrollo de un sistema bajo el paradigma de WBE con los siguientes límites y alcances:

- Este sistema está dirigido para que los educadores puedan armar materiales didácticos enfocados al reconocimiento de estímulos visuales.
- Estos materiales didácticos podrán ser guardados para reutilización del propio autor u otros educadores, ya que al crearlos, éstos quedarán disponibles para la comunidad inscrita en el sistema.
- El acceso a estos materiales didácticos, estará restringido a usuarios que estén dados de alta en el sistema.
- El educador puede utilizar los componentes de los materiales didácticos, con el objeto de madurar la percepción visual del alumno en las primeras etapas del aprendizaje de la lecto-escritura.
- Se podrá utilizar el sistema, siempre y cuando exista una infraestructura de cómputo adecuada y cuente con un buen ancho de banda suficiente o acorde a las necesidades para la conexión a la Internet / Web, estas características son descritas en el Anexo B.
- La realización de los recursos multimedia se diseñarán en otras herramientas y el sistema los integrará por medio de componentes de software multimedia.

## 1.5 Beneficios esperados

Los beneficios que se esperan obtener del desarrollo de esta tesis son los siguientes:

- Proporcionar a los educadores una herramienta para armar material didáctico, que complemente los materiales tradicionales para reconocimiento de estímulos visuales como madera, plastilina, o la elaboración de plantillas difíciles de realizar, modificar y transportar.
- Dar al educador una herramienta para exponer sus materiales didácticos, para que pueda utilizarlos en los centros educativos o los hogares, con el objeto de obtener la maduración de la percepción visual de los alumnos.
- Facilitar al educador materiales didácticos que le permitan auxiliar a los alumnos durante el proceso de la lecto-escritura.
- Reducir los costos derivados del manejo de licencias de software.

## 1.6 Organización de la tesis

La presente tesis se organizó a través de 6 capítulos los cuales se describen a continuación:

1. Capítulo 1. Se presenta una introducción sobre la Educación basada en *Web*, los aspectos que intervienen en la lectura, la importancia del proceso de la lecto-escritura, la razón del enfoque hacia las gnosias visuales, el planteamiento del problema, la justificación, los objetivos a alcanzar y los beneficios esperados del desarrollo de esta tesis.
2. Capítulo 2. Se describe el estado del arte con soluciones afines de software existente que se aplica en niños durante las primeras etapas de la lecto-escritura y a partir de ello proponemos y justificamos nuestra solución.
3. Capítulo 3. En este capítulo se lleva a cabo el análisis y diseño del sistema propuesto. En este capítulo se definen las pantallas de interfaz y diagramas UML del sistema.
4. Capítulo 4. En este capítulo se describe la implementación del sistema.
5. Capítulo 5. En este capítulo se realizan pruebas al sistema y se muestran sus resultados.
6. Capítulo 6. En este capítulo se presentan las conclusiones de la tesis y se enmarcan los trabajos a futuro.
7. Al final se presentan las referencias bibliográficas y electrónicas consultadas durante la elaboración de la tesis, y tres anexos: manual del usuario, especificaciones del hardware y software; y las encuestas realizadas a profesores en la etapa de prueba.

---

---

## CAPÍTULO 2

### ESTADO DEL ARTE

---

---

En este capítulo se discutirán los conceptos básicos de la lecto-escritura, así como los niveles cognitivos de los niños en sus inicios hacia la lectura. Después se compararán varios tipos de software de autores distintos, que han ocupado la computadora como herramienta para la aplicación de ejercicios que intervienen en la estimulación visual. Finalmente, se expone una propuesta de solución al problema a resolver.

## 2.1 Introducción

Una definición del concepto de lecto-escritura es la del investigador Azcoaga [1] quien señala:

*“La lecto-escritura es la capacidad para leer y escribir. Para que el proceso de la lectoescritura tenga lugar es indispensable la interacción de un conjunto de actividades neurofisiológicas en los sectores superiores del sistema nervioso central. La particularidad de las funciones cerebrales superiores respecto del aprendizaje de la lecto-escritura se describen con el nombre de lenguaje, praxias y gnosias, descritas más adelante”.*

Así mismo, Azcoaga señala que las funciones cerebrales superiores desempeñan la labor de analizadores mediante los órganos de los sentidos, que descomponen la información recibida por los analizadores auditivos, visual, táctil, motor, etc. Las funciones realizadas son de análisis y síntesis para la consolidación de estereotipos (esquemas o patrones sensoperceptivos fijos).

Las funciones cerebrales superiores que intervienen en el proceso de la lecto-escritura son: lenguaje, praxia y gnosia:

- El lenguaje es la capacidad de comunicación que emplean los seres humanos para expresar a sus semejantes sus sentimientos e ideas, por medio de un conjunto ordenado de signos orales o escritos. Éste se puede ubicar en tres etapas de desarrollo las cuales se describirán a continuación:
  1. La primera etapa de la comunicación es una etapa preparatoria para la adquisición del lenguaje, la cual tiene dos periodos sucesivos del juego vocal siendo los siguientes:
    - a) Propioceptivo: ciertas actividades innatas (respiración, grito, llanto, succión y deglución) van determinando fórmulas sensoriales y motoras, formando el material constituido del juego vocal; se caracteriza por la coincidencia de cierto número de aferencias que convergen por su simultaneidad, a través de las repeticiones que son las que dan lugar a la consolidación de estereotipos. Cada estereotipo es diferente de los demás es decir, que en cada uno de ellos participa una variación que da lugar a la creación de nuevos estereotipos a partir de los consolidados.
    - b) Propioceptivo-auditivo: comienza en el aprendizaje propiamente dicho, el niño refuerza las emisiones que coinciden con los estímulos auditivos que

recibe constantemente, a estas emisiones se les denomina “estereotipos fonomáticos”.

2. La segunda etapa de la comunicación se caracteriza por la adquisición progresiva de los “estereotipos motores verbales”, que resultan del conjunto de emisiones de estereotipos fonemáticos en un activo proceso analítico-sintético que tiene como protagonista al analizador “motor verbal cinestésico”.

Al mismo tiempo paulatinamente desarrolla la capacidad de adquisición de contenidos significativos para una palabra dada, ya que el análisis y síntesis de los “estereotipos verbales”, se hacen con el “analizador verbal”.

3. La tercera etapa (es el segundo nivel lingüístico de comunicación) es el lugar donde se tienen actividades analítico-sintácticas de los analizadores “motor verbal cinestésico”; sirven en las funciones lingüísticas y en los aspectos psicológicos que apoyan el lenguaje. Por lo que el lenguaje, interviene en el proceso de aprendizaje de la lecto-escritura.

- La praxia es una determinada organización de movimientos en secuencia que tienden a un objetivo. En el proceso de aprendizaje interviene la actividad motora la cual implica constantemente, la contracción de cada músculo, que origina un mensaje sensorial que llega a la corteza cerebral.

Estos mensajes generados por la contracción de los músculos se denominan aferencias "propioceptiva" (sensación de uno mismo) o "cinestésico" (sensación de movimiento).

El eslabón aferente está constituido por los estímulos de los músculos, tendones y articulaciones, llegando a la zona frontal de la corteza cerebral.

Estas actividades motoras pasan por diferentes periodos:

- Generalización primaria: incluye una cantidad de actividades musculares no ligadas a la realización estricta de ese comportamiento motor.
- Labilidad: Existen errores en la sucesión o en el acierto que caracterizan esa actividad motora. A través de la repetición de actividades musculares en una cierta simultaneidad, se consolida la síntesis correspondiente de esas aferencias propioceptivas a la corteza cerebral.
- Consolidación: En esta etapa se considera completamente organizada la actividad motora a través de la repetición y el reforzamiento, formándose

la sucesión de estereotipos cinestésicos-motores o propioceptivo-motores en sentido fisiológico.

- La gnosia es la capacidad de reconocimiento sensorial respecto de hechos externos al individuo.

En el proceso de aprendizaje se alcanza esta capacidad de reconocimiento por la actividad analítica-sintáctica de diversos analizadores. En general, uno, dos o más constituyen los protagonistas principales de determinada gnosia. En otros casos varios analizadores intervienen en la constitución de gnosias complejas, tal es el caso de las gnosias que intervienen en el aprendizaje de la lecto-escritura. La adquisición de las gnosias consiste en la aferencia simultánea de conjuntos de estímulos que llegan a la corteza cerebral, creando las condiciones adecuadas para una síntesis y cuando la coincidencia se repite varias veces, la síntesis tiende a consolidarse.

Por lo que el reconocimiento de estímulos auditivos (gnosia auditiva), visuales (gnosia visual), de espacio (gnosia espacial) y tiempo - espacio (gnosia temporo-espacial) son factores que intervienen en el proceso de aprendizaje de la lecto-escritura.

La iniciación del aprendizaje de la lecto-escritura está constituido por un proceso continuo de adquisición de reconocimiento visoespacial. La identificación de formas, letras, sílabas, palabras, con su sonido, hace que el alumno consolide un nuevo tipo de síntesis entre un estereotipo fonemático, un estereotipo motor verbal y un estereotipo verbal simultáneamente.

Como parte de la lecto-escritura se da un reconocimiento de estímulos visuales que se localiza dentro del factor sensorial-motor, en donde el niño con base en una actividad va elaborando un reconocimiento, distinción de formas y comprensión de su significado de acuerdo a la ordenación de espacio y tiempo de signos gráficos que le permitan alcanzar el nivel de madurez para el aprendizaje de la lectura.

A continuación mostramos algunos de los programas elaborados en computadora para el ámbito educacional, que son utilizados por distintos terapeutas, educadores y padres de familia, para estimular el reconocimiento de estímulos visuales y auditivos:

## **2.2 Software educativo - WIN-ABC**

El objetivo principal de WIN-ABC es trabajar de forma sencilla, agradable e intuitiva las técnicas de instrumentos básicos (lectura, escritura y de cálculo

matemático) [2] y estimular la motivación hacia el aprendizaje de estas técnicas básicas, fomentando la autonomía del niño / niña. También puede facilitar el aprendizaje de una segunda lengua (caso de comunidades con lengua propia o dialecto).

Además de los objetivos propios de las actividades, la utilización del ratón como instrumento de comunicación con la computadora, facilita en el alumno su desarrollo de la coordinación visomotora y la movilidad de los ojos le ayudará a la disociación de manos y dedos; esto en su conjunto, también, le aportará madurez para afrontar el aprendizaje de la lectura y la escritura.

El programa está dirigido especialmente a niños con necesidades educativas especiales que no hayan adquirido, o les cueste trabajo adquirir, los conocimientos básicos. También es adecuado para estimular a niños a partir de 4 años sin problemas de aprendizaje, y especialmente a niños que, debido a su inmadurez perceptual y pobre destreza en ejercicios de papel y lápiz, no pueden progresar en la escuela.

En la Figura 2.1 se puede observar una imagen del software WIN-ABC, éste es un programa educativo compuesto por más de 50 actividades integradas en cinco menús: Identificar, Copiar, Asociar, Memorizar y Calcular.



Figura 2.1. Presentación del software educativo: WIN-ABC.

La Figura 2.2 muestra la actividad rotación / inversión de dibujos que consiste en ejercicios de localización de dibujos iguales al modelo tratando de distinguir los dibujos simétricos o invertidos. El objetivo es desarrollar la capacidad para situar y localizar

objetos en una posición espacial determinada para evitar las confusiones que algunos niños presentan durante el aprendizaje de la lecto-escritura en relación a letras que invertidas o giradas tienen un significado distinto: d, b, p y q, así como evitar que el alumno gire o invierta las letras al escribir.



Figura 2.2. Pantalla de WIN-ABC del ejercicio rotación-inversión.

### 2.3 Software educativo - EL MONO COCO

Las actividades que componen el juego “El mono Coco” [3], van encaminadas a la consecución de una serie de objetivos de desarrollo de capacidades, como: resolver laberintos, discriminación visual, etc. La elección del mayor entre varios objetos o animales, tiene en cuenta que el tamaño visual del objeto no corresponda al objeto o animal real. Esto quiere decir que el alumno puede elegir entre varios objetos dibujados con el mismo tamaño pero que realmente son muy diferentes en lo que respecta a dicha propiedad; por ejemplo: elegir entre una casa y un lápiz (dibujados los dos con el mismo tamaño). Los diferentes ejercicios que contiene este software son los siguientes:

- La resolución de laberintos.
- El reconocimiento visual de letras, objetos parecidos, formas geométricas, etc.
- La resolución de rompecabezas sencillos.
- La memoria visual.
- Las series de objetos por colores y tamaños.

En la Figura 2.3 se presenta el menú de las actividades que contiene el programa.



Figura 2.3. Menú de selección de actividades del software educativo “El mono Coco”.

## 2.4 Software educativo – JUGAR CON

El objetivo general del programa “Jugar con” [4], es facilitar el proceso de enseñanza-aprendizaje de actividades mediante la computadora; el programa estimula el desarrollo de capacidades físicas, afectivas, intelectuales y sociales del niño, relacionando diferentes áreas o ámbitos curriculares.

En la Figura 2.4 se puede observar la portada de “Jugar con” que se compone de tres actividades para el aprendizaje o la evaluación, las cuales son:

- Colorear.
- Formas.
- Posiciones corporales.

Cada una de estas, a su vez, presenta de dos a cuatro niveles diferentes de dificultad (Nivel 0, 1, 2 y 3), que tratan de introducir progresivamente al alumno en los conceptos básicos de los colores, las formas y las posiciones corporales.



Figura 2.4. Portada del Software educativo "Jugar con".

Todos los programas presentan al comienzo las opciones de aprendizaje (con parpadeo del estímulo para seleccionar en la pantalla), o de evaluación (sin parpadeo). La ventaja para el educador es que puede hacer una evaluación inicial del conocimiento del alumno y en base a ella pasar a la modalidad de aprendizaje. Cerrando el ciclo con otra evaluación para comprobar los conceptos adquiridos.

## 2.5 Software educativo – ACTIVIDADES DE LECTO-ESCRITURA

Este software esta dirigido a niños que se encuentran en el proceso del aprendizaje de la lectura [5]. En el menú inicial mostrado por la Figura 2.5 se observan los diferentes bloques de letras que utiliza con las sílabas.

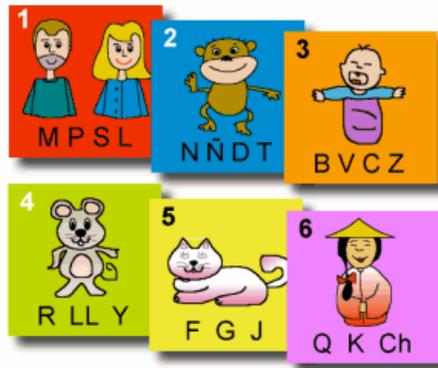


Figura. 2.5. Portada del software “Actividades de lecto-escritura”.

En esta actividad se realizan ejercicios que unen sílabas a través de una línea hacia el sonido de la sílaba escuchada, realizadas con las diferentes sílabas del abecedario, como se muestra en la Figura 2.6.

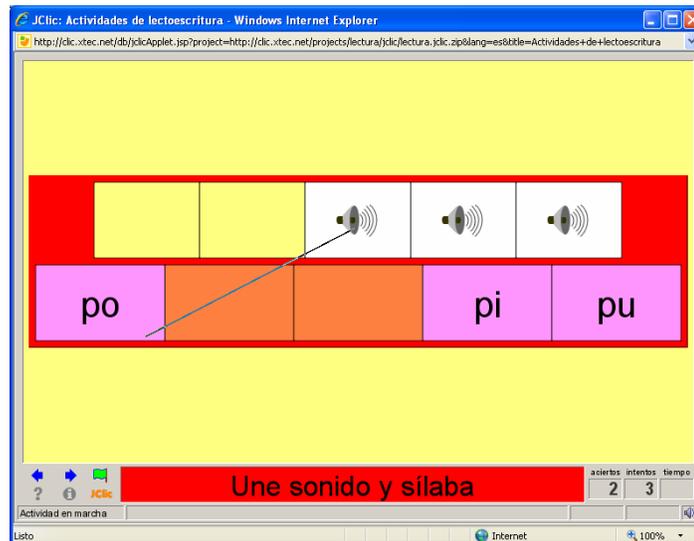


Figura 2.6. Uniendo sonido escuchado con la sílaba correspondiente.

## 2.6 Software educativo - LOS COLORES

El software cumple una función fundamental en la adquisición del conocimiento en edades tempranas. “LOS COLORES” [6] apoya a educadores para que los niños distingan los colores, a través de este software interactivo que se muestra en la Figura 2.7.

Andrés Sánchez Márquez ✉

**Equipo de Apoyo a Ciegos y Deficientes Visuales de Jerez de la Frontera**

Jerez de la Frontera (Andalucía)

Proyecto de actividades sobre colores (rojo y azul) destinado a Educación infantil y Educación Especial. Está adaptado para discapacitados visuales y/o motóricos usando tableta digitalizadora.

Adaptado para discapacidad visual baja visión.

Área	Plástica y visual, Diversos
Niveles	Infantil (3-6), Primaria (6-12)
Fecha	03/11/06
Licencia de uso	 Esta obra está bajo una licencia de Creative Commons



Figura 2.7. Descripción del software educativo “LOS COLORES”.

La intención de este software es que el niño logre el reconocimiento de los colores. En la Figura 2.8 muestra la actividad para lograr el objetivo, a través de un sonido emitido por la computadora, el niño debe señalar el color dando un clic sobre la figura que tiene el color que escuchó.

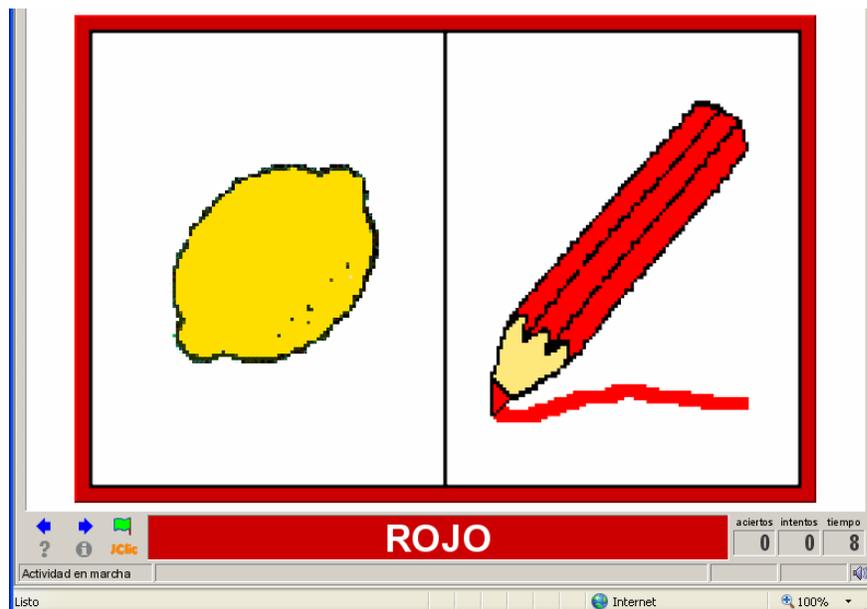


Figura 2.8. Actividad para elegir el color escuchado.

## 2.7 Software educativo - EN SERIE

El aprendizaje de las matemáticas no se debería fundamentar sólo en lo formal y deductivo sino también en lo empírico e inductivo. El programa “En serie” [7] se lleva a cabo el aprendizaje como se muestra en la Figura 2.9, el cual a través de operaciones concretas como contar, comparar, clasificar y relacionar, auxilia al niño en la

adquisición de representaciones lógicas y matemáticas, haciendo generalizaciones de un algo en un todo.

María del Carmen Manuel Vez ✉

**CEIP Casería de Ossio**  
San Fernando - Cádiz

Proyecto de actividades sobre seriación para educación infantil. El proyecto está dividido en tres partes cada una de ellas en función del número de elementos que constituyen la serie.

Área	Matemáticas
Nivel	Infantil (3-6)
Fecha	26/02/07
Licencia de uso	Esta obra está bajo una licencia de Creative Commons

Figura 2.9. Descripción del Software educativo “En serie”.

En la Figura 2.10 se muestra la dinámica de estas actividades en las que se tiene que completar la serie con respecto a la figura que lógicamente sigue después de que la computadora propone un patrón. El programa tiene 3 niveles con 2, 3 y 4 figuras a ordenar.

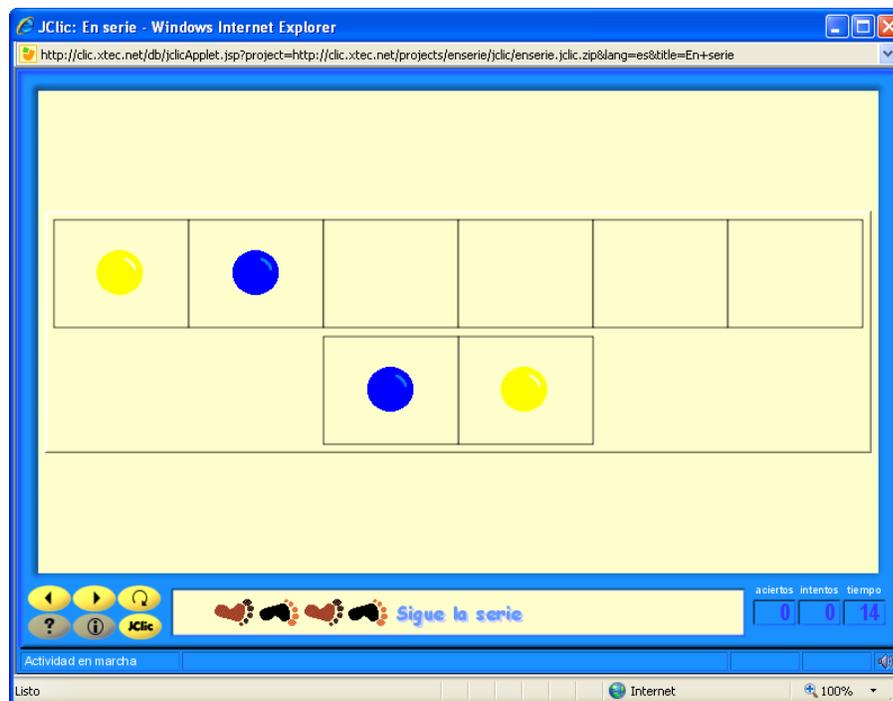


Figura 2.10. Actividad de dar un clic en la figura que sigue.

## 2.8 Software educativo - LOS TAMAÑOS

En nuestra vida cotidiana el distinguir entre tamaños es algo natural, pero en los comienzos de la madurez visual que necesitan los individuos que apenas aprenden a leer es indispensable esta distinción entre tamaños. En la Figura 2.11 se muestra la

portada del software “LOS TAMAÑOS “ [8], como herramienta de apoyo para distinguir entre tamaños.

Andrés Sánchez Márquez 

**Equipo de Apoyo a Ciegos y Deficientes Visuales de Jerez de la Frontera**  
 Jerez de la Frontera (Andalucía)

Paquete de actividades sobre tamaños (grande - pequeño) destinado a Educación infantil y Educación Especial. Está adaptado para discapacitados visuales y para ciegos, y discapacitados motóricos usando tableta digitalizadora.

Adaptado para discapacidad visual baja visión.

Área	Plástica y visual, Diversos
Niveles	Infantil (3-6), Primaria (6-12)
Fecha	03/11/06
Licencia de uso	 Esta obra está bajo una licencia de Creative Commons



Figura 2.11. Descripción del Software educativo “LOS TAMAÑOS”.

En la Figura 2.12 se presenta una actividad que desarrolla este software la cuál despliega distintas figuras en un tamaño pequeño y otro grande, con un sonido y un mensaje que le indica al alumno que escoja el grande o el pequeño, cuando el alumno da un clic con el ratón en la figura elegida, el programa responde con un sonido de ¡Bien hecho! y un mensaje escrito en pantalla.

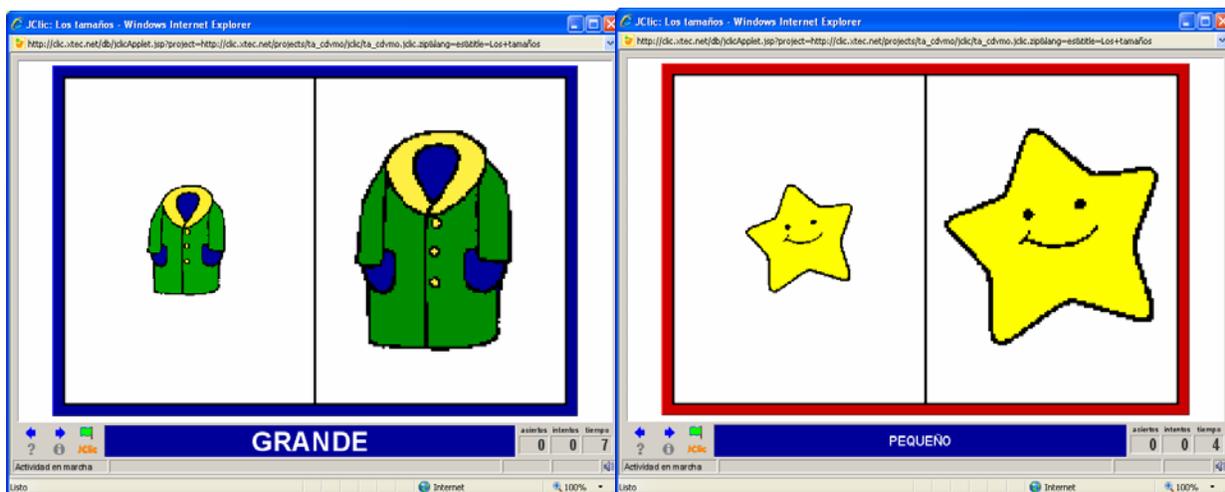


Figura 2.12. Actividad de dar un clic en la figura grande o pequeña.

## 2.9 Software educativo – LAS LETRAS SUENAN

Las características gráficas son los patrones visuales particulares de una palabra, formados por una disposición única de letras que forman la palabra (baño contra daño

u hola contra bola). Al aprender a leer, los niños deben almacenar información gráfica (o de apariencia) para distinguir de manera precisa entre las palabras impresas, por ello las actividades propuestas en el software “Las letras que suenan” [9] coadyuvan al alumno en el reconocimiento del patrón y del sonido que emite este patrón. En la Figura 2.13 se presentan las actividades en que se dividen por bloques de letras.

Antonio Piñeiro Dopico 

**CEIP Rabadeira**  
Oleiros (Galiza)

Actividades para el aprendizaje de la lectoescritura complementarias del material impreso del mismo título disponible en: <http://centros.edu.xunta.es/ceipderabadeira>.

Las actividades están organizadas en cuatro paquetes. En los dos primeros se trabajan las vocales y las consonantes S, M, F, R y L utilizando sólo el ratón. En los otros dos se trabajan todas las consonantes e incluyen actividades de escritura con teclado.

Área	Lenguas
Niveles	Infantil (3-6), Primaria (6-12)
Fecha	27/04/06
Licencia de uso	 Esta obra está bajo una licencia de Creative Commons

**las letras suenan**

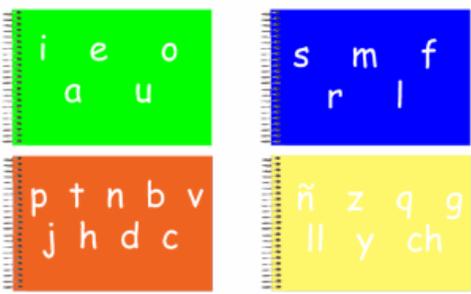


Figura 2.13. Descripción del Software educativo “Las letras que suenan”.

La Figura 2.14 muestra una actividad en la que se despliegan diferentes letras en la pantalla; el alumno debe seleccionar, con el ratón, aquellas que sean iguales a la del patrón presentado, y se emite un estímulo audible de correcto o incorrecto cuando selecciona la letra correcta.

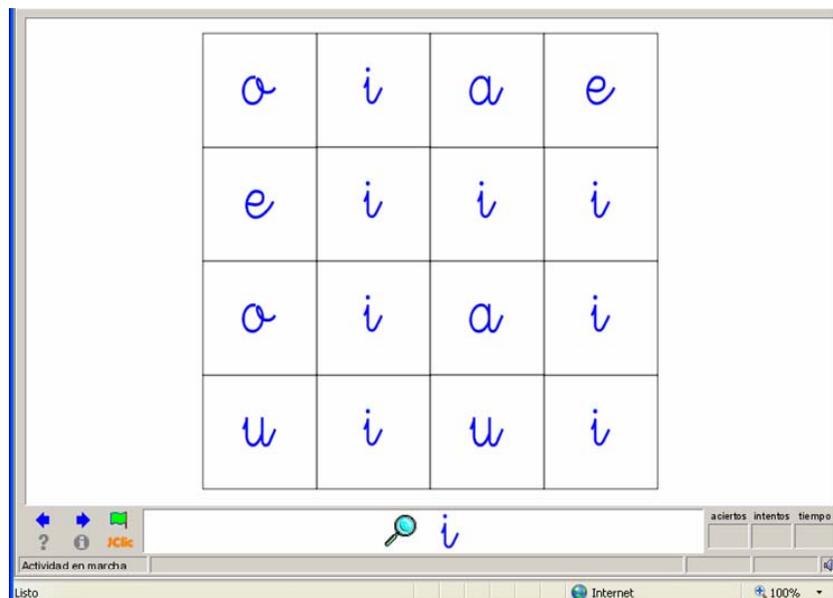


Figura 2.14. Actividad de dar un clic en la letra que escuchó o vio en el patrón.

## 2.10 Software educativo - APRENDE LOS ANIMALES

El software “Aprende los animales” [10] consta de cinco actividades diferentes, enfocadas al aprendizaje de los animales. En la Figura 2.15 se encuentran descritas dichas actividades y una portada de este software.

Teresa López Vicente ✉

Actividades sobre los animales para educación infantil y educación especial. En la actividad encontramos 5 partes:

- Animales
- Puzzles
- Memory
- Señala el animal
- Busca la pareja

Área	Ciencias experimentales
Nivel	Infantil (3-6)
Fecha	20/02/06

**MONO**



**OSO**



**LEÓN**



**PATO**



Figura 2.15. Descripción del Software educativo “Aprende los animales”.

En la Figura 2.16 se presenta una actividad que muestra la figura del animal, la forma de escribirse y su pronunciación. En la Figura 2.17 se muestra la actividad en la que el alumno tiene que armar un rompecabezas con la figura de un animal. La Figura 2.18 visualiza cómo el alumno juega a través de la computadora un memorama, es decir tiene que encontrar las cartas de las figuras iguales, las cuales se encuentran ocultas. La Figura 2.19 da una muestra de diferentes animales y el programa menciona un animal y el alumno tiene que seleccionar el que escuchó. En la Figura 2.20 se aprecia la escena donde el alumno debe establecer la relación entre la figura del animal y el nombre del mismo.

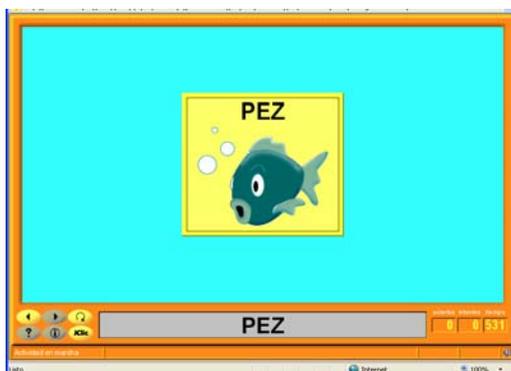


Figura 2.16. Actividad de reconocimiento del animal.



Figura 2.17. Actividad de rompecabezas.



Figura 2.18. Actividad de memorama.



Figura 2.19. Actividad de señala al animal.

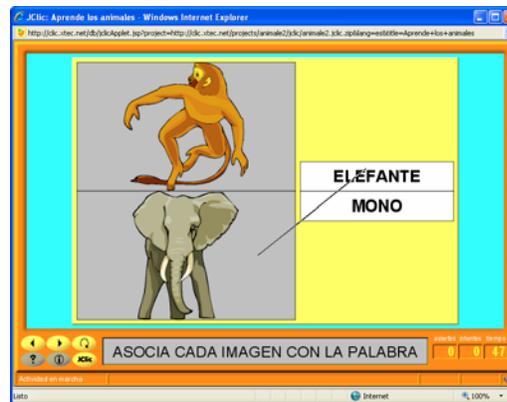


Figura 2.20. Actividad que une el nombre con la figura que le corresponde.

## 2.11 Software educativo – FIGURAS

El software "Figuras" [11] gira entorno al reconocimiento de las figuras básicas (cuadro, círculo y triángulo) utilizando los colores. Inicia el programa presentando de manera auditiva los colores, resaltando la figura que tiene el color. La portada se presenta en la Figura 2.21.

César Bardillo Ramos

**CP María Díaz de Muñoz**  
Béjar - Salamanca (Castilla y León)

Actividades de reconocimiento de figuras básicas (círculo, rectángulo, cuadrado y triángulo) asociando dicho reconocimiento a los colores básicos (rojo, amarillo, verde y azul).

Área	Matemáticas
Niveles	Infantil (3-6), Primaria (6-12)
Fecha	17/01/06
Licencia de uso	 Esta obra está bajo una licencia de Creative Commons

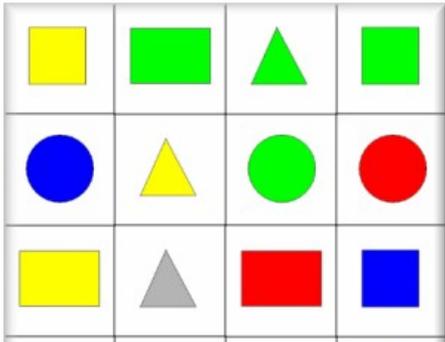


Figura 2.21. Descripción del Software educativo "Figuras".

Posteriormente en la Figura 2.22 visualiza los colores que, a su vez, mediante una instrucción auditiva menciona los nombres de los mismos. En la Figura 2.23 muestra la actividad donde el alumno debe seleccionar las figuras del color que el software haya nombrado con audio.

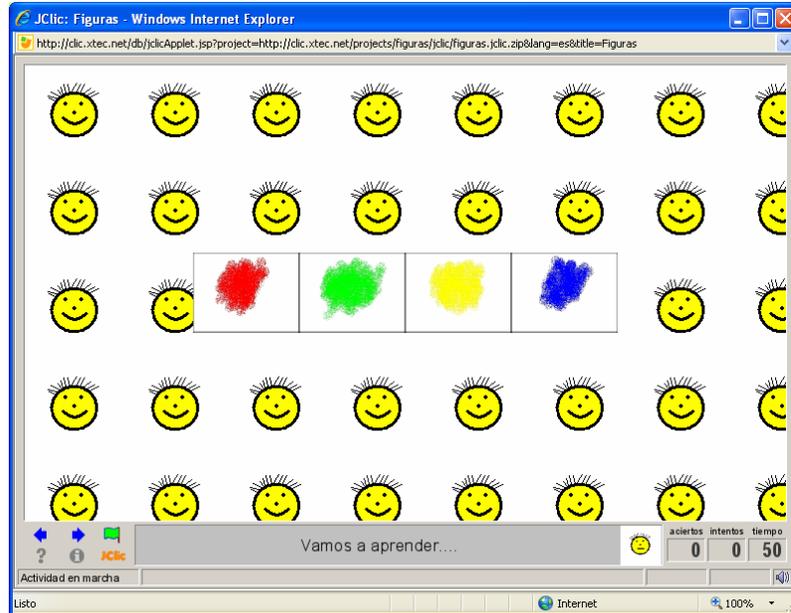


Figura 2.22. Reconocimiento de los colores.

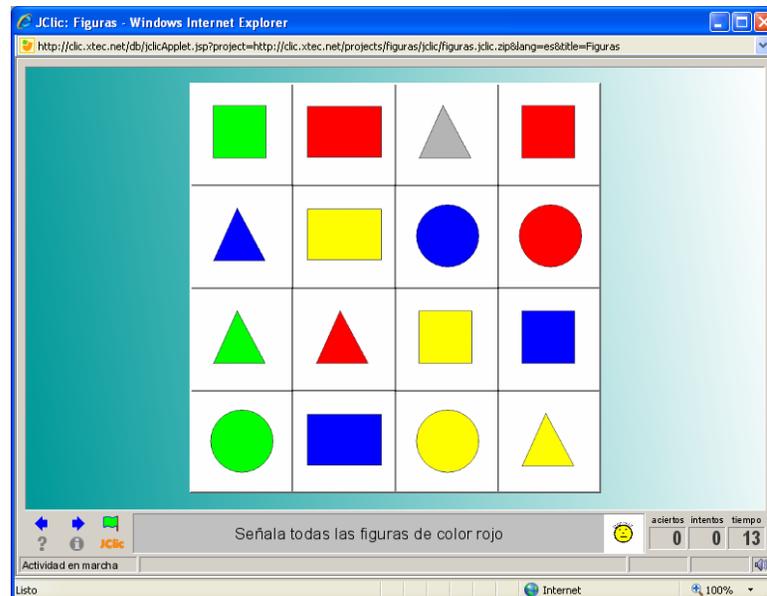


Figura 2.23. Reconocimiento de las figuras.

## 2.12 Software educativo - LOS PRIMEROS NÚMEROS POR VÍA VISUAL DIRECTA

Al igual que el reconocimiento de las letras, el reconocimiento de los números es muy importante durante los inicios del aprendizaje de la lectura y escritura en los individuos, por ello la preocupación de llevar el contenido de los libros de preescolar a un software [12] que ayude en el aprendizaje de los números como se observa en la Figura 2.24.

Eduardo Martín Sánchez ✉

**CEIP Cristóbal Colón**  
El Puerto de Santa María - Cádiz (Andalucía)

Con esta aplicación de clic se intenta que los niños y niñas se formen imágenes de cada número con los instrumentos más usados en la escuela: los dedos de las manos, las regletas de Cuisenaire, los puntos del dado... En total hay 7 grupos de actividades. En los dos primeros se trabaja con grupos de 4 números (1, 4 y 2, 5) y el cambio de actividad es automático. Pueden hacerlo niños pequeños. Los 3 siguientes son de seis números y se trabajan del 1 al 8. Los dos últimos son en grupos de 9 números (1, 9 y 2, 10).

Área	Matemáticas
Niveles	Infantil (3-6), Primaria (6-12)
Fecha	16/09/05
Licencia de uso	Esta obra está bajo una licencia de Creative Commons

Figura 2.24. Descripción del Software educativo “Los primeros números por vía visual directa”.

En el ejemplo de la Figura 2.25 se muestra una de las actividades del software donde su finalidad es unir las imágenes que representan el mismo número, como se ve en el patrón de la parte inferior de la pantalla.

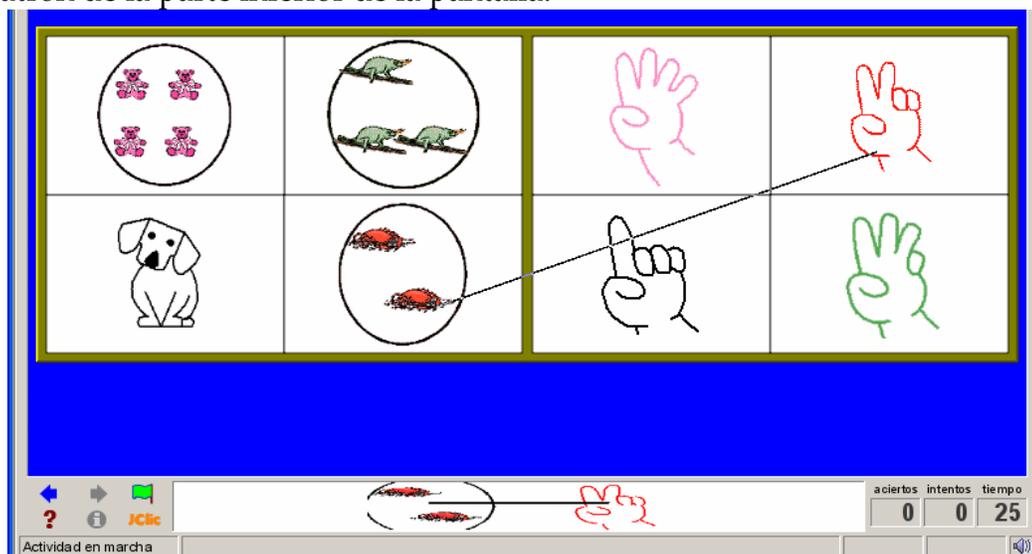


Figura 2.25. Actividad uniendo las dos catarinas con el dos indicado con los dedos.

## 2.13 Software educativo – PACO EL CHATO

Es un software mexicano [13] que relata un cuento con imágenes y audio como se presenta en la Figura 2.26; las actividades que el alumno tiene que hacer es ordenar secuencias de las imágenes según fueron apareciendo en la historia.

Rafael Neftaly Zepeda 

**Centro de Recursos e Información para la Integración Educativa No. 2.**

Ixtlán del Río, Nayarit (México)

"Paco, el chato" es un cuento que hace notar a los escolares la importancia de saber nuestro nombre. Puede emplearse entre las primeras actividades de alfabetización de un niño, por lo significativo que les resulta a ellos trabajar con actividades en las que usan su nombre. La sección de actividades incluye situaciones de análisis de palabra y secuenciación.

Área	Lenguas
Niveles	Infantil (3-6), Primaria (6-12)
Fecha	16/09/05
Licencia de uso	 Esta obra está bajo una licencia de Creative Commons



Figura 2.26. Descripción del Software educativo “Paco el Chato”.

## 2.14 Software educativo – DESBERDINTASUNAK

Este software [14] presenta figuras en las que hay que encontrar las diferencias; un ejemplo se muestra en la Figura 2.27; además, contiene rompecabezas para ser armados por los alumnos y así se da un incentivo motivacional.

Ana M<sup>a</sup> Álvarez de Arcaya  y Blanca Besga 

Esta actividad va dirigida a los niños y niñas del 2º ciclo de Educación Infantil. Las actividades están secuenciadas por años en orden de dificultad. El juego consiste en buscar las diferencias existentes entre 2 dibujos supuestamente iguales.

Intercalados se encuentran puzzles con distinto número de piezas, para hacer el ejercicio más ameno y evitar la rutina. Como desenlace de cada bloque de ejercicios aparece una animación.

Área	Diversos
Nivel	Infantil (3-6)
Fecha	17/09/04
Licencia de uso	 Esta obra está bajo una licencia de Creative Commons



Figura 2.27. Descripción del Software educativo “Desberdintasunak”.

## 2.15 Software educativo - ADENTRO AFUERA

Como se describe en la introducción de la Figura 2.28, en este software [15] se desarrollan las capacidades y destrezas del alumno en los campos visuales y auditivos, para reforzar la madurez que el niño debe lograr en los comienzos del aprendizaje de la lectura y escritura, siendo un factor importante el desarrollar la percepción viso-espacial y el distinguir entre las cosas que están adentro y las cosas que están afuera de otra.

Las capacidades que desarrolla en el niño son: comprensión oral, percepción visual, percepción auditiva, control motriz.

Las destrezas que habilita este software en los niños son: escuchar y ejecutar instrucciones, direccionalidad, memoria auditiva y visual, discriminación auditiva y coordinación óculo-manual.

Viviana Casis Abedrapo 

**Colegio Especial Rotario Paul Harris**  
Las Condes (Xile)

Paquete de actividades que quiere trabajar dos campos de la educación principalmente:

- ♦ Capacidades: Comprensión Oral; percepción visual, percepción auditiva, control motriz.
- ♦ Destrezas: escuchar y ejecutar instrucciones, Direccionalidad, Memoria visual, Motilidad Ocular, Vocabulario visual, Percepción de formas, conciencia auditiva, memoria auditiva visual, discriminación auditiva y coordinación óculo manual.



Área	Diversos
Nivel	Infantil (3-6)
Fecha	10/06/04

Figura 2.28. Descripción del Software educativo “ADENTRO AFUERA”.

## 2.16 Software educativo - JUGANDO ACTIVO MI INTELIGENCIA

Ya con anterioridad se ha hablado de los rompecabezas, de unir las figuras, de la ordenación de secuencias, pero aquí se agregan las sumas y la terminación de palabras, esto quiere decir que implica que los alumnos que interactúen con este software [16] necesitan saber leer y saber sumar. La Figura 2.29 nos describe un poco más estas actividades.

Jenny Silva Aguilera 

**Colegio Teresa Videla de González**  
La Serena (Xile)

Este paquete de actividades está diseñando para niños pequeños, se encuentran actividades de puzzles, identificaciones, encontrar parejas, ordenar secuencias, asociaciones, cantidad, números, sumas básicas, terminaciones de palabras.

Área	Diversos
Nivel	Infantil (3-6)
Fecha	10/06/04
Licencia de uso	 Esta obra está bajo una licencia de Creative Commons



Figura 2.29. Descripción del Software educativo “Jugando activo mi inteligencia”.

## 2.17 Software educativo - ¿CUÁL ES CUÁL?

Como en otros ejercicios vistos en este software se refuerza la percepción espacial (posición en el espacio), para que el alumno logre una destreza en la memoria visual, percepción de forma, la coordinación ojo y mano [17].

Ana M<sup>a</sup> Álvarez de Arcaya  y Blanca Besga 

Esta actividad está desarrollada para niños y niñas de 3, 4 y 5 años. Consiste en una vez variados los elementos de un dibujo en color en diferentes posiciones en el espacio asociarlos con su imagen correspondiente en blanco y negro. La dificultad va de menor a mayor según la edad del alumnado.

Área	Diversos
Nivel	Infantil (3-6)
Fecha	04/11/03
Observaciones	Contenido textual mínimo
Licencia de uso	 Esta obra está bajo una licencia de Creative Commons

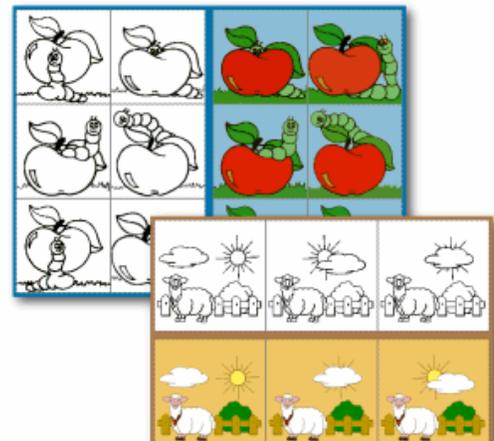


Figura 2.30. Descripción del Software educativo ¿Cuál es cuál?.

Un ejemplo de sus actividades se muestra en la Figura 2.30 y están enfocadas a unir las figuras sin color con su figura igual a color, utilizando 3 grados de dificultad; el primer grado muestra tres figuras, el segundo 4 figuras y el tercero más de 5 figuras.

## 2.18 Software educativo - ANICLIC MÉTODO DE LECTURA

En el software “ANICLIC MÉTODO DE LECTURA” [18] descrito en la Figura 2.31 tiene el propósito de utilizar fonemas que se relacionan a imágenes (principalmente de animales), para facilitar el aprendizaje de la lectura de una forma interactiva y sencilla a través de la computadora, utilizando los recursos multimedia.

Juan Carlos Rodríguez Lorenzo ✉

Aniclic es la versión para Clic de *Animalitos*, método de lectoescritura para las clases de lengua y cultura españolas publicado por el autor en Alemania en 1995.

Sigue la misma secuencia de presentación de fonemas y grafías de *Animalitos*, así como su concepto de enseñanza de la lectura; con las ventajas de motivación que ahora presenta al transformarse en una aplicación informática interactiva.

Tampoco se debe considerar ANICLIC como un auténtico método de lectura, sino como un recurso más que, unido al trabajo diario, propicie el avance en el aprendizaje lector. ANICLIC está pensado para ser usado durante dos o tres cursos; pero como se puede usar como refuerzo educativo (también para alumnos con necesidades educativas especiales), es el profesor/a quien decide el momento y la idoneidad de su aplicación.

Adaptado para discapacidad visual baja visión.

Área	Lenguas
Niveles	Infantil (3-6), Primaria (6-12)
Fecha	16/06/03



Figura 2.31. Descripción del Software educativo “ANICLIC MÉTODO DE LECTURA”.

En la Figura 2.32 presenta una actividad en la que el niño tiene que señalar la imagen que se relaciona con la primera letra con que comienza el nombre de la imagen.

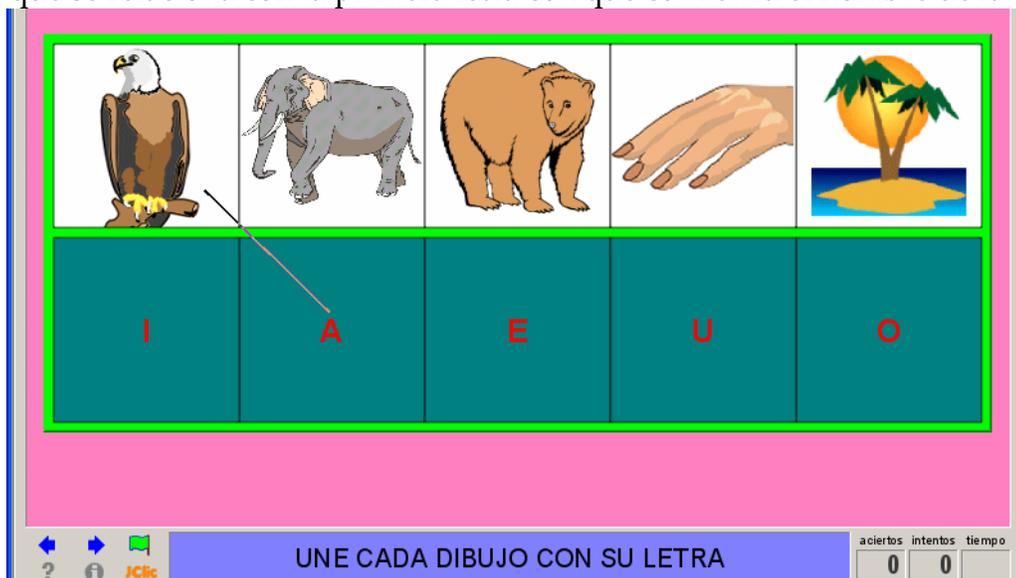


Figura 2.32. Actividad que une la imagen con la letra que comienza.

Otra actividad consiste en que la computadora emite una frase y la muestra en la pantalla, para que el alumno llene los espacios a través del teclado, como se puede observar en la Figura 2.33.

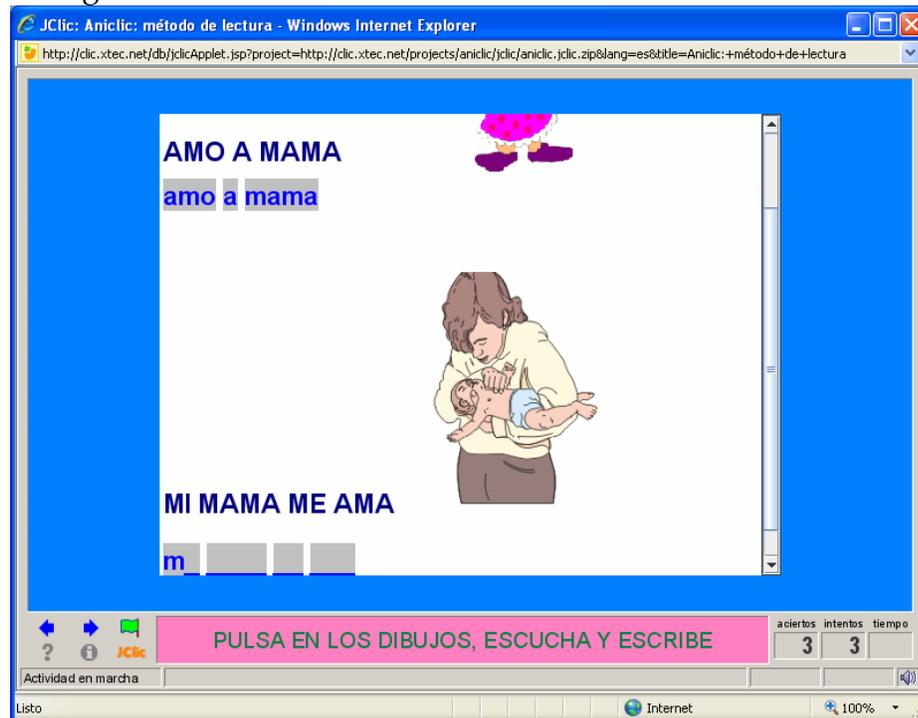


Figura 2.33. Actividad donde se escribe el patrón visto y escuchado.

## 2.19 Software educativo - A GAIVOTA DE 'A COVIÑA'

En la Figura 2.34 se muestra el software "A GAIVOTA DE 'A COVIÑA'" [19] que relata un cuento sobre los problemas que puede acarrear el derramamiento de petróleo; la actividad tiene el objetivo de lograr que el niño ordene las ideas relatadas durante el cuento para que al final el educador evalúe su atención.

Sabela Núñez Singala 

Paquete de actividades para niños y niñas de Educación Infantil y primer ciclo de Primaria en las que se trabaja un cuento sobre las consecuencias del desastre ecológico provocado por el hundimiento del petrolero "Prestige". El paquete consta de tres partes:

- ◆ Cuento
- ◆ Puzzles
- ◆ Comprensión

Área	Ciencias sociales
Niveles	Infantil (3-6), Primaria (6-12)
Fecha	11/06/03



Figura 2.34. Descripción del Software educativo "A GAIVOTA DE 'A COVIÑA'".

Las actividades que realiza son enfocadas para el análisis de situaciones, en la que el niño aumenta su capacidad de memoria y de observación. En la Figura 2.35 se puede observar en la parte inferior una pregunta y el niño tiene que seleccionar la imagen que representa la respuesta correcta de entre las Figuras.



Figura 2.35. Actividad donde hay que seleccionar, la repuesta a lo que se pregunta.

## 2.20 Software educativo - JUGANDO APRENDO

En la Figura 2.36 se muestra la presentación del software “Jugando Aprendo”[20] que tiene por objetivo reforzar la percepción viso-espacial, para que el alumno adquiera destrezas en la memoria visual, para distinguir la forma de los objetos, la coordinación con la mano y los ojos, así como desarrollar su percepción auditiva al escuchar y ejecutar las instrucciones oídas.

Verónica Cares Gallardo ✉

**Colegio Talcahuano**  
Talcahuano (Xile)

Aplicación dividida en cuatro áreas de trabajo:

- ◆ Rompecabezas que van incrementando en el nivel de dificultad.
- ◆ Orientación: Izquierda-derecha, identificando dibujos que van en una u otra dirección.
- ◆ Memorizar: Serie de juegos de "memoria" que van aumentando la cantidad de fichas.
- ◆ Buscar diferencias. Una serie para buenos observadores.



Área	Diversos
Nivel	Infantil (3-6)
Fecha	21/10/02
Última revisión	27/11/02

Figura 2.36. Descripción del Software educativo “Jugando aprendo”.

Los programas de rompecabezas, memoramas, buscando las diferencias, y orientación izquierda-derecha, se visualizan en la Figura 2.27.

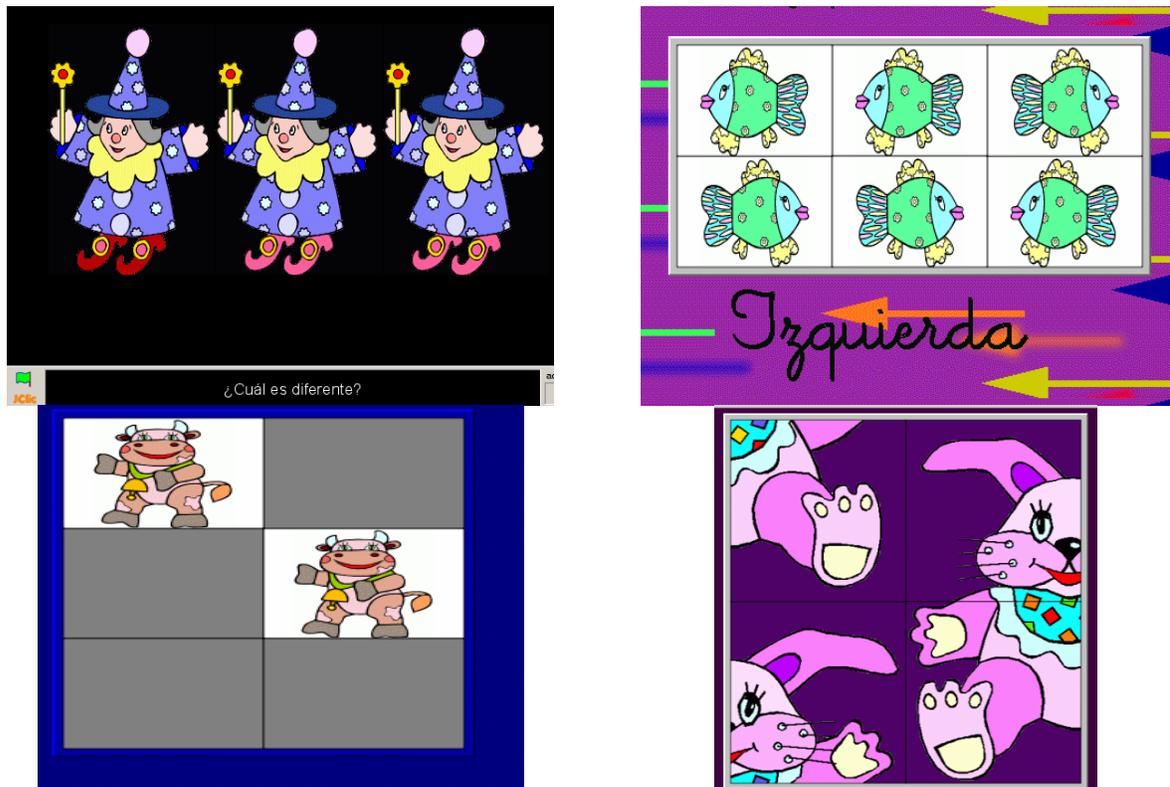


Figura 2.37. Actividades: Busca las diferencia, orientación izquierda-derecha, memorama y rompecabezas.

A continuación se describe un software parecido a la propuesta de esta tesis.

## 2.21 Software – JCLIC AUTOR

Se trata de una herramienta que permite crear y editar las actividades interactivas para hacer más fácil su comprensión. La Figura 2.38 muestra el menú con el cual se interactúa para generar actividades en "Jclic Autor" [21] creando distintos formatos de archivos e imágenes y entornos personalizables.

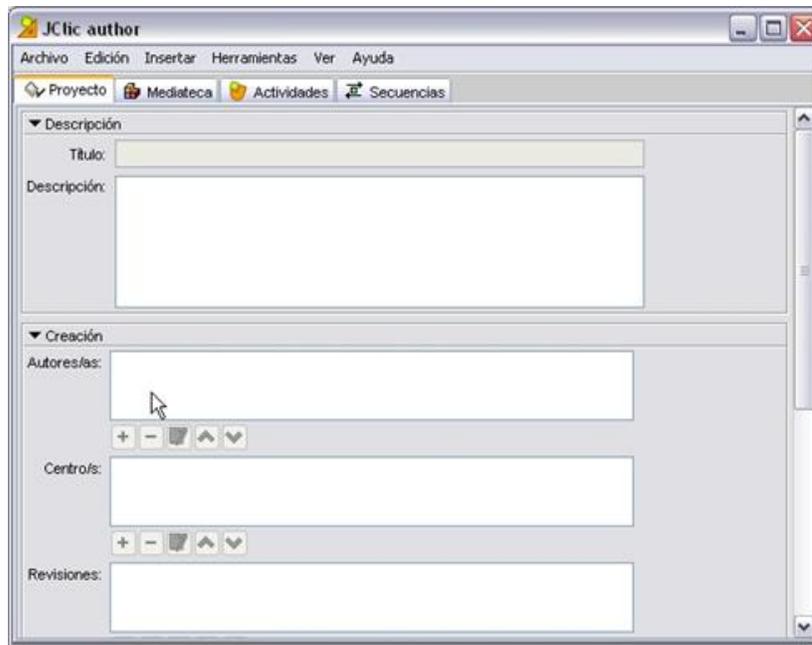


Figura 2.38. Pantalla para desarrollo de actividades.

Con JClíc se puede crear las siguientes actividades: Puzzles o rompecabezas detallados en la Figura 2.39.

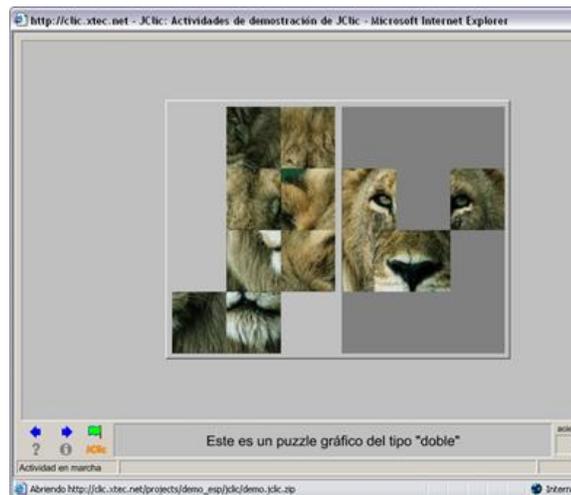


Figura 2.39. Vista de los Puzzles o rompecabezas.

La información que maneja puede ser gráfica, textual o sonora; por ejemplo las asociaciones que se ven en la Figura 2.40 permiten establecer las relaciones que pueden existir entre dos conjuntos. Al igual que en los rompecabezas la información son imágenes, videos, sonidos, texto, etc.

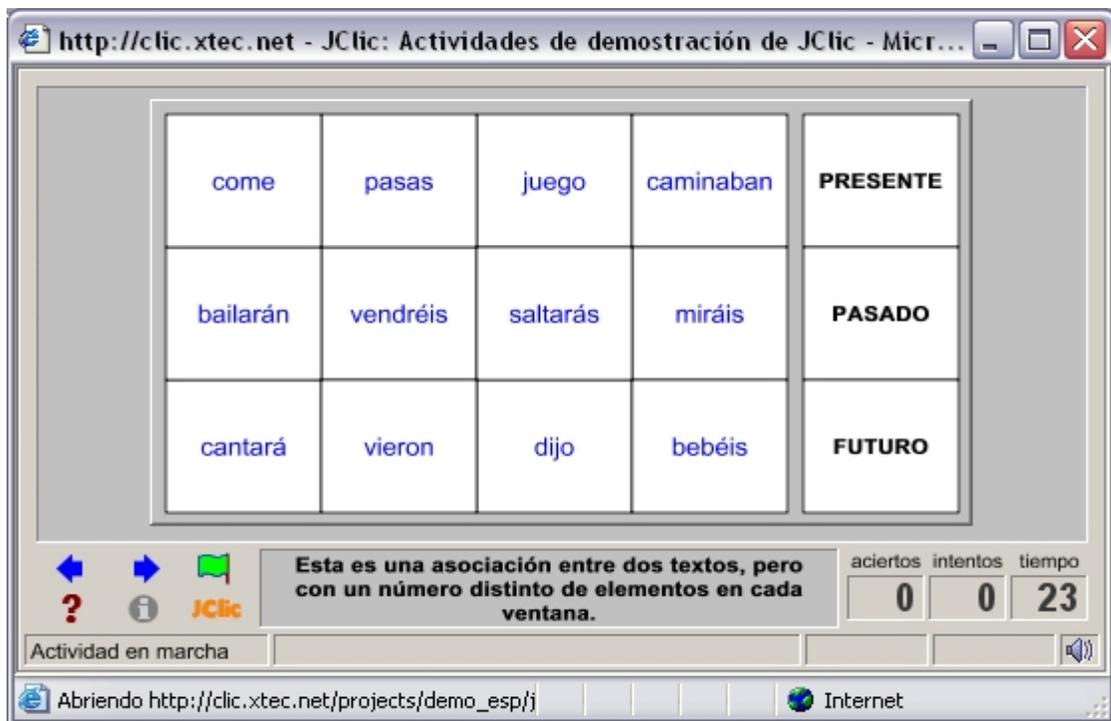


Figura 2.40. Vista de una pantalla con la actividad de Asociación.

En la actividad “Sopas de letras”, representada por la Figura 2.41, es necesario encontrar las palabras escondidas entre en un grupo de letras. A la vez que se van descubriendo las palabras puede aparecer información asociada que presenta en formato gráfico, textos, etc.



Figura 2.41. Vista de una pantalla con la actividad sopa de letras.

La actividad “crucigramas” mostrada en la Figura 2.42 ejemplifica la creación de crucigramas, en el programa se dan definiciones, que pueden tratarse de texto, imágenes o sonidos; en el tablero cuadrado para el que va contestando la actividad se puede ayudar con escribir sus respuestas en los espacios sugeridos.



Figura 2.42. Vista de una pantalla con la actividad sopa de crucigrama.

Las actividades de texto son: las de completar, corregir, seleccionar dentro de elementos, respuesta escrita a casillas, llenar huecos, identificar letras o palabras y ordenar párrafos o palabras. En la Figura 2.43 se muestra la pantalla en la que se escribe el nombre de la imagen señalada en la casilla.



Figura 2.43. Vista de una pantalla con la actividad de texto: respuesta escrita a casillas.

La descarga e instalación de este software se hace desde el buscador de actividades de la página Web de Jclíc, seleccionada la opción “instalar en la computadora”. Al seleccionar esta opción como se puede observar en la figura 2.24 se lleva al usuario a través de las indicaciones hasta tener completamente instalado el software en la computadora y poderse utilizar; en la Figura 2.45 se observa la ejecución de programa instalado.

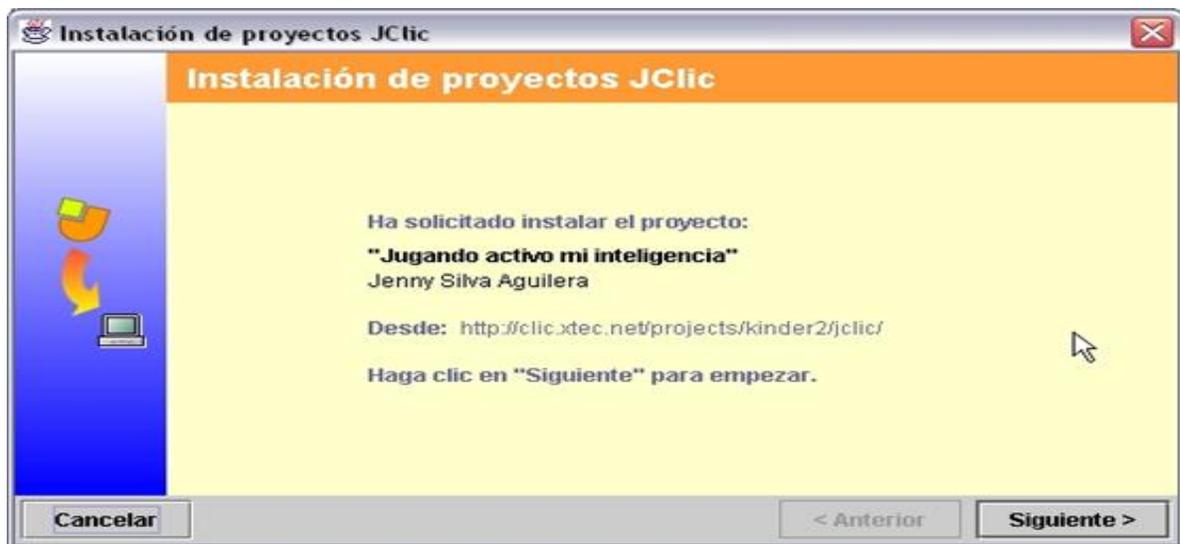


Figura2.44. instalación del paquete jugando activo mi inteligencia en la computadora.

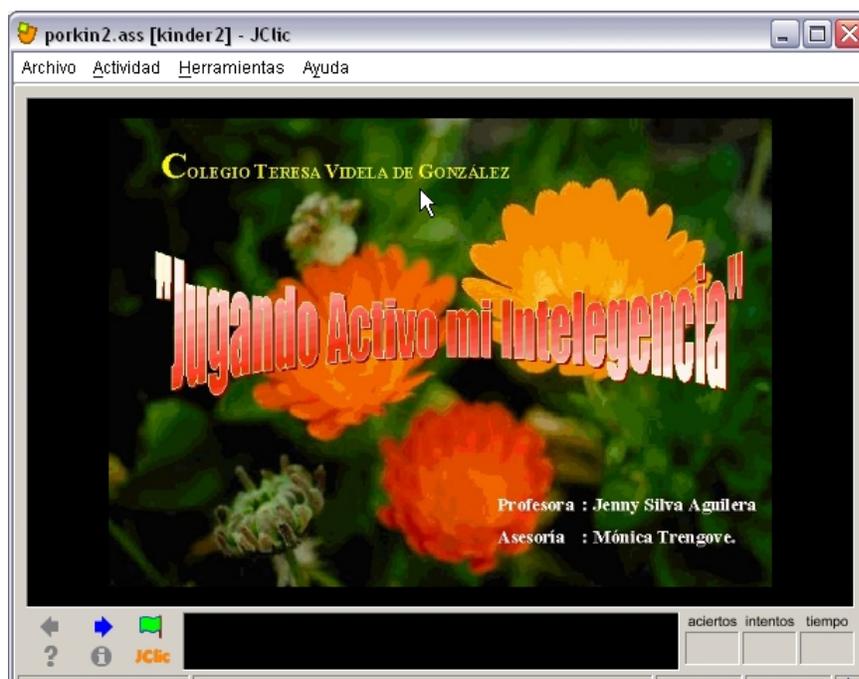


Figura2.45. Programa “jugando activo mi inteligencia” al empezar el programa.

## 2.23 Propuesta de solución

De las diferentes soluciones descritas anteriormente se retomaron distintos aspectos para el desarrollo de un sistema auxiliar de material didáctico para la estimulación de reconocimiento visual en niños usando el paradigma de WBE, que en este trabajo llamaremos “Sistema AMDERV- WBE “ por el título de la tesis: Sistema Auxiliar de Material Didáctico para la Estimulación del Reconocimiento Visual usando el paradigma de Educación Basado en Web.

Se propone diseñar un software de apoyo a profesores, para que los profesores puedan armar diversos materiales didácticos para presentar y manejar información, fortaleciendo el dominio del concepto (reconocimiento visual) y las habilidades del alumno.

Se pretende que el software sea flexible y se pueda adaptar de acuerdo al criterio del profesor, para el armado del material didáctico que empleará con los alumnos; este criterio está determinado por la temática a tratar, ofreciendo una interacción constante, respondiendo a cada acción tomada por el alumno y dirigiéndola hacia otra que produzca o enriquezca su proceso de aprendizaje. Esto permitirá reforzar de manera positiva y retroalimentar en forma constante los avances de los alumnos haciendo hincapié que siempre será bajo la supervisión del profesor.

Finalmente uniendo ideas de otros softwares en el mercado, con un buen sustento enfocado hacia la estimulación visual, se logre obtener una herramienta sencilla pero a la vez versátil, de tal forma que se desarrollen materiales didácticos diversos y así el aprendizaje sea una diversión, un acto lúdico no rutinario, generando materiales didácticos enfocados a estímulos de reconocimiento visual, que apoyen al profesor durante el aprendizaje de la lecto-escritura de los alumnos.

Nota: hay que señalar que en la mayoría de los programas descritos anteriormente en su función pedagógica, no se encontraron detalles de su implementación como software, a excepción de Jclíc que es software libre, y se distribuye bajo los términos de la Licencia Pública General GNU (General Public License, GPL por sus siglas en inglés) utilizando un servidor de aplicaciones Tomcat y MySQL para el DBM, la tecnología que utiliza es Applets bajo el lenguaje Java y el formato para almacenar las actividades son archivos XML, utilizando recursos multimedia, la diferencia con respecto a la propuesta de solución, para este sistema es que se utilizará el *framework struts* de Java.

La Tabla 2.1 presenta un cuadro descriptivo con las características de las anteriores soluciones y las propuestas del “Sistema AMDERV-WBE”.

Los primeros cinco términos que se utilizaron para la comparación se tomaron de acuerdo a Marianne Frostig quien fue una de las primeras personas en reconocer e investigar el vínculo entre la neurología y el aprendizaje. En 1964 Frostig comenzó a promover teorías que fueron vanguardistas para su época. Propuso el concepto que “el desarrollo perceptual precede al desarrollo conceptual” e identificó una correlación estrecha entre el aprendizaje y las habilidades visuales / perceptuales. Sus teorías y herramientas de evaluación son utilizadas ampliamente para identificar y remediar los problemas de aprendizaje de los niños. Su trabajo pionero originó cientos de proyectos de investigación y algunos de esos estudios derivaron y condujeron hacia cinco facultades de la percepción visual la cuales son: coordinación visomotriz, constancia perceptual, figura a fondo, posición en el espacio y relaciones espaciales.

Los últimos cinco términos mencionados en el párrafo anterior se tomaron a petición de los profesores consultados durante la etapa de análisis, quienes hicieron hincapié en los detalles que tenían los softwares que ocupaban (idioma, ayudas de utilización, que ellos pudieran modificar) y lo que les gustaría tener para estimular a sus alumnos [73]. La descripción de cada uno de estos términos se indica a continuación.

1. **Coordinación visomotriz.** Es la capacidad de coordinar la visión con los movimientos del cuerpo o de sus partes; por ejemplo cuando, el niño ve un objeto y trata de alcanzarlo, sus manos están guiadas por la vista.
2. **Constancia perceptual.** Supone la posibilidad de percibir que un objeto posee propiedades invariables, como forma, posición y tamaño específico, el tamaño, el brillo y el color a pesar de la variabilidad de la imagen sobre la retina del ojo. Por ejemplo las estructuras bidimensionales o tridimensionales pueden ser reconocidas por el que las percibe como pertenecientes a ciertas categorías de formas, cualquiera que sea su tamaño, color, textura, modo de presentación o ángulo de visión.
3. **Figura a fondo.** Se toma como principio que percibimos con mayor claridad aquellas cosas a las que se presta atención, el cerebro humano está organizado de manera que puede seleccionar de entre un conjunto de estímulos que le llegan al alumno de un número limitado que se convierte en el centro de interés. Estos estímulos seleccionados forman la figura en nuestro campo perceptual; por ejemplo un niño que juega en un patio haciendo rebotar una pelota dirige su atención a ésta, que es la figura dentro de la escena que percibe. Otros detalles del sitio como un cajón de arena, columpios, o juguetes, no son el centro de su

interés y conforman el fondo del que se advierte de forma confusa y del que se tiene conciencia en la medida suficiente para no chocar contra ellos.

4. **Posición en el espacio.** Se define como la relación en el espacio de un objeto con el observador, la persona es el centro de su propio mundo y percibe los objetos que están al frente, delante, arriba, abajo y a un lado de sí mismo. Cuando le dices a un niño que ponga su mano derecha arriba de su cabeza está utilizando esta facultad.
5. **Relación espacial.** Es la capacidad de un observador de percibir la posición de dos o más objetos en relación con sí mismo y respecto los unos de los otros. Por ejemplo, un niño que está enhebrando cuentas debe percibir la posición de la cuenta y del hilo en relación consigo mismo, y también la de la cuenta y del hilo en relación la una con el otro.
6. **Idioma.** El lugar de origen de cada software tiene el idioma del lugar en que fue elaborado; la mayoría de los softwares se hicieron en España y otros están en idioma inglés; una de las propuestas de esta herramienta es que el profesor puede grabar los sonidos en el idioma que él este manejando.
7. **Acento.** Existen muchos países que hablan el Español, pero no tienen el mismo acento. Esto es importante porque el acento ocasiona problemas en los niños en el momento del aprendizaje de la lectura al no escuchar correctamente la pronunciación por un acento diferente.
8. **Reforzamiento.** Es una indicación audible y/o visible donde se felicita a los alumnos si se contesta una respuesta correctamente, o no.
9. **Modificable por el profesor.** Es la flexibilidad que tiene el profesor para adaptar sus materiales didácticos, cambiando los textos, imágenes y sonidos; de esta forma se logra que las secuencias expuestas de los materiales didácticos no se repitan hacia los alumnos.
10. **Interactivo.** Acción o actividad mutua de un agente (hombre-máquina) sobre o con otro (hombre-máquina), que implica a ambos en un proceso de estímulo y respuesta.
11. **Ayuda.** Programa que cuenta con instrucciones que indican el manejo del sistema.

**Tabla 2.1.** Comparación de softwares existentes y el Sistema propuesto.

TÉRMINO	Coordinación visomotriz	Constancia Perceptual	Figura a fondo	Posición en el espacio	Relación espacial	Idioma	Acento	Reforzamiento	Modificable por el profesor	Interactivo con fácil uso	Ayuda
WIN – ABC	✓	✓	•	✓	•	✓	•	✓	•	✓	•
EL MONO COCO	✓	•	✓	•	✓	✓	✓	✓	•	✓	•
JUGAR CON	✓	•	•	✓	✓	✓	•	•	•	✓	✓
ACTIVIDADES DE LECTO-ESCRITURA	✓	✓	•	•	•	✓	✓	✓	•	✓	✓
LOS COLORES	✓	✓	•	•	•	✓	•	✓	•	✓	•
EN SERIE	✓	✓	•	✓	•	✓	•	✓	•	✓	✓
LOS TAMAÑOS	✓	✓	•	•	•	✓	•	✓	•	✓	•
LAS LETRAS SUENAN	✓	✓	•	•	•	✓	•	✓	•	✓	✓
APRENDE LOS ANIMALES	✓	✓	•	•	✓	✓	•	✓	•	✓	•
FIGURAS	✓	✓	•	•	•	✓	✓	✓	•	✓	•
LOS PRIMEROS NÚMEROS POR VÍA VISUAL DIRECTA	✓	✓	•	•	•	✓	•	✓	•	✓	•
PACO EL CHATO	✓	•	•	•	•	✓	✓	✓	•	✓	•
DESBERDINTASUNAK	✓	✓	•	✓	✓	•	✓	✓	•	✓	•
A DENTRO AFUERA	✓	•	✓	✓	✓	✓	•	✓	•	✓	•
JUGANDO ACTIVO MI INTELIGENCIA	✓	✓	•	•	✓	✓	✓	✓	•	✓	•
¿CUÁL ES CUAL?	✓	✓	✓	✓	✓	✓	•	✓	•	✓	•
ANICLIC MÉTODO DE LECTURA	✓	✓	•	•	•	✓	✓	✓	•	✓	•
A GAIVOTA DE “A COVIÑA”	✓	•	•	•	•	•	✓	✓	•	✓	•
JUGANDO APRENDO		✓	✓	✓	✓	✓	✓	✓	•	✓	•
JCLIC AUTHOR	✓	✓	✓	✓	✓	✓	✓	✓	✓	•	•
SISTEMA AMDERV-WBE	✓	✓	✓	✓	✓	✓	✓	•	✓	✓	✓

**Simbología:**

- ✓ El software contiene el término.
- El software no contiene el término.

Como se puede observar en la Tabla 2.1 el sistema que se quiere desarrollar debe contener los términos señalados para obtener un software que sea una herramienta de apoyo, para el desarrollo de material didáctico en la estimulación del reconocimiento visual y de esta forma contribuir en el área de la enseñanza de la lecto-escritura.

### **Resumen**

La mayoría de los programas expuestos en este capítulo son elaborados en otros países como Chile, Perú, España, y están enfocados a estimular visualmente al niño con juegos comunes como los rompecabezas, la unión de sonidos con figuras, la distinción de un elemento (patrón) con respecto a otros, juego de parejas, series, ordenaciones, etcétera. Sin embargo, en estos juegos las actividades están establecidas dentro de los programas con la desventaja de que no se pueden cambiar de diseño, forma, color, audio, etc. Los componentes están fijos y de esta manera el niño acaba aburriéndose o aprendiendo de memoria las actividades. Por lo que, se propone un software para armar materiales didácticos que apoyen a los profesores, en el que se integre interactividad, flexibilidad, diversidad, fácil uso, muestre multimedios, utilice el idioma del profesor y le proporcione una ayuda en el manejo del sistema.

En el siguiente capítulo se describirá el diseño a utilizar para que se logre implementar este sistema.

---

---

## CAPÍTULO 3

### ANÁLISIS Y DISEÑO DEL SISTEMA

---

---

**En este capítulo se describe el análisis y diseño del sistema AMDERV-WBE, a través de una descripción de los diagramas UML y las interfaces del sistema.**

En el capítulo anterior se estableció la justificación para implementar un sistema que apoye a los profesores a desarrollar sus materiales didácticos por lo que ahora hay que establecer una metodología.

### **3.1 Metodología para el desarrollo del sistema**

En el desarrollo de un sistema se necesita de una metodología que permita planificar, estimar, analizar los requisitos, el diseño de estructuras de datos, programas, procedimientos, la codificación, las pruebas y el mantenimiento de éste, Además de, establecer un ciclo de vida del desarrollo del sistema, destacando cuatro etapas: análisis (desarrollado en el capítulo anterior), diseño, implementación y pruebas las cuales se describirán en lo subsecuente.

Para el diseño del sistema AMDERV-WBE se recurrirá a un conjunto de modelos, lenguajes y otras herramientas de apoyo para la representación de los datos en esta etapa, por lo que la metodología a utilizar es la Orientada a Objetos (Object Oriented, OO por sus siglas en inglés), porque sus principales ventajas son:

- La capacidad para modelar de forma intuitiva los requerimientos del usuario.
- La identificación y organización de los conceptos del dominio de la aplicación y no tanto la de su representación final.
- La eliminación de fronteras entre fases debido a la naturaleza interactiva del desarrollo OO.
- Una nueva forma de concebir los lenguajes de programación donde se incorporan bibliotecas de clases reutilizables.
- Hay un alto grado de iteración, lo que lleva a una forma de trabajo más dinámica.
- Se logra con facilidad dividir el sistema en varios subsistemas independientes fomentando así la creación de componentes.
- Gracias a los modelos más realistas y consistentes, se da lugar a productos más flexibles, fáciles de dar mantenimiento y reutilización.
- Su mejor adaptación a las nuevas características de las aplicaciones informáticas que hoy en día se requieren tanto para procesamiento distribuido basado en la filosofía Cliente / Servidor, como en las interfaces gráficas de usuario.

En este sentido hay que resaltar el gran esfuerzo de los autores Jim Rumbaugh, Grady Booch e Ivar Jacobson por unificar las notaciones utilizadas en los métodos orientados a objetos que han dado como resultado el Lenguaje

de Modelado Unificado (Unified Modeling Language, UML por sus siglas en inglés) [37].

Como lo indica su nombre UML es un lenguaje de modelado unificado que considera que un modelo es una simplificación de la realidad, el objetivo de este capítulo es realizar el modelo del sistema capturando las partes esenciales del mismo, para facilitar este modelado se realiza una abstracción y se plasma bajo una notación gráfica que se especificará enseguida.

## 3.2 Diagramas de UML del sistema

Los modelos gráficos de UML que se utilizarán para elaborar el modelado del sistema son:

### 3.2.1 Diagrama de casos de uso

El diagrama de casos de uso muestra la relación entre los actores (cliente) y los casos de uso (operaciones) del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

En nuestro sistema está conformado por tres actores: el PROFESOR, ALUMNO y ADMINISTRADOR.

En la Figura 3.1 se muestra el diagrama de casos de uso que representa las acciones, de forma general, que puede realizar un profesor dentro del sistema: primero inicia sesión, entra a las actividades destinadas para el profesor y finaliza sesión.

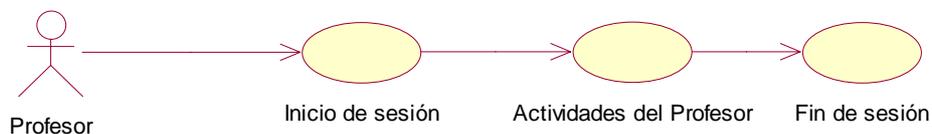


Figura 3.1 Diagrama de caso de usos para el acceso al profesor .

En la Figura 3.2 se muestra el diagrama de casos de uso que representa las acciones del alumno: al iniciar la sesión entra a las actividades que con antelación desarrollaron profesores y después finaliza la sesión.

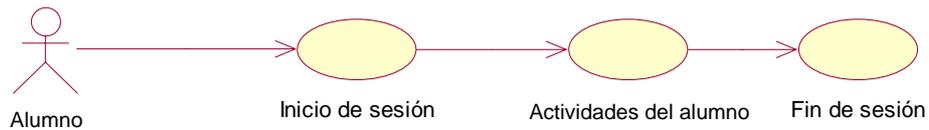


Figura 3.2 Diagrama de casos de uso para el acceso al alumno.

En la Figura 3.3 se muestra el diagrama de casos de uso para modelar de forma general las actividades del administrador en el sistema; ésta, constará de inicio de sesión, actividades del administrador y fin de sesión.

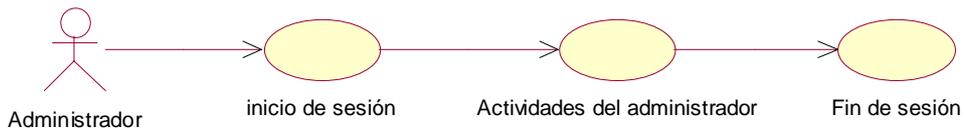


Figura 3.3 Diagrama de caso de usos para el acceso al administrador.

Ahora se procederá a describir las actividades de cada actor. La Figura 3.4 muestra el diagrama de caso de usos del profesor donde inicia sesión y sus actividades se dividen en consulta de materiales didácticos, generador de materiales didácticos, ayuda y salir de sesión.

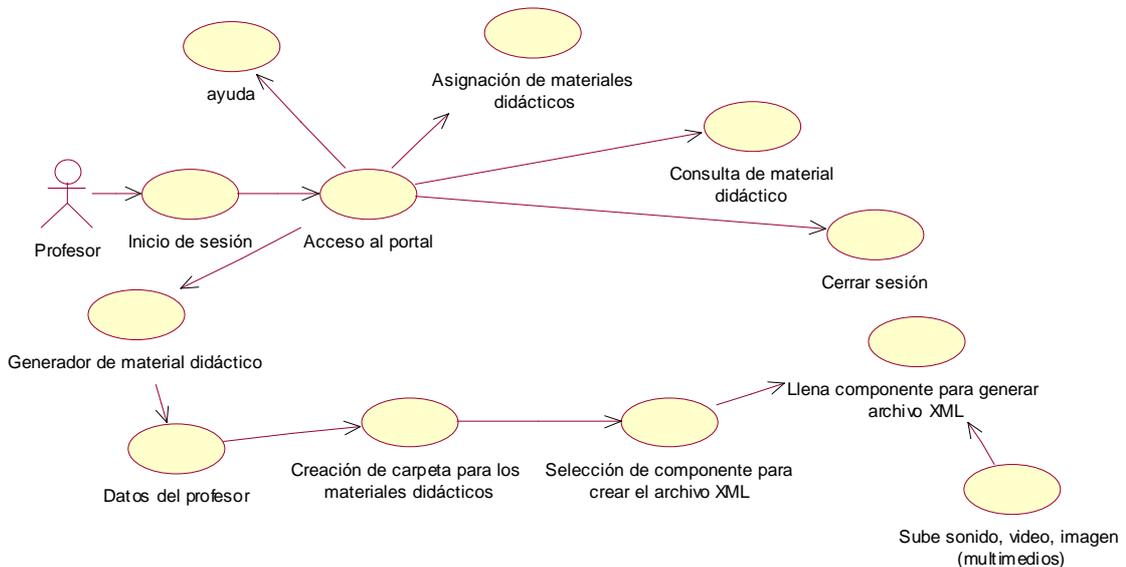


Figura 3.4 Diagrama de caso de usos de el ambiente de trabajo cuando el profesor ingresa al sistema.

En la Figura 3.5 se muestra con mayor detalle las actividades que puede realizar el alumno dentro del sistema; así el alumno inicia sesión y las actividades a realizar por el alumno al acceder al portal son: ayuda, consulta de material didáctico y cerrar sesión.

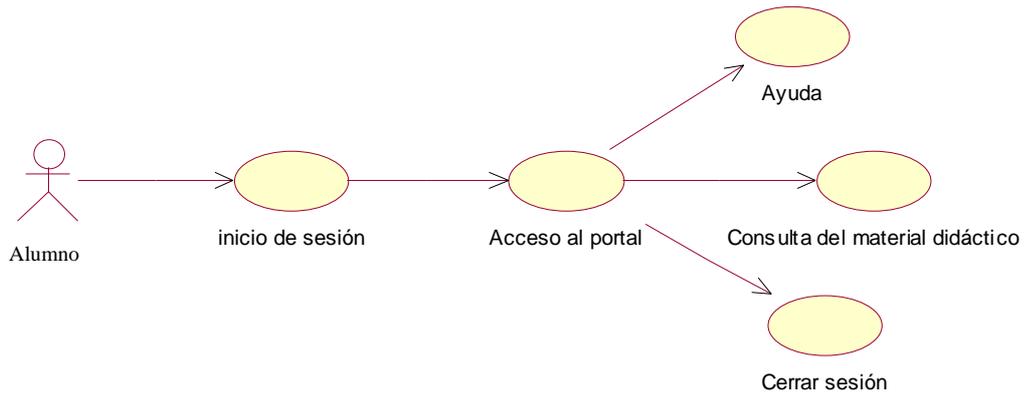


Figura 3.5 Diagrama de casos de uso del ambiente de trabajo cuando el alumno ingresa al sistema.

Y finalmente el diagrama de casos de uso que se muestra en la Figura 3.6 es el detalle de las actividades que podrá realizar el administrador: primero inicia sesión, al acceder al sistema el administrador podrá dar de alta a los usuarios introduciendo sus datos personales para su posterior reconocimiento de nombre y password que se necesitará para el acceso al sistema. Estos usuarios como ya se mencionó pueden ser profesores, alumnos o administradores, se podrá cambiar los datos de los usuarios existentes o darlos de baja (borrarlos) y cerrar sesión.

En la tabla 3.1 se describen cada caso de uso vistos en los diagramas, para tener una mayor comprensión de ellos.

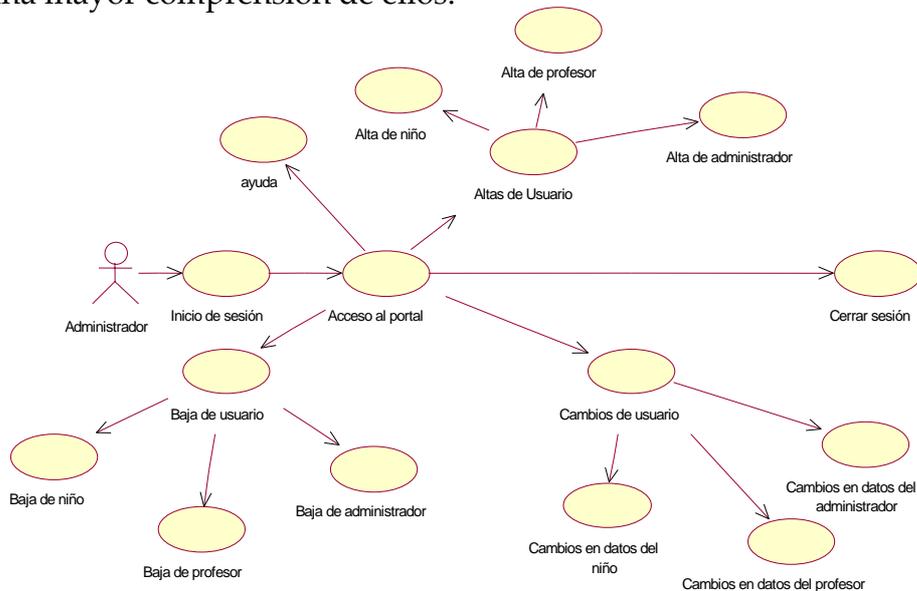


Figura 3.6 Diagrama de caso de usos de el ambiente de trabajo cuando el administrador ingresa al sistema.

**Tabla 3.1** Descripción de los casos de uso del sistema.

<p>&lt;&lt;&gt;&gt; Inicio de sesión</p> <p>Permite al usuario (alumno, profesor, administrador) identificarse para tener acceso al sistema, introduciendo nombre de usuario y contraseña.</p>
<p>&lt;&lt;&gt;&gt; Acceso al portal</p> <p>Permite al usuario la entrada al portal en donde dependiendo del inicio de sesión, tendrá acceso a las actividades correspondientes al tipo de usuario.</p>
<p>&lt;&lt;&gt;&gt; Cerrar Sesión</p> <p>Permite al usuario salir del sistema.</p>
<p>&lt;&lt;&gt;&gt; Ayuda</p> <p>Da alternativas de cómo utilizar las herramientas del portal.</p>
<p>&lt;&lt;&gt;&gt; Asignación de materiales didácticos</p> <p>Permite al usuario acceder al módulo de los materiales didácticos ya elaborados para que se proporcionen a los interesados.</p>
<p>&lt;&lt;&gt;&gt; Consulta de materiales didácticos</p> <p>Permite al usuario interactuar con el material didáctico.</p>
<p>&lt;&lt;&gt;&gt; Generador de materiales didácticos</p> <p>El profesor accede al módulo de generación de materiales didácticos, en el cual se realizan los procesos para armar los materiales didácticos y agregarlos a las carpetas.</p>
<p>&lt;&lt;&gt;&gt; Datos personales del profesor</p> <p>El profesor consulta su información personal además de poder regresar al portal a continuar con la creación de material o aplicación de un material ya elaborado.</p>
<p>&lt;&lt;&gt;&gt; Creación de la carpeta para el material didáctico</p> <p>El profesor escribe el título del material didáctico, el sistema creará la carpeta en la que estarán todos los componentes de Flash y los archivos del Lenguaje de Marcas Extensible (<i>eXtensible Markup Language</i>, XML por sus siglas en inglés), que se crean al terminar de elaborar el material didáctico, al mismo tiempo se depositarán los videos, sonidos, imágenes que el profesor le asigne al material didáctico.</p>
<p>&lt;&lt;&gt;&gt; Selección de plantilla</p> <p>El profesor selecciona el formato de la plantilla (memorama, relación de columnas, relación de sonido con imagen) que se necesita para elaborar el material didáctico, posteriormente se crea el archivo XML prototipo_materialdidactico.XML, que leerá el componente ComponentePlantilla.swf.</p>
<p>&lt;&lt;&gt;&gt; Llenar los datos del componente</p> <p>El profesor llena el componente, ubicando el contenido de éste (textos, imágenes, sonidos, etc.).</p>
<p>&lt;&lt;&gt;&gt; Envía al servidor sonido, video, imagen (multimedios)</p> <p>El profesor envía los sonidos, videos o imágenes hacia el servidor donde se colocan en la carpeta adecuada para que se utilicen cuando se requieran.</p>

<<>>Alta de alumno El administrador realiza el alta de alumnos en la base de datos del sistema.
<<>>Alta de profesor El administrador realiza el alta de profesores en la base de datos del sistema.
<<>>Alta de administrador El administrador realiza el alta de otros administradores en la base de datos del sistema.
<<>>Baja de alumno El administrador realiza la baja en la base de datos del sistema al alumno.
<<>>Baja de profesor El administrador realiza la baja de la base de datos del sistema al profesor.
<<>>Baja de administrador El administrador realiza la baja de la base de datos del sistema a otros administradores.

### 3.2.2 Diagramas de Clases

Diagrama de clases. Un diagrama de clases sirve para visualizar las relaciones entre clases involucradas del sistema, las cuales pueden ser asociativas, de herencia o de uso y se compone por clases y relaciones entre éstas.

Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones. Una clase puede representarse de forma esquemática, con los atributos y operaciones suprimidos por un rectángulo con el nombre de la clase. En la Figura 3.7 muestra el diagrama de clases del sistema.

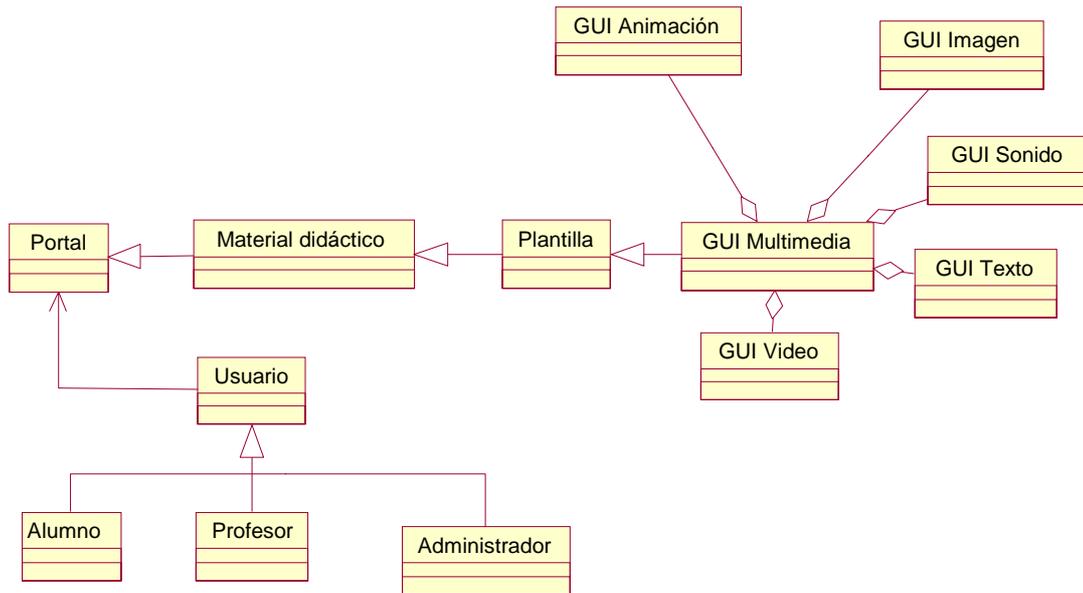


Figura 3.7. Diagrama de clases que modela el sistema.

Un objeto se representa de la misma forma que una clase pero se subraya como se puede observar en la Figura 3.8. En la parte superior aparecen el nombre del objeto junto con el nombre de la clase subrayados en este caso Jeanett:Administrador, siguiendo la siguiente sintaxis: nombre\_del\_objeto: nombre\_de\_la\_clase, con los atributos nombre y password. También puede representarse un objeto sin un nombre específico, por lo que sólo aparece el nombre de la clase.

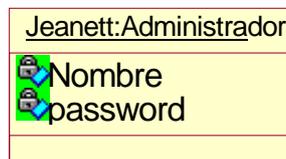


Figura 3.8. Representación de un objeto de la clase Administrador.

Las asociaciones entre dos clases se representan mediante una línea que las une. La línea puede tener una serie de elementos gráficos que expresan características particulares de la asociación. A continuación se verán los elementos gráficos más importantes.

El nombre de la asociación es opcional y se muestra como un texto que está próximo a la línea. Se puede añadir un pequeño triángulo negro sólido que indica la dirección en la cual se lee el nombre de la asociación. En el ejemplo de la Figura 3.9 se puede leer la asociación como “Usuario accede al Portal” de nuestro sistema.

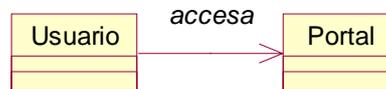


Figura 3.9. Representación de una asociación directa Usuario que accede al Portal.

Los nombres de las asociaciones se incluyen en los modelos para aumentar la legibilidad, pero a veces pueden tener demasiada información arriesgando a tener una saturación de ésta. En ese caso se puede suprimir el nombre de las asociaciones consideradas como suficientemente conocidas. En las asociaciones de tipo agregación y de herencia no se suele poner el nombre.

La agregación es la integración de varias clases en una sola, ésta clase es la que las representa. El símbolo de agregación es un diamante colocado en el extremo de la asociación, por ejemplo en la Figura 3.10, la interfaz gráfica de multimedia engloba a las interfaces gráficas de animación, imagen, sonido, texto y video.

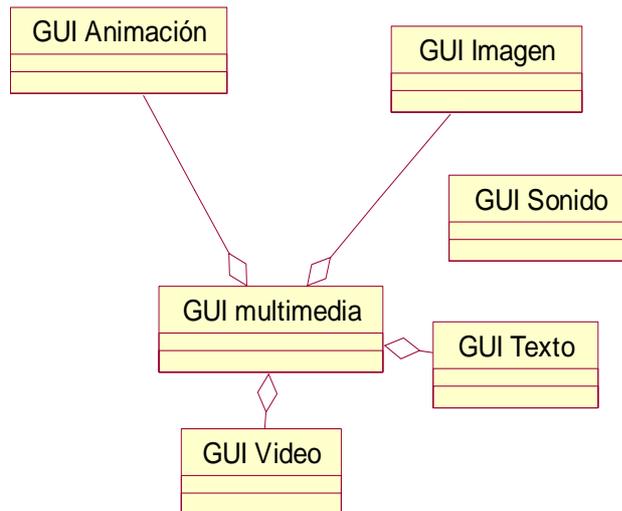
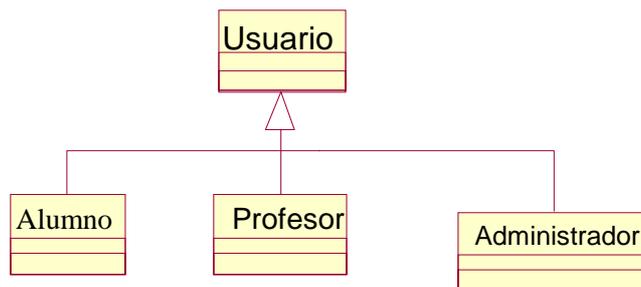


Figura 3.10 Representación de una agregación en la clase GUI multimedia.

La herencia indica que una subclase hereda las operaciones y atributos específicos de una super clase, por consiguiente la subclase además de tener sus propias operaciones y atributos, tendrá las características y atributos visibles de la super clase.

Si se tiene una relación de herencia con varias clases subordinadas, pero en un diagrama no se quieren poner todas, esto se representa mediante puntos suspensivos. En el ejemplo de la Figura 3.11, sólo aparecen en el diagrama tres tipos de usuarios de nuestro sistema; el modelo completo la clase “Usuario” tiene subclases adicionales, que son “Alumno”, “Profesor” y “Administrador”.



3.11. Representación de una súper clase Usuario que hereda a Alumno, Profesor y Administrador.

A continuación se describirán tres clases de los subsistemas más representativos que integran al sistema: Login, creación de archivos XML y Subida de archivos al sistema.

### 3.2.2.1 Diagrama de la clase subsistema Login

En la Figura 3.12 se muestra la clase LoginForm que tiene como atributos el password y userName y como métodos reset() y Validate(), es la que obtiene la información del formulario del usuario.

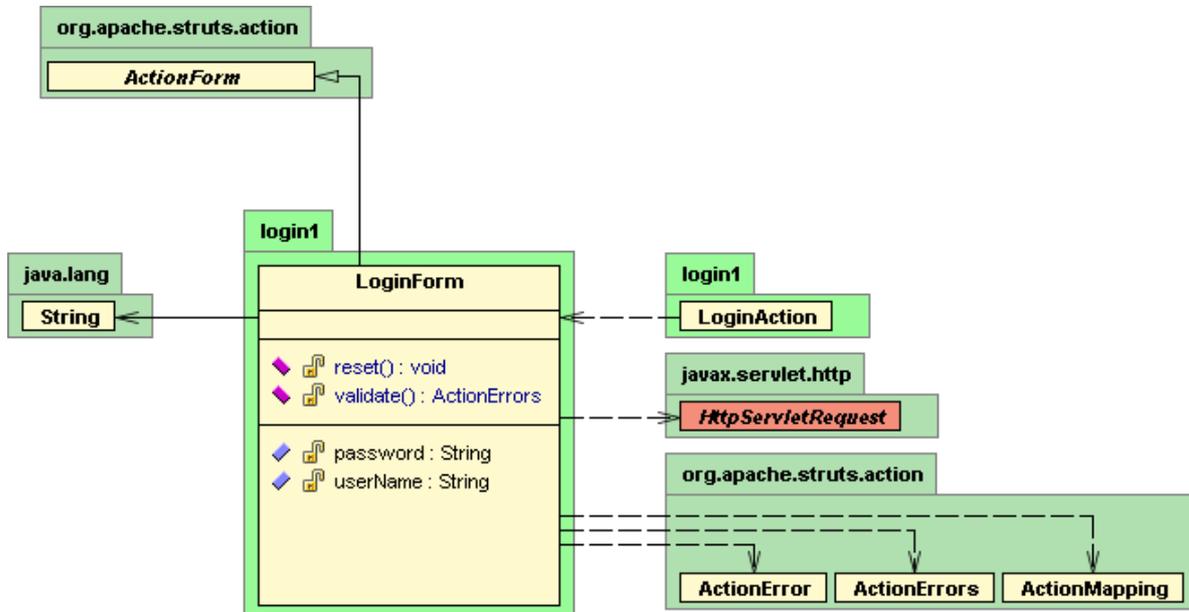


Figura 3.12. Clase LoginForm.

En la Figura 3.13 se muestra el diagrama de la clase LoginBean y es la que se encarga de obtener y enviar información a la base de datos, esta clase es un Java Bean.

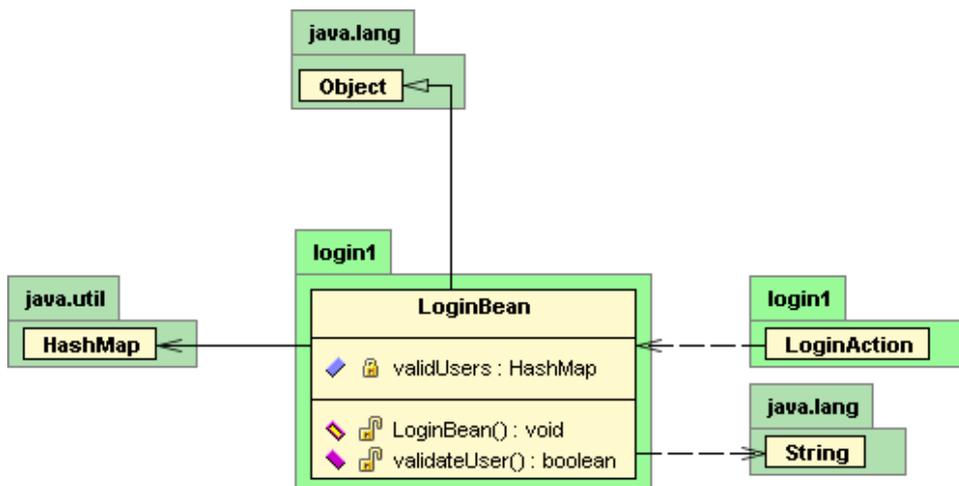
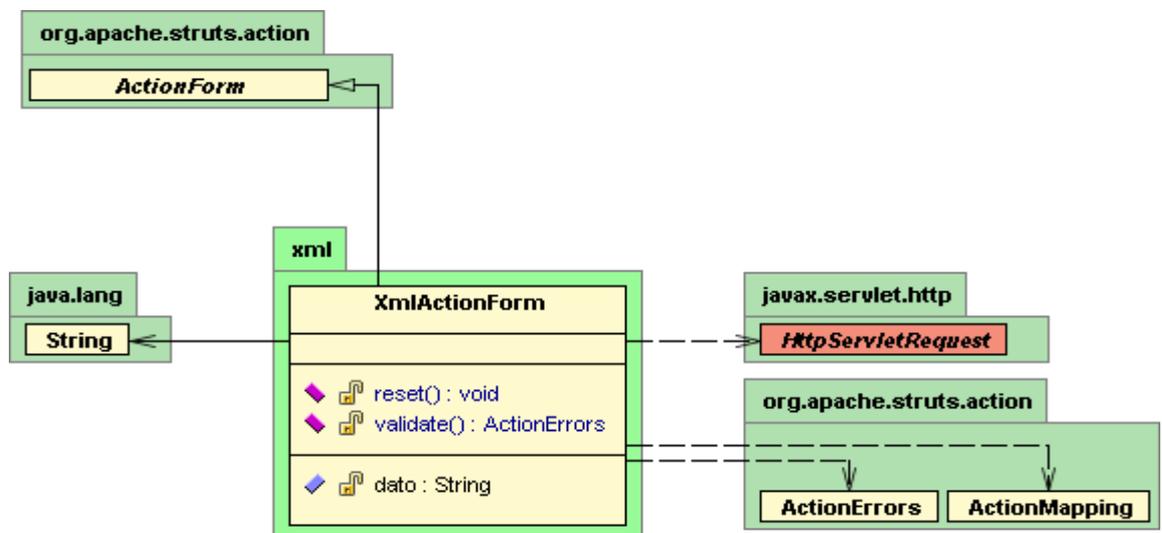


Figura 3.13. Clase LoginBean.

Esta clase es importante para lograr la validación del tipo de usuario, para que posteriormente se desplieguen las acciones que le corresponden hacer en función del tipo de usuario validado.

### 3.2.2.2 Diagrama de la clase subsistema de creación de archivos XML

También tenemos las clases encargadas de hacer la persistencia de los datos por medio de archivos XML, ya que en vez de guardar la información de los multimedia en bases de datos, ésta se guarda en los archivos XML, así en la Figura 3.14 se muestra la clase XmlActionForm con el atributo dato y los métodos reset( ) y validate( ). Se utiliza para guardar la información que proporciona el usuario para posteriormente guardarla en un archivo XML por medio de la clase XmlFile, para posteriormente ser ocupado por los componentes.



3.14. Clase XmlActionForm.

En la Figura 3.15 se muestra el diagrama de clase del archivo XmlFile que se encarga de hacer una parte de persistencia del sistema por medio de archivos XML.

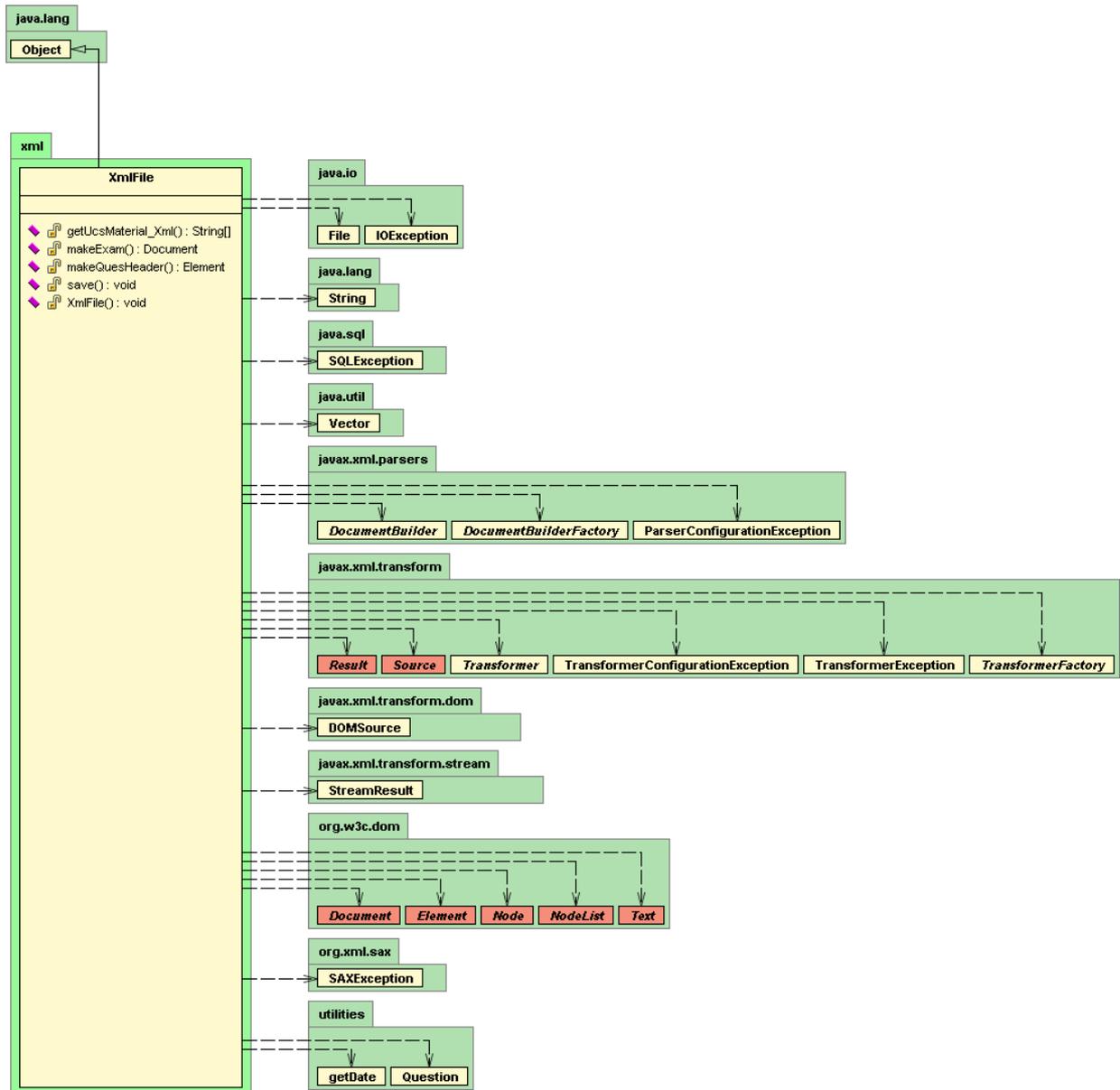


Figura 3.15. Clase XmlFile.

### 3.2.2.3 Diagrama de la clase subsistema de envío de archivos

Existe una clase que obtiene los datos de los archivos multimedia creados que se llama `StrutsUploadForm` y cuenta con un solo atributo en el que se almacena la información como nombre del archivo, directorio en los que se encuentran los archivos multimedia; se describe en la Figura 3.16.

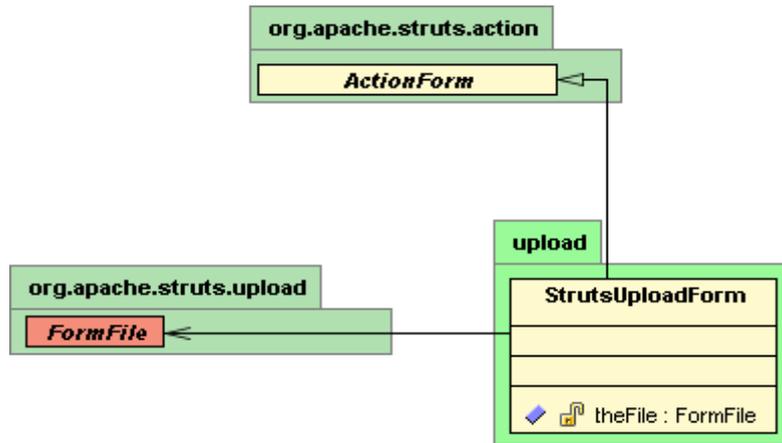


Figura 3.16. Clase StrutsUploadForm.

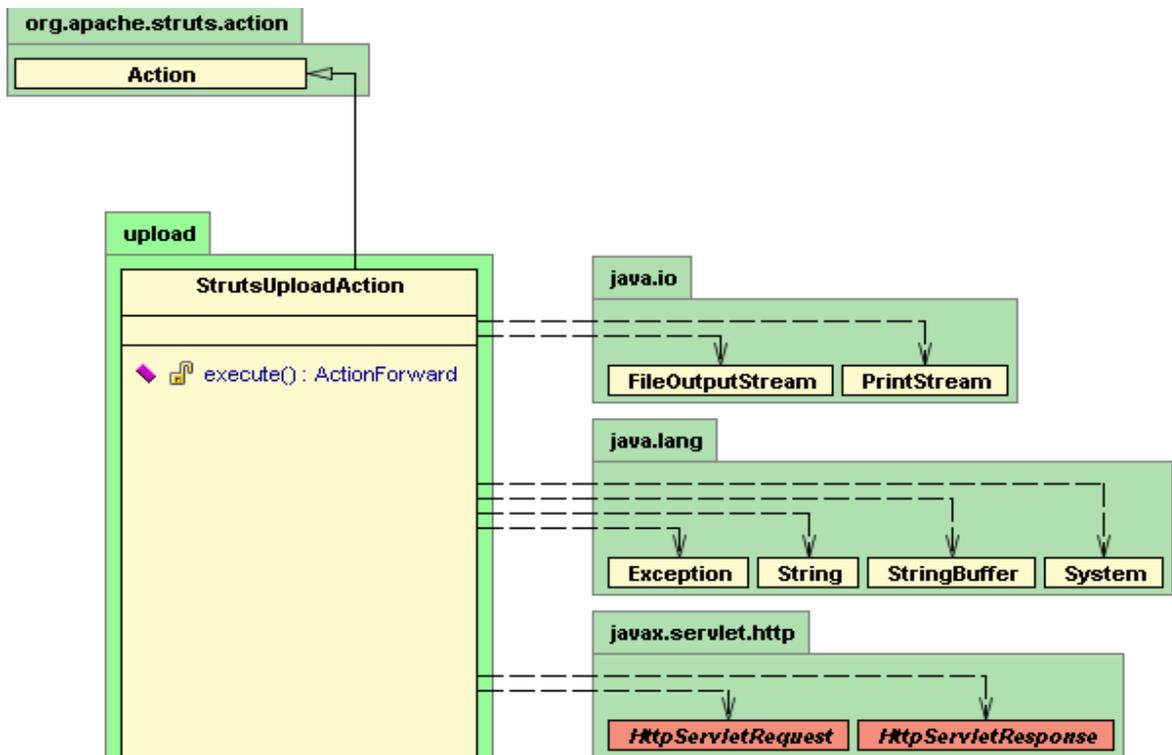


Figura 3.17. Clase del StrutsUploadAction.

Y, finalmente, la figura 3.17 muestra la clase `StrutsUploadAction` que se encarga de guardar en una carpeta determinada estos archivos multimedia y así poder ser utilizados en el sistema.

### 3.2.3 Diagramas de secuencia del sistema

Un diagrama de secuencia muestra una interacción ordenada según la secuencia temporal de eventos.

El diagrama muestra los objetos participantes en la interacción, los mensajes que intercambian y se ordenan, según su secuencia en el tiempo. En el

eje vertical se representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden establecido. Cada objeto o actor tiene una línea vertical y los mensajes se representan mediante flechas entre los distintos objetos como se aprecia en la Figura 3.18.

El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) ya sea en el margen izquierdo o junto a las transiciones a las que se refieren.

A continuación se da una breve descripción de la interacción de secuencias que se muestra en la Figura 3.4, el objeto sesión pide los datos al actor profesor, quien inicia sesión proporcionando los datos nombre y password, el sistema verifica en la base de datos si el usuario es de tipo profesor, si los datos se ingresaron de forma correcta, el usuario tiene dos alternativas: de consultar el material didáctico o de generar un nuevo material didáctico, si escoge la primera opción solo visualiza los materiales didácticos elaborados, si escoge la segunda opción el profesor crea un nuevo material didáctico, por lo que el sistema genera una nueva carpeta de material didáctico, pide catálogos de multimedia y si existen, el profesor los utiliza de tal forma que se genera el nuevo material didáctico, al término el profesor cierra su sesión y finaliza.

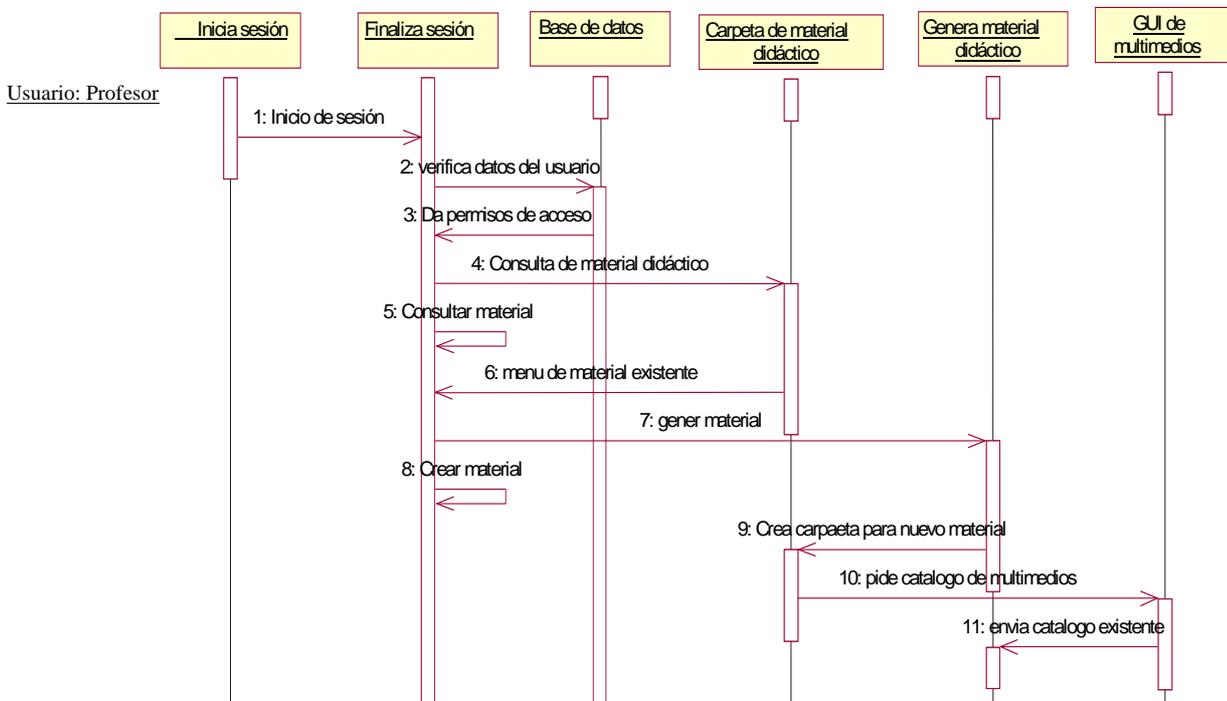


Figura 3.18. Diagrama de Secuencias del sistema.

### 3.2.4 Diagrama de Actividades

A continuación se muestra en la Figura 3.19 el diagrama de actividades; a través de éste se representará gráficamente el orden en el que se realizarán las acciones dentro del sistema AMDERV-WBE.

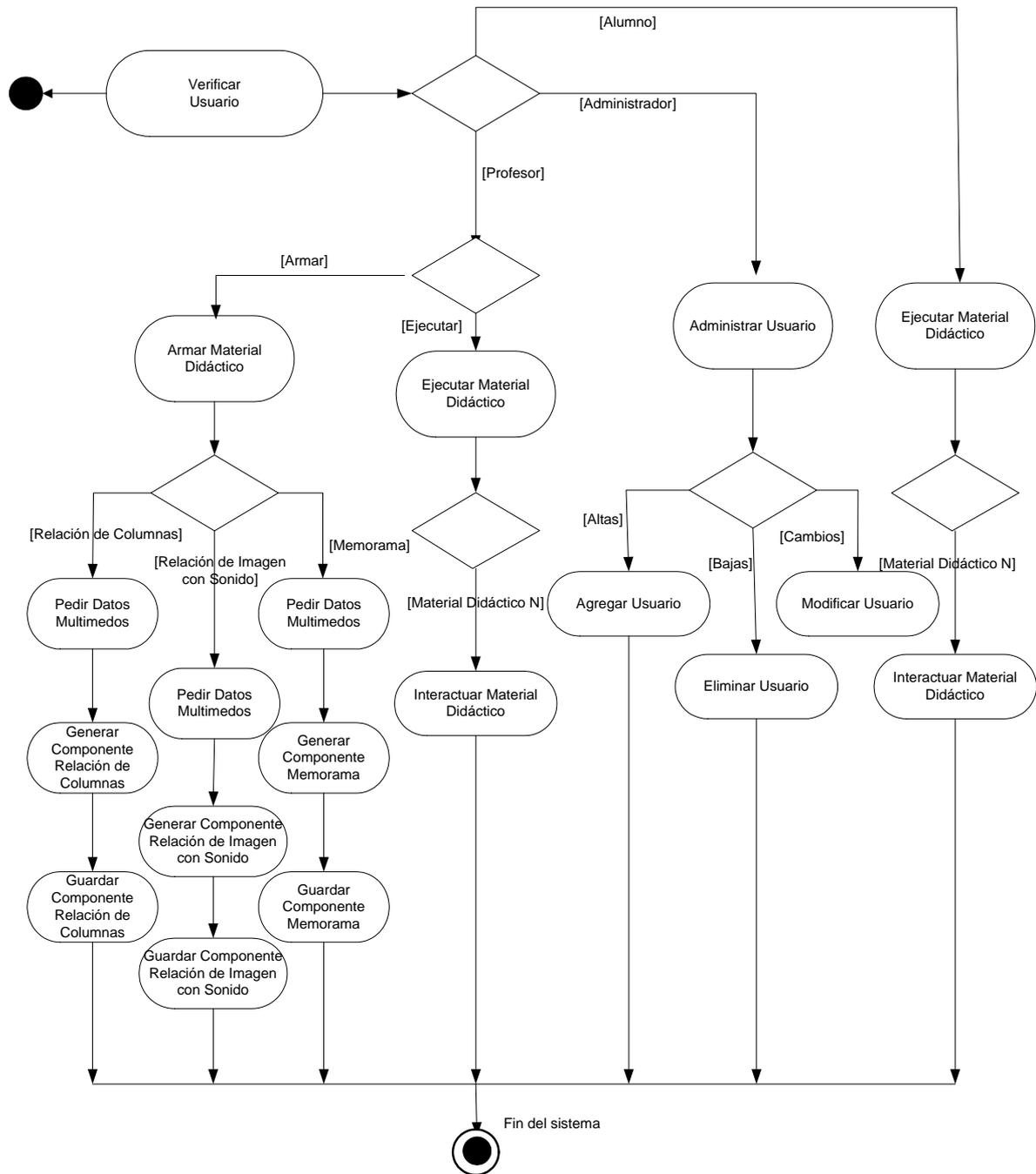


Figura 3.19. Diagrama de Actividades del sistema.

### 3.3 Patrón de diseño Modelo-Vista-Controlador

El patrón de diseño Modelo-Vista-Controlador (*Model-View-Controller*, MVC por sus siglas en inglés), es un patrón de arquitectura para desarrollo de software, que separa los datos de la aplicación, la interfaz del usuario y la lógica de control [28].

Cuando se está siguiendo el patrón MVC se debe tener tres partes bien definidas:

1. El problema a resolver. En esta parte no es determinante cómo se obtengan los datos o como se muestran, o quién es el encargado de guardarlos en un medio persistente (ya sea una base de datos, un archivo de texto, XML, registro, etc.); por ejemplo al realizar la página de acceso al sistema todas las reglas de verificación de nombre y password son independientes de cómo se va a visualizar las cajas de texto para introducir los datos en la pantalla, tal vez se quieran colocar al centro a la derecha o a la izquierda. Con estos datos se construirá el modelo donde se verifica quién es el usuario, si el password coincide con el usuario, y qué tipo de usuario para que dependiendo del tipo sean las acciones correspondientes al usuario a mostrar, o si ya se terminó la sesión por citar algunas reglas del modelo.
2. La presentación visual. Aquí se diseña la vista o interfaz con el usuario; siguiendo con el ejemplo anterior se presentarían las cajas de texto (con los textos indicando que hay que introducir el nombre del usuario, el password y el color o fondo que va a llevar la pantalla).
3. El código que toma decisiones. Esta es la parte de control ya que son los acontecimientos que afectarán a la vista, aquí se reciben los eventos de entrada (un clic, un cambio en un campo de texto, etc.).

En la Figura 3.20 se muestra la interacción entre cada una de las partes que integran a este patrón.

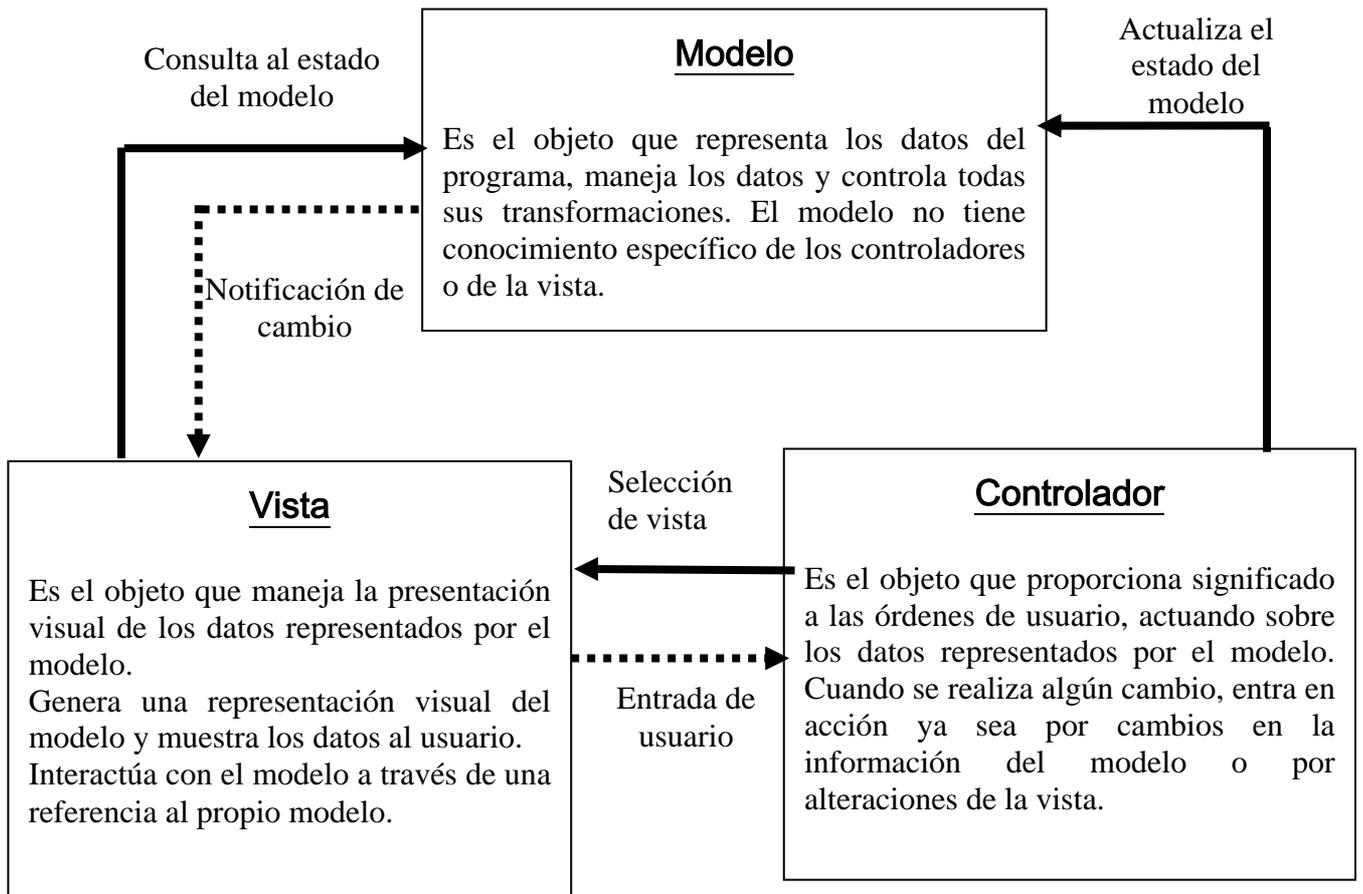


Figura 3.20. Modelo-Vista-Controlador.

Las ventajas que se desprenden al realizar el proyecto basado en el patrón MVC son:

- Mejora en la reusabilidad del código.
- El fácil mantenimiento al sistema.

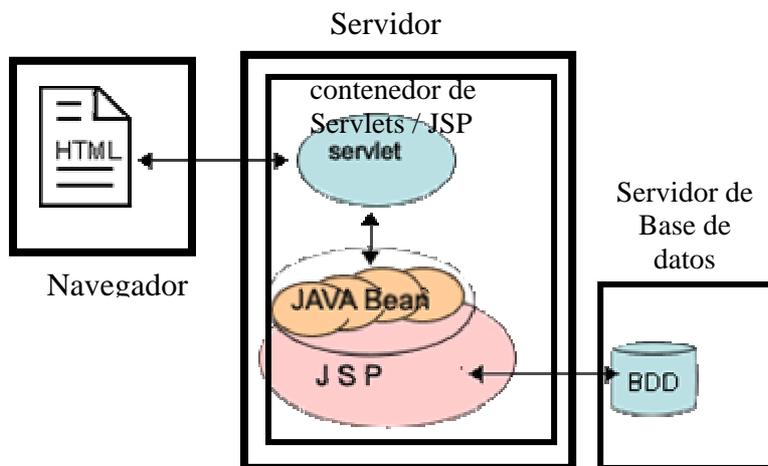


Figura 3.21. Arquitectura JavaBeans-JSPs-Servlets=Model-View-Controller.

- La separación entre la lógica de negocios y la presentación.

- Distintos diseños de presentación.
- Ahorro en el tiempo de desarrollo.
- La utilización de una tecnología ya desarrollada como JavaBeans-JSPs-Servlets, para el manejo de la complejidad del sistema como se muestra en la Figura 3.21.

MVC es un patrón del cual se han desarrollado varios *frameworks* como, por ejemplo, Apache Struts, y Java Server fases.

En este sistema se implementará un *middleware* con base en el modelo MVC en conjunto con DOM que a continuación se explica:

### 3.4 Modelo Objeto Documento

El Modelo Objeto Documento (*Document Objec, Model, DOM* por sus siglas en inglés), fue creado por *World Wide Web Consortium (W3C)*, que busca la forma de representar los elementos de un documento HTML/XML como objetos que tienen sus propios métodos y propiedades; el modelo define la estructura lógica de los documentos y la manera de acceder y manipularla. El DOM permite el acceso a la estructura de una página XML mediante el mapeo de los elementos de la misma, construyendo un árbol de nodos. La Figura 3.21 muestra una estructura típica de un código XML, código que se tomo del sistema realizado.

```
<?xml version="1.0" encoding="UTF-8"?>
<SELECCION_DE_AREA PLANTILLA="prototipo_SAs.swf">
<PREGUNTA RESPUESTA="OP1-0;OP2-0;OP3-0;OP4-0;OP5-0;OP6-1"
SONIDO="sounds/verde.mp3" TEXTO="identifica el color
verde">
<OPCION IMAGEN="images/helicopteroamarillo.jpg">a</OPCION>
<OPCION IMAGEN="images/helicopteroazul.jpg">b</OPCION>
<OPCION IMAGEN="images/helicopterogris.jpg">c</OPCION>
<OPCION IMAGEN="images/helicoptero rojo.jpg">d</OPCION>
<OPCION IMAGEN="images/helicoptero rosa.jpg">e</OPCION>
<OPCION IMAGEN="images/helicoptero verde.jpg">f</OPCION>
</PREGUNTA></SELECCION_DE_AREA>
```

Figura 3.22. Fragmento de un código XML.

DOM permite representar el código de la Figura 3.22 en una estructura de árbol como se muestra en la Figura 3.23.

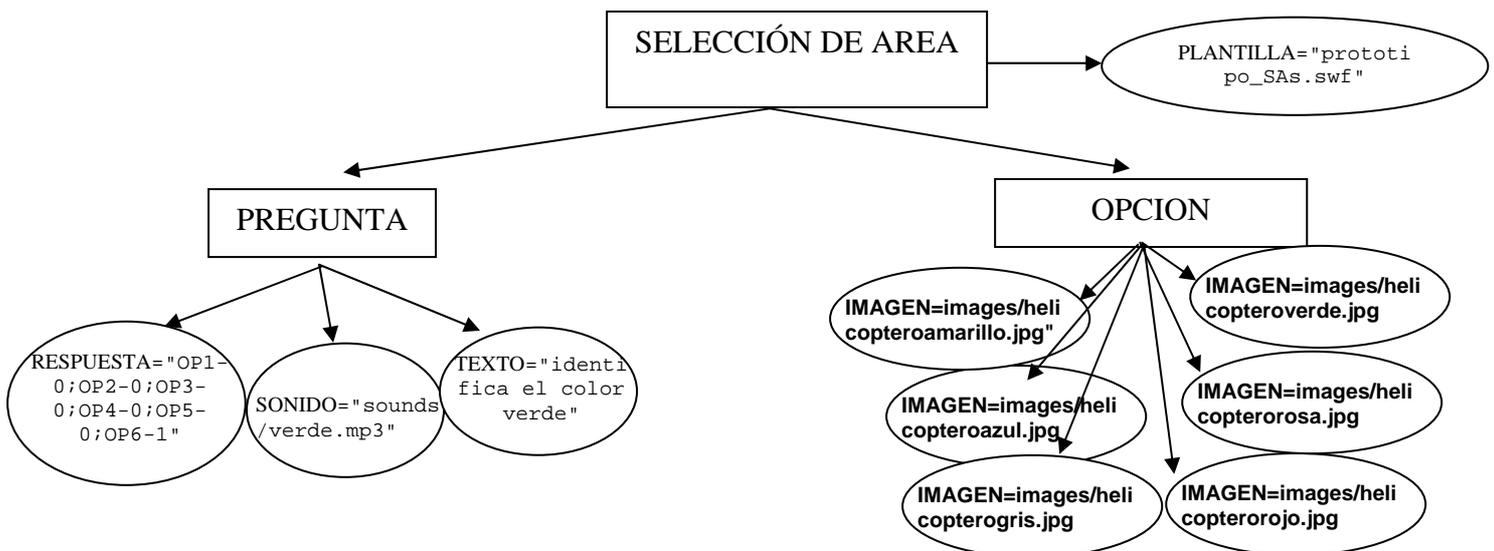


Figura 3.23. Representación del DOM para el código XML.

Antes que nada se enfatizará que DOM es una API y no un analizador (*parser*). DOM lee todo el contenido del documento en una pasada o vuelta y crea en memoria una estructura jerárquica en forma de árbol la cual representa el documento y luego puede usarse para hacer el mapeo de clases de java, creando una imagen en memoria y los datos se recorren varias veces para sacar la información para las aplicaciones.

DOM es una API independiente del lenguaje de programación tanto para lenguajes procedimentales como orientados a objetos, a través de su interfaz *node*, la cual es la interfaz base de todos los objetos dentro de un documento. Aunque no todos los nodos pueden tener nodos hijos, todos los nodos tienen métodos *appendChild( )*, *getFirstChild( )* por poner algunos de ellos. Los atributos de los objetos se pueden obtener con el método *getTagName( )*.

Por lo tanto, nuestros componentes harán uso de los archivos XML representados elementos del documento utilizando a DOM.

A continuación describiremos que es un componente.

### 3.5 Componentes de Software

En la industria electrónica, como en otras industrias, se acostumbra utilizar componentes para construir placas, tarjetas, etc. En el campo del software la idea es la misma. Se puede crear un interfaz de usuario en un programa Java en base a componentes: paneles, botones, etiquetas, caja de listas, barras de desplazamiento, diálogos, menús, etc., en un intento por conseguir una mayor interoperabilidad y eficiencia, los desarrolladores de aplicaciones han establecido reglas de procedimiento, programación orientada a objetos y bibliotecas de software reusable. Normalmente y si pueden evitarlo, los desarrolladores rara vez crean aplicaciones a partir de cero, sino que intentan aprovechar la infraestructura de hardware, software y herramientas ya existentes, así como componentes de software previamente creados, todo ello para controlar el tiempo y el costo del desarrollo y del despliegue de las aplicaciones. Aunque estos métodos adoptan nombres diferentes, se designan normalmente como "*componentes de software*".

Por lo tanto la definición de Componente de Software que se va a utilizar para esta tesis es la de Szyperski que dice lo siguiente: "*Un Componente de Software es una unidad de composición con interfaces especificadas contractualmente y dependencias explícitas de contexto. Un componente de Softwares puede ser entregado de forma independiente y es sujeto de composición por terceras partes*" [35].

### Los Beneficios del Desarrollo de Software basado en Componentes:

Con los componentes se logra alcanzar un mayor nivel de reutilización del software en tiempos de desarrollo más cortos, ya que se pueden readaptar y moldear para otra tarea distinta gracias a sus interfaces y su construcción a nivel binario característica que vuelve independiente al código fuente y por lo tanto puede ejecutarse sobre distintos sistemas sin necesidad de recompilarlo, haciendo más flexibles a los componentes y logrando manejar el reto al cambio con mayores ventajas [74].

Simplifica las pruebas porque son ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.

Simplifica el mantenimiento del sistema ya que existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.

Una mayor calidad dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

A continuación se muestra el diseño de los componentes a construir para nuestro sistema, porque un aspecto importante del sistema es generar un conjunto de componentes de software multimedia con las ventajas antes mencionadas. En la Figura 3.24 se muestra el modelo general en donde se encuentra el componente que obtiene los datos de los archivos multimedia (texto, imágenes, sonidos, etc.) a utilizar desde un archivo XML que permite reconfigurar dinámicamente a este componente y se ocupará otro archivo XML para obtener la secuencia en la que se muestran éstos.

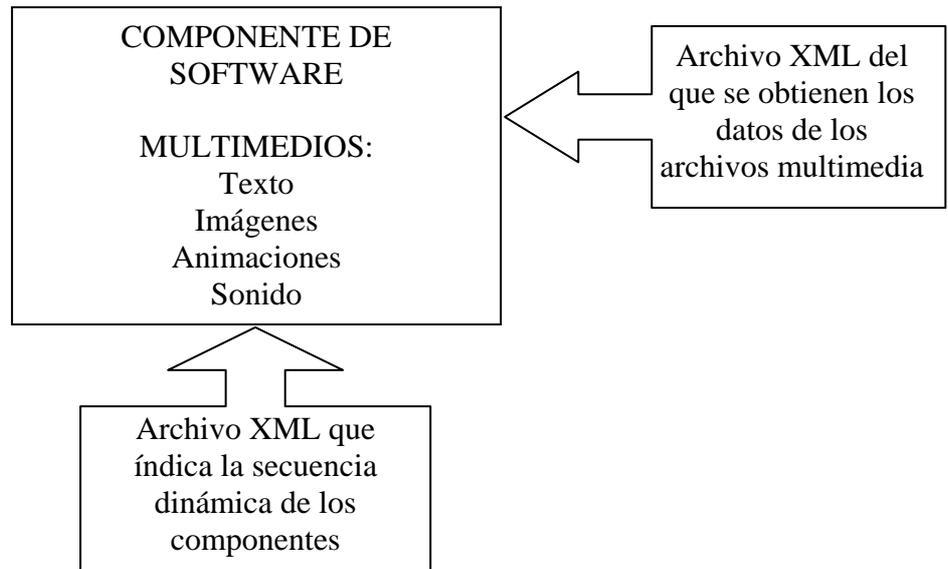


Figura 3.24. Modelo general de los componentes de software y archivos XML utilizados.

### 3.5.1 Diseño de la interfaz de los componentes de software

A continuación se describirá el diseño de los componentes de software, estableciendo el vínculo visual de la interacción del sistema con el usuario; los componentes que se construirán son tres: un memorama, uno de relación de columnas y uno de relación de imagen con sonido.

La interfaz del componente memorama mostrará al usuario pares de imágenes distribuidas aleatoriamente, como se muestra en la Figura 3.25.

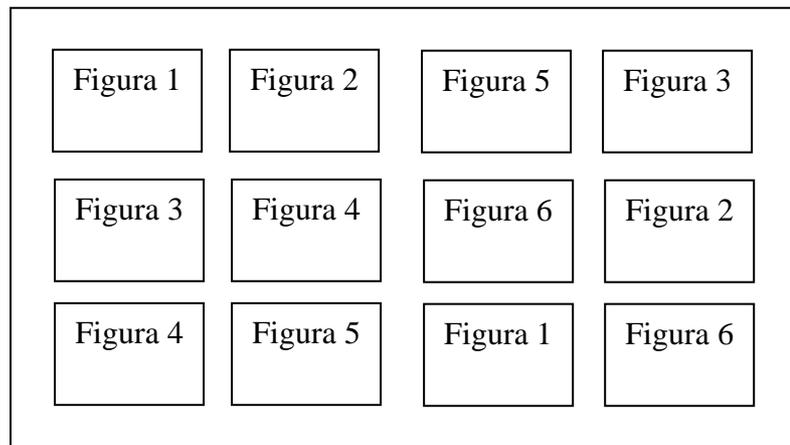


Figura 3.25. Diseño de la interfaz del componente memorama.

Los datos nombre y ruta de ubicación de las imágenes que se visualizarán en la interfaz de la Figura 3.25, se obtendrán a través de la interfaz

de la Figura 3.26 que esta conformada por seis cajas de textos, con botones que servirán de ayuda para utilizar el explorador de Windows y localizar las imágenes, para que en la caja de texto de la imagen queden los datos de estas, a un lado hay una caja de texto para el título.

<b>Memorama</b>			
Imagen 1:	<input type="text"/>	Examinar	título <input type="text"/>
Imagen 2:	<input type="text"/>	Examinar	título <input type="text"/>
Imagen 3:	<input type="text"/>	Examinar	título <input type="text"/>
Imagen 4:	<input type="text"/>	Examinar	título <input type="text"/>
Imagen 5:	<input type="text"/>	Examinar	título <input type="text"/>
Imagen 6:	<input type="text"/>	Examinar	título <input type="text"/>

Figura 3.26. Diseño de la interfaz de obtención de datos para el componente memorama.

La interfaz del componente relación de columnas, se muestra en la Figura 3.27 en la que se distribuirán en dos columnas de cuatro imágenes; las imágenes de la derecha deben tener relación con las figuras de la izquierda.

Acciones a realizar	
Imagen 1	Imagen 2
Imagen 2	Imagen 4
Imagen 3	Imagen 3
Imagen 4	Imagen 1

Figura 3.27. Diseño de la interfaz del componente relación de columna.

Los datos de las imágenes se deberán obtener a través de la interfaz que se ve en la Figura 3.28; se requerirá llenar una caja de texto para indicar las acciones a realizar, ocho cajas de texto que guardarán la dirección y el nombre

de la imagen a mostrar, acompañadas de botones que auxiliarán para mostrar el explorador de Windows y poder seleccionar los datos de la imagen; además con un título que las describirá.

## Relación de Columnas

Introduzca la pregunta para esta relación de columnas

Imágenes de la Columna de la Izquierda

Imagen 1:	<input type="text"/>	Examinar	título	<input type="text"/>
Imagen 2:	<input type="text"/>	Examinar	título	<input type="text"/>
Imagen 3:	<input type="text"/>	Examinar	título	<input type="text"/>
Imagen 4:	<input type="text"/>	Examinar	título	<input type="text"/>

Imágenes de la Columna de la Derecha

Imagen 5:	<input type="text"/>	Examinar	título	<input type="text"/>
Imagen 6:	<input type="text"/>	Examinar	título	<input type="text"/>
Imagen 5:	<input type="text"/>	Examinar	título	<input type="text"/>
Imagen 6:	<input type="text"/>	Examinar	título	<input type="text"/>

Figura 3.28. Diseño de la interfaz de obtención de datos para el componente relación de columnas.

El diseño de la interfaz del componente relación de imagen con sonido se muestra en la Figura 3.29, en la que se distribuirán seis imágenes, de las que hay que seleccionar una que tenga las características que indiquen las instrucciones a realizar, apoyado de un sonido que, también, podrá ser ingresado por el usuario.

Instrucciones a realizar

Figura 1	Figura	Figura 3
Figura 4	Figura 5	Figura 6

Figura 3.29. Diseño de la interfaz del componente relación de imagen con sonido.

La Figura 3.30 describe el diseño de la interfaz que se utilizará para obtener los datos para el componente relación de imagen con sonido, en donde

hay una caja de texto para poner la pregunta (las instrucciones a realizar), una caja de texto para poner los datos del archivo de sonido, acompañada de un botón para ayudar con el explorador de Windows a la selección de éste, seis cajas de texto para colocar los datos de las imágenes, auxiliadas de un botón para abrir el explorador de Windows y seleccionar los datos de la imagen, con otra caja de texto a un lado para agregar un título a la imagen y finalmente seis botones de radio para indicar cual es la imagen que tiene la respuesta a las instrucciones dadas.

## Relación de imagen con sonido

Texto de la pregunta

Archivo de sonido:

Imagen 1:	<input type="text"/>	<input type="button" value="Examinar"/>	título	<input type="text"/>
Imagen 2:	<input type="text"/>	<input type="button" value="Examinar"/>	título	<input type="text"/>
Imagen 3:	<input type="text"/>	<input type="button" value="Examinar"/>	título	<input type="text"/>
Imagen 4:	<input type="text"/>	<input type="button" value="Examinar"/>	título	<input type="text"/>
Imagen 5:	<input type="text"/>	<input type="button" value="Examinar"/>	título	<input type="text"/>
Imagen 6:	<input type="text"/>	<input type="button" value="Examinar"/>	título	<input type="text"/>

Seleccione la imagen de la contestación 
  Imagen1 
  Imagen2 
  Imagen3 
  Imagen4 
  Imagen5 
  Imagen6

Figura 3.30. Diseño de la interfaz de obtención de datos para el componente relación de imagen con sonido.

## Resumen

En este capítulo se definió y utilizó el enfoque orientado a objetos para resolver el diseño de desarrollo del sistema AMDERV-WBE, implementado las especificaciones con la herramienta UML que da la posibilidad de hacer los diagramas como los de caso de uso, clases, secuencias y actividades que detallan el diseño de este sistema, también se hace referencia al patrón MVC por sus características de separar la interfaz, el control y el modelo que permiten la simplificación en el desarrollo de aplicaciones *Web*. Se explica el modelo DOM, pieza importante para generar los archivos XML para guardar la información que mostrarán los componentes; por ello se da una definición y sus ventajas. Este capítulo sirve de introducción a los conceptos que se manejarán en el capítulo 4, en el que describirá la implementación de nuestro sistema.

---

---

## CAPÍTULO 4

### IMPLEMENTACIÓN DEL SISTEMA

---

---

**La implementación de cualquier sistema es la codificación del diseño en un lenguaje de programación y como nuestro sistema está basado en una arquitectura Cliente / Servidor, se explica todo lo que conlleva para poder implementarlo, también se describirán los subsistemas que integran al sistema AMDERV-WBE y se dará una breve descripción de los componentes que se desarrollaron.**

## 4.1 Descripción del funcionamiento del sistema AMDERV-WBE

El sistema se utilizará a través de un navegador *Web*, en el que un usuario hará una petición al servidor, dando los datos de autenticación, estos datos serán escritos en una vista que desplegará en una JSP y serán enviados y administrados por un *servlet* auxiliado por un *Java Bean* para acceder a la base de datos y verificar esta información; posteriormente, el *Java Bean*, devuelve los datos después de la verificación y el *servlet* nuevamente a través de una JSP muestra el ambiente según sea el caso: para el profesor, alumno, administrador y usuario no existente.

El profesor accederá a través de la JSP a realizar las acciones de: armar materiales didácticos o utilizarlos.

- Si accesa al armado de materiales didácticos se mostrará mediante una JSP un formulario para guardar el nombre del material a armar, luego con los *struts* y *servlets* se coordinarán la elaboración de archivos persistentes, que guardaran la información que utilizaran los componentes, a su vez la copia de estos componentes, para crear un directorio con todos éstos archivos, y así, quede guardado para su posterior utilización.
- Si accesa a utilizarlos se mostrará una JSP con el listado de todos los materiales didácticos elaborados y bastará un clic para empezar a interactuar con el material didáctico elegido.

El Alumno podrá acceder a través de una JSP que listara los materiales didácticos, previamente armados por el profesor, para que pueda utilizarlos donde las tareas las resolverá el *framework* de *struts*.

El administrador accederá a través de una JSP para que *servlets* y *Java Beans* proporcionen las JSPs con los datos para dar de alta, bajas o cambios en los datos de los usuarios.

A continuación se describirá la arquitectura Cliente / Servidor, las tareas realizadas por las JSPs, *servlets*, *struts* y modelos para el desarrollo de la aplicación *Web*.

### 4.1.1 Arquitectura Cliente / Servidor

La arquitectura Cliente / Servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes; por lo tanto, la

arquitectura Cliente / Servidor divide una aplicación en tres componentes básicos: un cliente, un servidor y la red.

En este modelo las aplicaciones se dividen de tal forma que el servidor contiene la parte que debe ser compartida por varios usuarios y en el cliente permanece sólo lo particular de cada usuario.

Las funciones de los clientes son: manejo de la interfaz del usuario. No depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo, captura y validación de los datos de entrada, generación de consultas e informes sobre las bases de datos.

Las funciones de los servidores son: presenta a todos sus clientes una interfaz única y bien definida, gestión de periféricos compartidos, control de accesos concurrentes a bases de datos compartidas y enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos equipos. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en equipos de alto rendimiento [34].

Ahora haciendo uso de esta arquitectura el objetivo es hacer posible la entrega de contenido educativo por medio de la computadora hacia el estudiante a través de la *Web* y a los profesores darles la posibilidad de crear material educativo mediante un navegador. El navegador funciona como cliente para estudiantes y profesores, y el servidor atiende las peticiones de ellos como se observa en la Figura 4.1

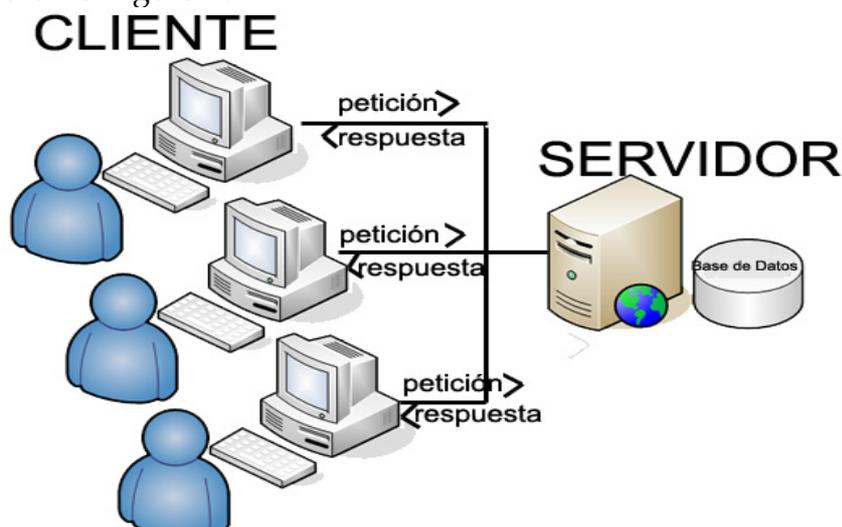


Figura 4.1. Arquitectura Cliente / Servidor.

El cliente suele estar en una computadora diferente a la del servidor y se comunican entre sí a través de protocolos de comunicación como HTTP que es un protocolo de petición/respuesta, donde un navegador *Web* solicita un documento y el servidor *Web* devuelve el documento, todo ello ocurre mediante un flujo de datos en HTML o Páginas de Servidor Java (*Java Server Pages*, JSP por sus siglas en inglés). Además, el navegador envía parámetros por medio del Localizador de Recursos Uniforme (*Uniform resource locator*, URL por sus siglas en inglés) utilizando el método “get” o “post” así el servidor *Web* invoca a un programa que responde a ciertos tipos de peticiones, a estos programas en Java se les denomina “*servlets*”, estos son clases Java que implementa el protocolo http.

#### 4.1.2 Java Server Page (JSP)

Las JSPs están compuestas de código HTML/XML y en su contenido se encuentran etiquetas especiales, con código incrustado Java teniendo como ventaja que una JSP puede procesar la petición incluyendo en ella misma la lógica de negocio, pero su real potencial es utilizar a las JSPs de forma separada a la parte del diseño con respecto al de la codificación. Un *servlet* generalmente es utilizado para procesar formas (requisiciones de usuarios), y generar contenido dinámico de manera similar a como lo hace un JSP, la diferencia estriba en que un *servlet* solo contiene lenguaje Java desde el inicio, mientras que un JSP contiene código incrustado Java y HTML/XML [33].

#### 4.1.3 Servlets

Los *servlets* permiten generar contenido dinámico y corren del lado del servidor. Los *servlets* son los programas encargados de enviar los parámetros y cabeceras de respuesta que, son variables de entorno, mientras el servidor ejecuta la tarea pedida por la aplicación. Una de las tareas más comunes es acceder a una base de datos y obtener una respuesta de forma dinámica, la respuesta se envía al navegador *Web* cliente que solicitó el servicio. Dentro de nuestro sistema se implementan para la verificación de datos al iniciar sesión.

Anteriormente se mencionó de que los *servlets* son clases Java que amplían la funcionalidad de un servidor *Web* generando páginas dinámicas en un motor de *servlets* (entorno de ejecución para administrar la carga y descarga de *servlets* dirigiendo las peticiones a los *servlets* y enviando las respuestas a los clientes). Los *servlets* son clases Java que implementan el protocolo HTTP y su ciclo de vida se puede observar en la Figura 4.2. En ella se muestra un servidor que se encarga de inicializar el *servlet* donde se inicia el método `init()`, el *servlet* mediante `service()` maneja las peticiones del cliente, para que finalmente el servidor elimine el *servlet*. Los *servlets* se ejecutan hasta que el servidor los

destruye, ya sea, al darse de baja el servidor, o dando de baja un *servlet* por medio del administrador del sistema, en ambos casos se ejecuta el método *destroy()* del propio *servlet*.

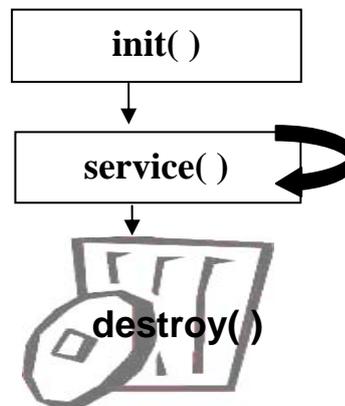


Figura 4.2. Ciclo de vida del *servlet*.

#### 4.1.4 Java Bean

Un Java *Bean* es una arquitectura que permite reutilizar componentes de software que pueden ser manipulados como herramientas de desarrollo ("*Builder Tools*"). Estos "*Beans*" pueden ser tan sencillos como un botón o complejos como el acceso a una base de datos; una característica de un Java *Bean* son los métodos (funciones) *get/set* encargados de obtener los atributos o asignarle valores a estos. Los Java *Beans* permiten agrupar funcionalidades para conformar una aplicación; por ejemplo, un Java *Bean* donde se agrupe la información personal, o los datos de un inventario, o bien los datos de una factura, etc., sin embargo, estos deben ser integrados con otros componentes para que realicen su función. En nuestro caso los Java *Beans* los utilizamos para la interacción con la base de datos estableciendo un conjunto de métodos que nos permitan actualizar, agregar, borrar información y obtener los datos del usuario.

#### 4.1.5 Struts

Un *struts* es una estructura que implementa el patrón MVC (Modelo-Vista-Controlador), a esta estructura se le llama *Framework*, el cual es un conjunto de objetos para simplificar el desarrollo de aplicaciones, que se relacionan directamente en un dominio y está constituido por clases, métodos abstractos que deben ser implementados. El *framework* tiene un *kernel* que constituye el núcleo básico que es reutilizable y al que se le hacen pocas modificaciones. La capacidad de reutilización del código y del diseño del *framework* permite una mayor productividad y simplifica los aspectos de desarrollo de la aplicación, en comparación con el desarrollo tradicional de los sistemas de software [24]. El desarrollo de *frameworks* ha sido exitoso en muchos dominios. Algunos ejemplos incluyen: *Microsoft Foundation Classes* (MFC), *Object Management Group* (OMG) [25], COM+ y DCOM de Microsoft [26].

Mediante el *framework* llamado *struts* se implementa el patrón MVC, gracias a las ventajas de ser concreto en la parte del controlador, agregando las conexiones entre el controlador y el modelo, y entre el controlador y la vista, independientemente de la vista particular. Concluyendo, las JSPs se utilizan para presentar información en el navegador del usuario. Los *Beans* guardan los resultados en el modelo. Los *servlets* coordinan y controlan la ejecución de los *Beans*. Este *framework* se utiliza para el desarrollo de aplicaciones *Web*.

A continuación se mencionan dos modelos, utilizados en el desarrollo de aplicaciones *Web*: JSPs - *Beans* y *Servlet* - JSPs - *Beans*, en nuestro sistema se utilizará el segundo, porque con el apoyo del *struts Framework*, se tendrá una mejor administración en el control de atención a las peticiones de cada componente de la aplicación.

#### 4.1.6 Modelos para el desarrollo de aplicaciones Web

- Model-1: JSPs para presentación y control y Java *Beans* para la lógica, primero a través de un navegador utilizando JSPs se hace la petición al servidor, después utilizando el Java *Bean* se obtiene y valida los datos, en ocasiones utilizando la base de datos, para posteriormente enviar una respuesta al navegador, como se ve en la Figura 4.3.

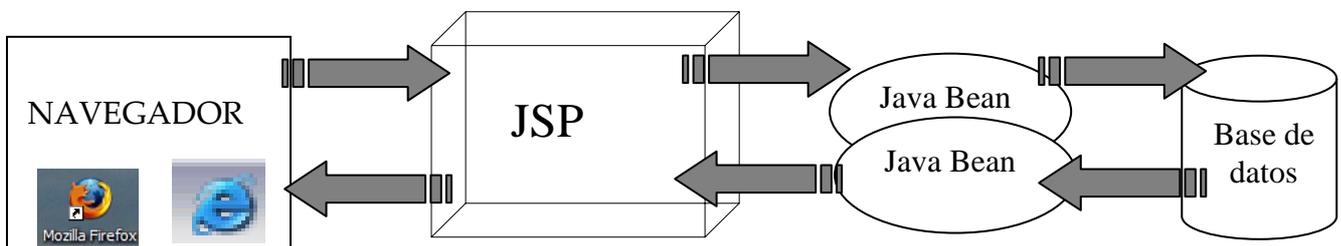


Figura 4.3. Diagrama de la arquitectura Java modelo 1.

- Model-2: maneja el patrón MVC, utiliza las JSPs para presentación, los *servlets* para el control y Java *Beans* para la lógica. A través de una JSP el navegador hace la petición al servidor, donde el *Servlet* manda la acción al Java *Bean*, éste obtiene y valida los datos accedendo a la base de datos, enviando una respuesta y a su vez el *Servlet* envía la respuesta a través de las JSPs para que muestre los resultados en el navegador, como se ve en la Figura 4.4, se vuelve a reiterar que es el modelo que se ocupará para la implementación de nuestro sistema.

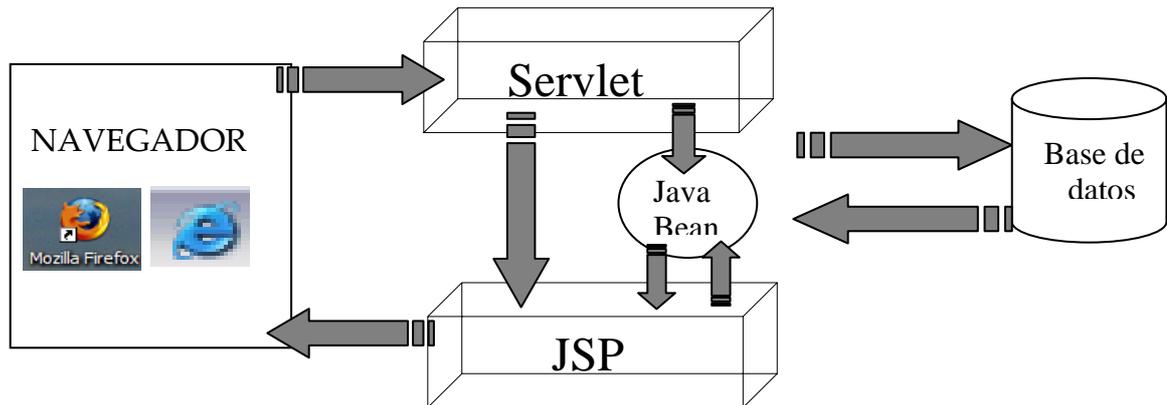


Figura 4.4. Diagrama de la arquitectura java modelo 2.

En la Figura 4.5 se describe en términos gráficos la arquitectura de nuestro sistema.

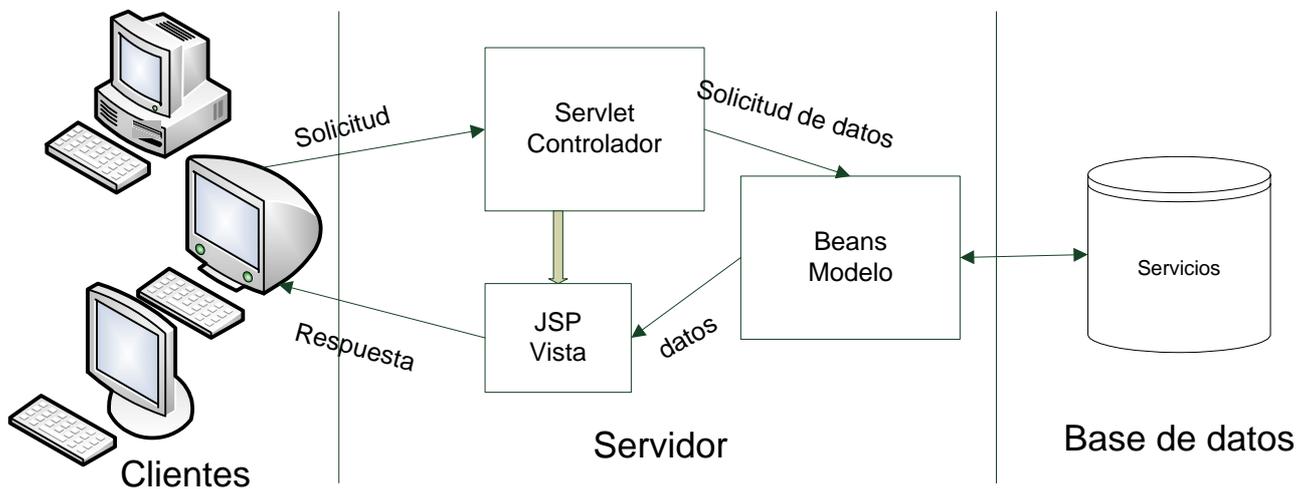


Figura 4.5. Arquitectura del Sistema AMDER-WBE

## 4.2 Implementación del controlador

Con *Struts* se implementa, de forma transparente para el programador, el controlador dentro del patrón MVC, así como la forma de comunicarse entre las tres capas del patrón MVC; La capa de la vista es la encargada de todas las peticiones del cliente, desde un formulario que es generado por la JSP, con la ayuda de la biblioteca de etiquetas HTML y de *struts*, las peticiones del cliente se transmiten a un *servlet* denominado *ActionServlet*, el que actúa como controlador, estas indicaciones se configuran en el archivo *struts-config.xml* con una serie de objetos *ActionMapping*, encargados de analizar las direcciones URL y la solicitud realizada por el diseñador de la aplicación; el *ActionServlet* obtiene una clase *RequestProcessor*, la que envía los datos al *ActionForm* que es un Java

*bean*, el *bean* se encarga de encapsular los datos, estos datos en ocasiones se validan en una base de datos, representando así al modelo. Posteriormente la clase *Action*, es el manejador específico de la aplicación para procesar la solicitud, ya sea que existan errores, se crean instancias de objetos error, o la respuesta sea correcta y se crea el objeto *ActionForward*, el objeto a través de la JSP muestra el resultado en la capa de la vista, como se puede observar en la Figura 4.6.

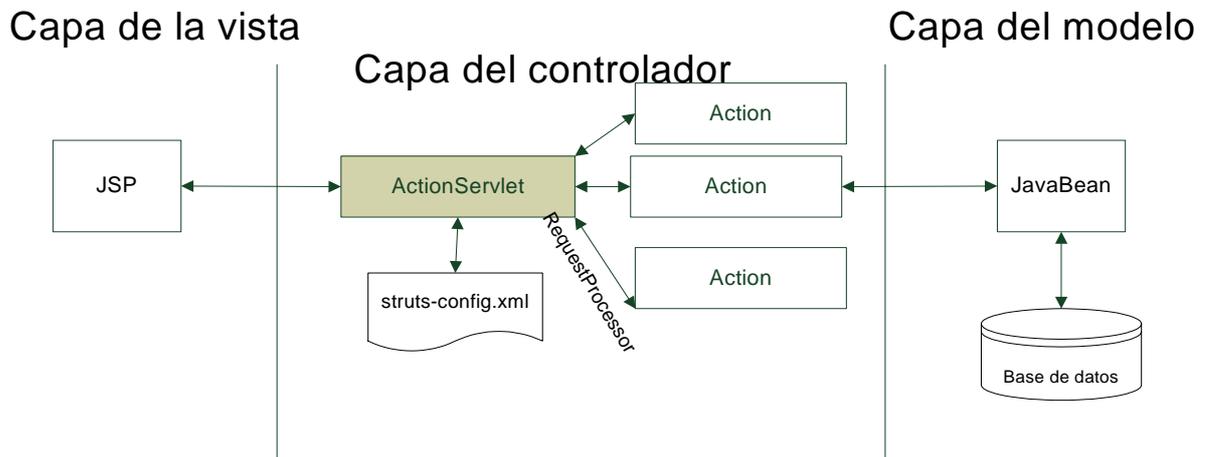


Figura 4.6. Implementación del patrón MVC con *struts*.

### 4.3 Implementación del sistema

Los *struts* son el marco de trabajo sobre el que se construye la *Web* con el lenguaje Java bajo el patrón MVC; esta implementación se realizó con un Entorno de Desarrollo Integrado (*Integrated Development Environment*, IDE por sus siglas en inglés) llamado *JBuilder 2005 Enterprise* ya que es un programa compuesto por un conjunto de herramientas como: el editor de código, el compilador, un depurador y un constructor de interfaz gráfica GUI. En el Anexo B se recopila la información de requerimientos de hardware y software.

A continuación se mostrarán las partes principales del sistema implementado a través de vistas del sistema, algunos diagramas de clases, *struts* y JSPs de la implementación.

#### 4.3.1. Subsistema Login

Este Subsistema se encarga de la autenticación de los datos del usuario en el sistema.

El primer proceso que se realiza en el sistema es de autenticar los datos del cliente, para esto se implementó una página JSP (*entrada.jsp*), la cual está en el lado de la vista, ver Figura 4.7; que se encarga de enviar los datos del cliente

al *ActionForm* (entradaActionForm.java en el controlador) y coloca en una clase los datos que el cliente proporciona (nombre y password). La Figura 4.8 muestra el diagrama de clases del ActionForm entradaActionForm.

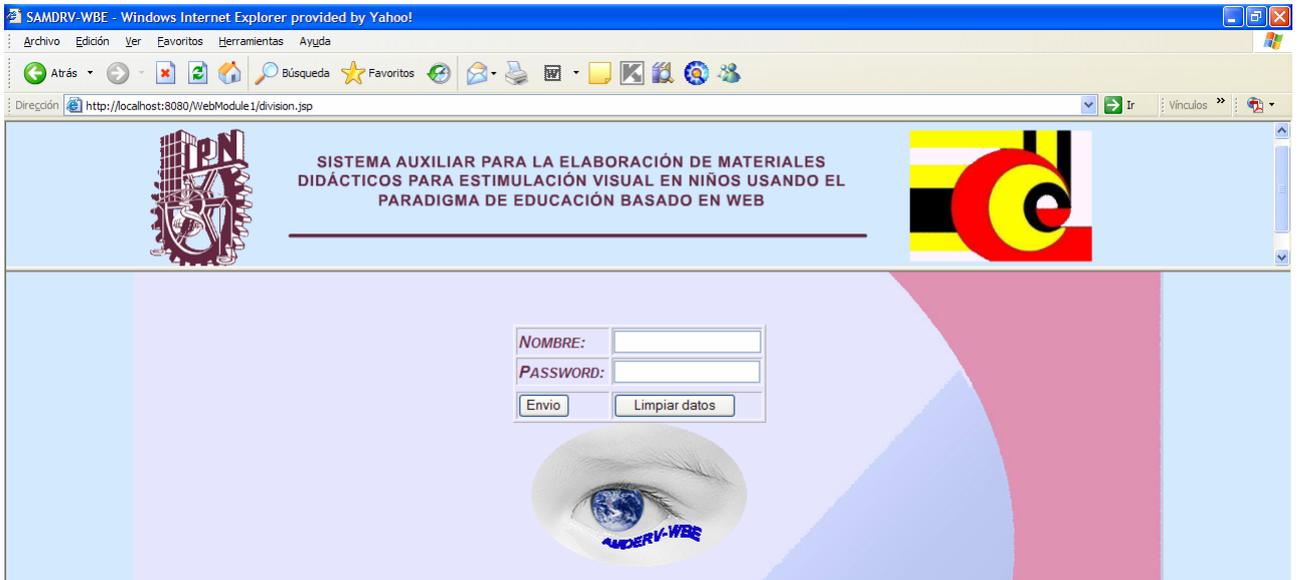


Figura 4.7. Vista de la página entrada.jsp

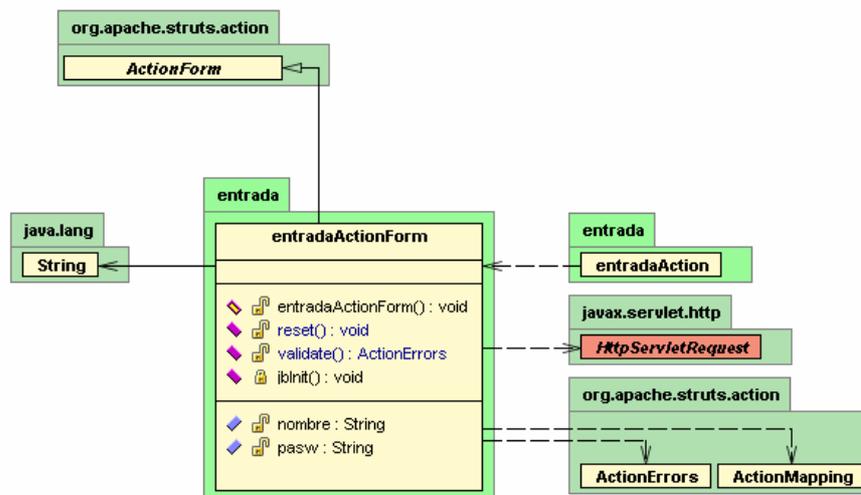


Figura 4.8. Diagrama de clase del ActionForm entradaActionForm.

Después se generó una subclase *Action* (entradaAction.java) que es parte del controlador, que recupera los datos de la petición del cliente manteniéndose esta información en la subclase *entradaActionForm*, y a continuación se determina en función del código, la acción a ejecutarse según el tipo de cliente, pero antes se valida que el identificador y password existan en la base de datos. Si hay un error, se envía un aviso como el de la Figura 4.9. Si es correcta la validación la base de datos regresa el tipo de usuario, existiendo tres tipos diferentes: administrador, profesor y alumno.

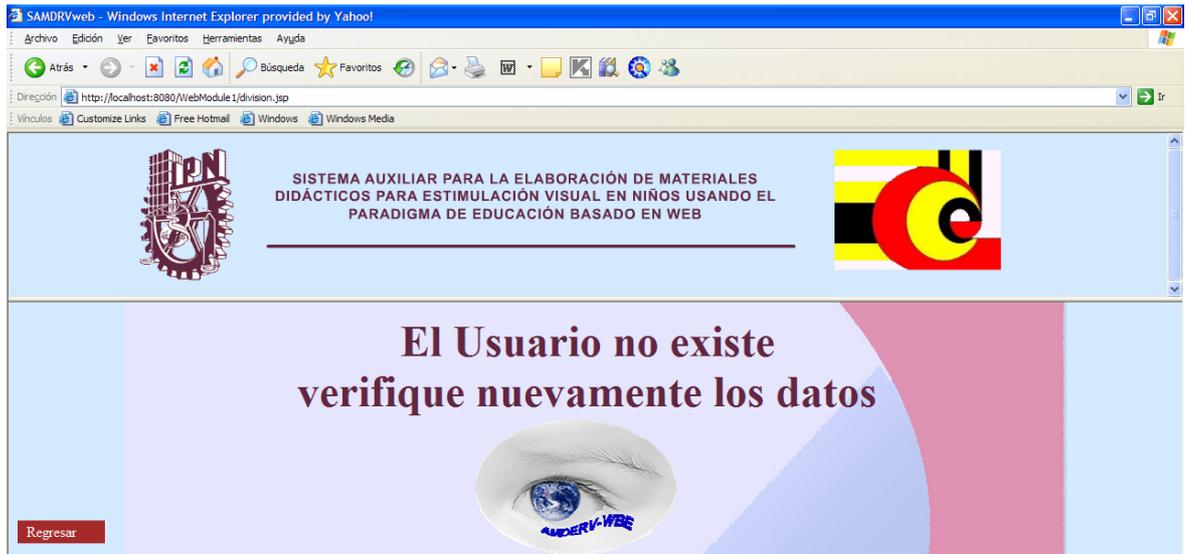


Figura 4.9 Página de salida cuando el cliente no existe en la base de datos.

El administrador puede realizar altas, bajas y cambios en los datos de los usuarios del sistema. El profesor puede crear y visualizar materiales didácticos, y por último el alumno solo puede visualizar el material didáctico ya creado con anterioridad por un profesor, para devolver el flujo a la subclase entradaAction.java y ésta envía el resultado a la vista, con cuatro posibilidades de login de los usuarios, que son las siguientes: Si no se encuentra en la base de datos (Figura 4.9), si es alumno (Figura 4.10), en caso de ser profesor (Figura 4.11), finalmente si el cliente es administrador (Figura 4.12).

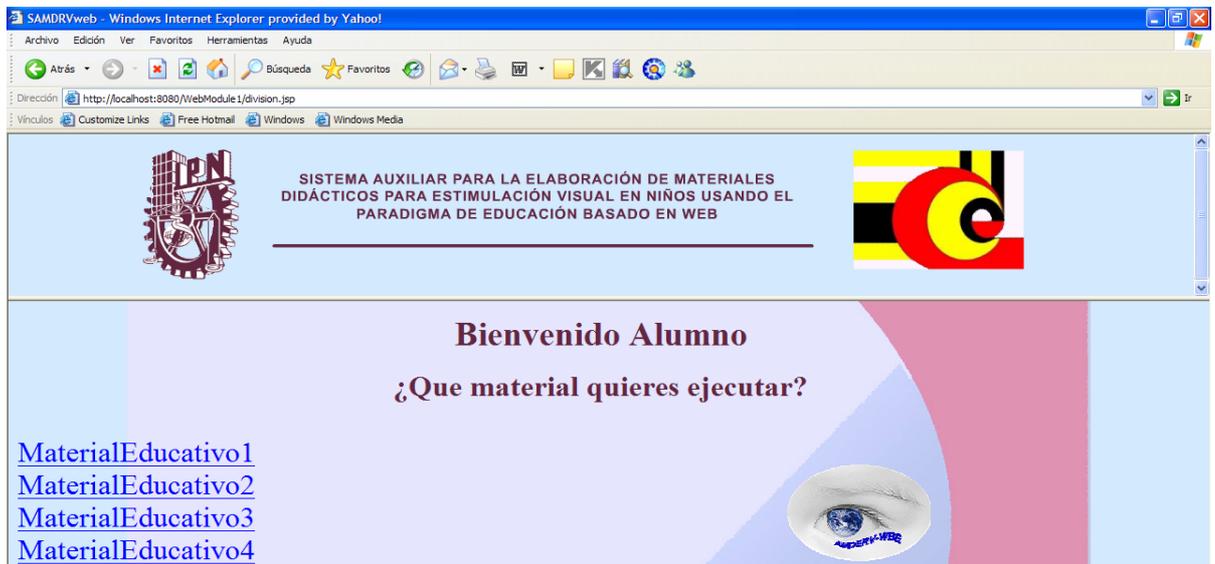


Figura 4.10. Página de salida cuando el cliente es Alumno.

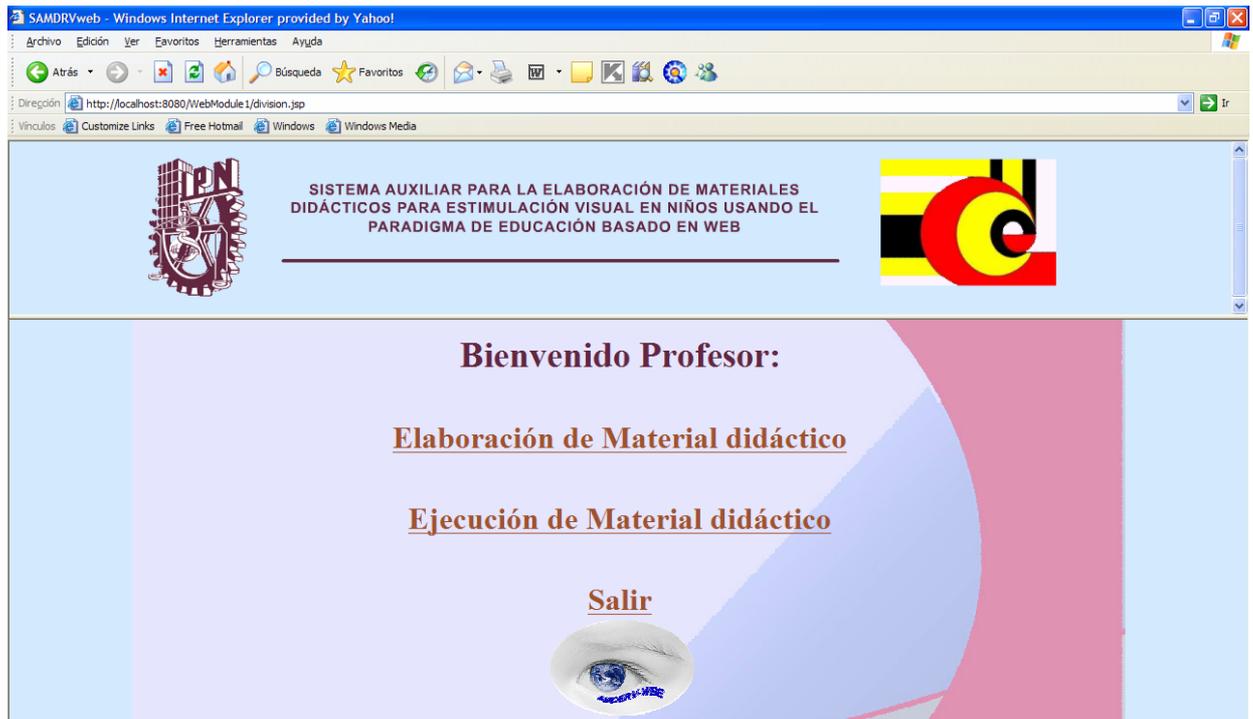


Figura 4.11. Página de salida cuando el cliente es un Profesor.

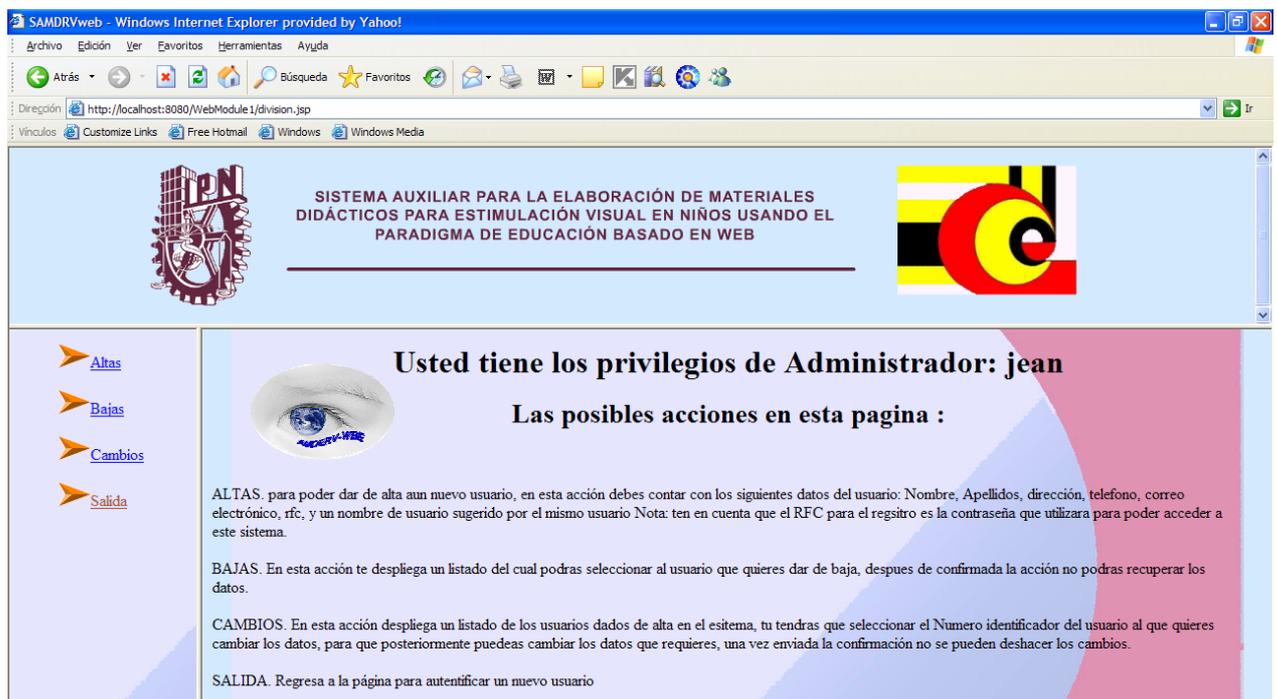


Figura 4.12. Página de salida cuando el cliente es un Administrador.

Realmente quien maneja el modelo es el Java *Bean* (BeanBaseDatos.java) y se encarga del modelo modificando, actualizando y agregando en la base de datos, mientras que el controlador implementado por *struts* mediante la clase

*ActionServlet* pone en la vista los resultados de la validación: *fracaso.jsp*, *alumno.jsp*, *profesor.jsp*, *administrador.jsp* según sea el caso.

Como resumen de lo descrito en los párrafos anteriores se muestra en la Figura 4.13 el gráfico de configuración de *struts*, denominado **struts-config.xml** del subsistema login.

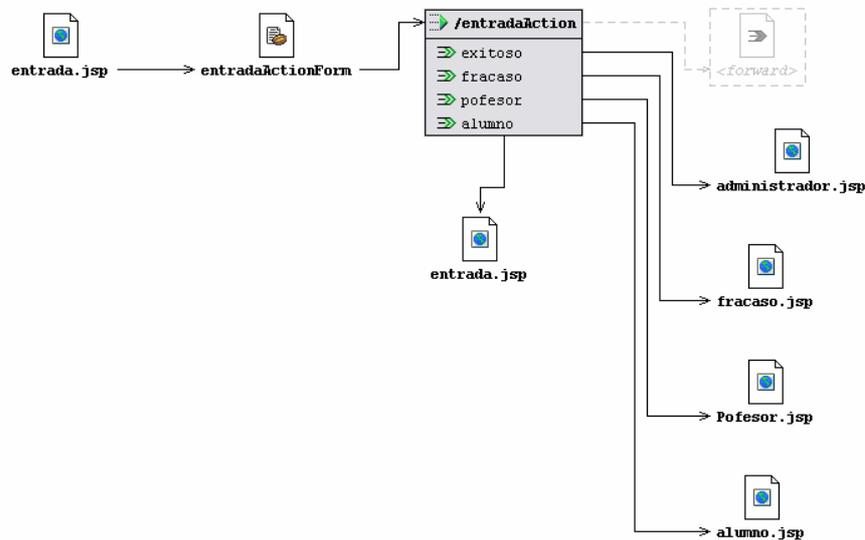


Figura 4.13. Diagrama del Subsistema de verificación del tipo de usuario con *struts*.

En los párrafos anteriores se menciono el acceso a una base de datos de la que a continuación se explicará con un poco más de detalle.

#### 4.3.1.1 Base de datos

El sistema cuenta con una base de datos para administrar los usuarios que pueden ingresar al sistema. Esta base de datos está administrada por un Sistema Administrador de Bases de Datos (*Data Base Management System*, DBMS por sus siglas en inglés). Un DBMS es el conjunto de programas que permiten la definición, manipulación y control de acceso para una o varias bases de datos. Con las características de facilitar la integridad, seguridad y confidencialidad de la información, accesar, almacenar con mínima redundancia, independizar el almacenamiento físico, compartir por diferentes usuarios y programas, insertar, actualizar, borrar y consultar los datos. El DBMS que se utilizó para implementar la base de datos es MySQL, en el cual se definió, construyó y se manipula la base de datos [66].

Definir una base de datos: consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenen. Construir una base de datos: es el proceso de almacenar los datos sobre algún medio de

almacenamiento. Manipular una base de datos: incluye funciones como consulta, actualización, etcétera de bases de datos [30].

Para la autenticación de usuarios se utiliza la tabla usuarios y los atributos que se muestran definidos en la Figura 4.14.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
nombre	VARCHAR(15)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
ap_pat	VARCHAR(20)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
ap_mat	VARCHAR(20)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
rfc	VARCHAR(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
direccion	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
codigo_post	VARCHAR(8)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
municipio	VARCHAR(20)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
telefono	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
usern	VARCHAR(20)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
passwd	VARCHAR(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
correo_e	VARCHAR(30)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
tusuario	VARCHAR(5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
id_usuario	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	

Figura 4.14. Atributos de la tabla usuarios.

### 4.3.2 Implementación de los componentes para el Sistema AMDERV-WBE

En la implementación existen ciertas condiciones para lograr que un componente sea funcional:

- 1) Los archivos de imágenes, sonidos, videos, textos y animaciones, las que se pueden obtener desde el Internet o editándolos en diversas herramientas entre ellas, las de edición de imágenes, edición de sonidos, edición de videos.
- 2) El programa que permita la **integración de medios** como imágenes, sonido, videos, texto, animación [30] (*Macromedia Flash 8*).
- 3) El **lenguaje de programación** que permita generar la interactividad y la reutilización de los medios dentro del componente (*ActionScript 2.0*). Como se muestra en el código en la Figura 4.15, en este caso el código se encarga de cargar dinámicamente las imágenes al componente de *Flash* por medio de un archivo XML generado dinámicamente.

```
function cargarPregunta(source)
{
    preg_xml = new XML();
    preg_xml.ignoreWhite = true;
    preg_xml.load(source);
    preg_xml.onLoad = function () {
        pregunta = this.firstChild.firstChild;
        preg_txt.text = pregunta.attributes.TEXTO;
        respuestaCorrecta = pregunta.attributes.RESPUESTA;
    }
}
```

```

        drags = pregunta.firstChild.childNodes;
        drag1.gotoAndPlay("on");
        drag1.pregunta.text = drags[0].firstChild.nodeValue;

drag1.contenedor.clip_mc.loadMovie(drags[0].attributes.IMAGEN);
    origen1._visible = true;
    if(drags.length >= 2)
    {
        drag2.gotoAndPlay("on");
        drag2.pregunta.text = drags[1].firstChild.nodeValue;

drag2.contenedor.clip_mc.loadMovie(drags[1].attributes.IMAGEN);
    origen2._visible = true;
    }
    if(drags.length >= 3)
    {
        drag3.gotoAndPlay("on");
        drag3.pregunta.text = drags[2].firstChild.nodeValue;

drag3.contenedor.clip_mc.loadMovie(drags[2].attributes.IMAGEN);
    origen3._visible = true;
    }
    if(drags.length == 4)
    {
        drag4.gotoAndPlay("on");
        drag4.pregunta.text = drags[3].firstChild.nodeValue;

drag4.contenedor.clip_mc.loadMovie(drags[3].attributes.IMAGEN);
    origen4._visible = true;
    }
    tmp = pregunta.childNodes;
    targets = tmp[1].childNodes;
    target1.gotoAndPlay("on");
    target1.respuesta.text = targets[0].firstChild.nodeValue;

target1.contenedor.multi.loadMovie(targets[0].attributes.IMAGEN);
target2.gotoAndPlay("on");
    target2.respuesta.text = targets[1].firstChild.nodeValue;

target2.contenedor.multi.loadMovie(targets[1].attributes.IMAGEN);
    if(targets.length >= 3)
    {
        target3.gotoAndPlay("on");
        target3.respuesta.text = targets[2].firstChild.nodeValue;

target3.contenedor.multi.loadMovie(targets[2].attributes.IMAGEN);
    }
    if(targets.length == 4)
    {
        target4.gotoAndPlay("on");
        target4.respuesta.text = targets[3].firstChild.nodeValue;

target4.contenedor.multi.loadMovie(targets[3].attributes.IMAGEN);
    }
    }
}

```

Figura 4.15. Código en *ActionScript* que carga la información de un XML al componente.

Esto permite obtener como resultado un componente de relación de columnas que se muestra en la Figura 4.16.

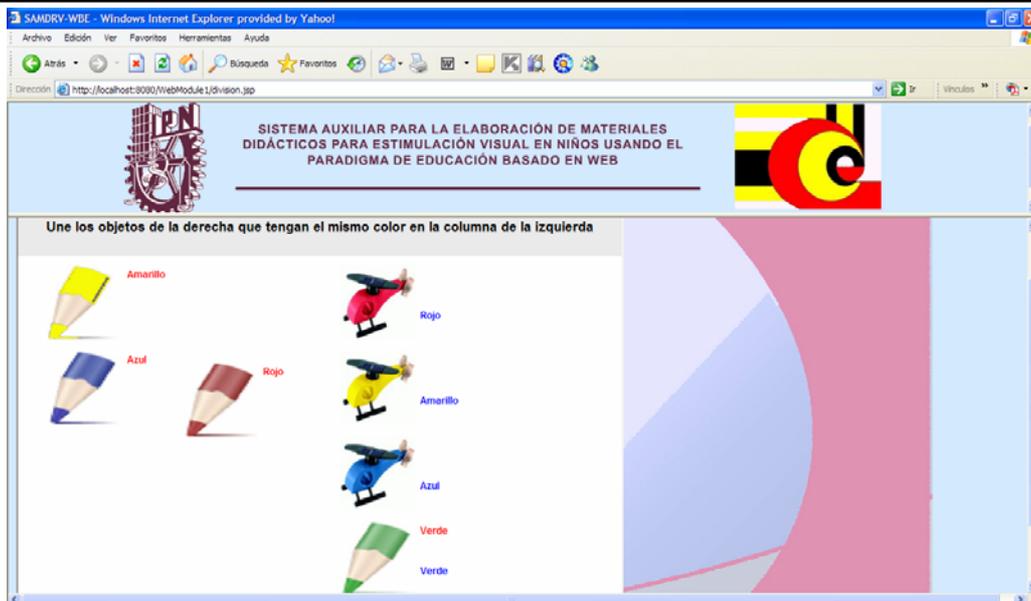


Figura 4.16. Componente de relación de columnas.

Los componentes elaborados para este sistema son tres: relación de columnas, memorama y relación de sonido con imagen.

El propósito del componente de “relación de columnas” es mostrar dos columnas de imágenes que se deben relacionar. Las imágenes de la columna izquierda se desplazan hacia la derecha para hacer coincidir con las imágenes de esta columna, de manera tal que ahí exista una correspondencia, como se aprecia en la Figura 4.16. El componente se llama prototipo\_RC.

El componente de “memorama” tiene como finalidad presentar un conjunto de imágenes cubiertas, para que el usuario encuentre los pares de imágenes iguales. Estas imágenes están en formato jpg, los nombres de las imágenes se pasan como parámetros. Como se puede ver en la Figura 4.17, se hace uso de la flexibilidad de la reutilización, porque se pueden optar por imágenes diferentes, las cuales se cargan en tiempo de ejecución mediante el componente memory.

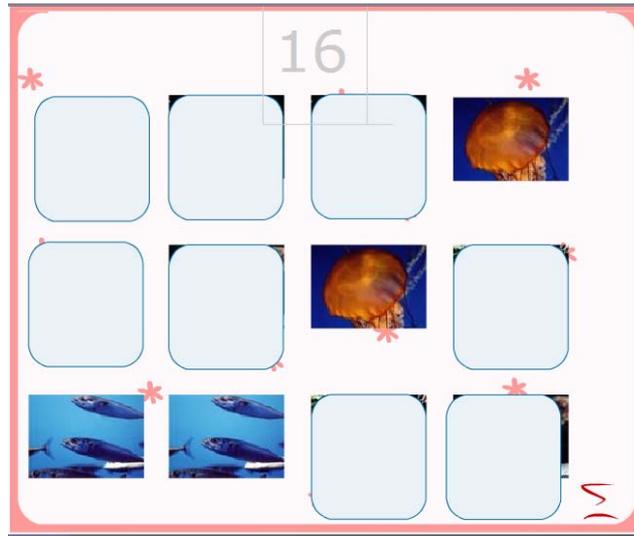


Figura 4.17. Componente de memorama.

El componente de “relación de sonido con imagen” tiene como objetivo presentar una serie de imágenes entre las cuales hay que seleccionar la indicada por la pregunta, que está relacionada con el sonido; en este componente se pasa por parámetro los nombres de las imágenes con formato jpg, el nombre del archivo del sonido en formato mp3 y la pregunta como texto como se muestra en la Figura 4.18. Todo se carga en tiempo de ejecución mediante el componente prototipo\_SAs.



Figura 4.18. Componente de relación de imagen con sonido.

La creación de las propias etiquetas, al estilo del Lenguaje de Marcado de Hiper Texto (*HiperText Markup Lenguaje*, HTML por sus siglas en inglés) utilizando el Lenguaje de marcado extensible (*eXtesible Markup Lenguaje*, XML por sus siglas en inglés), realiza un meta-etiquetado el cual se utiliza para poner la información que contendrá el componente como es: título, ruta de las imágenes, nombre de la imagen, la pregunta a elaborar, la solución correcta a la pregunta, etc. Todo ello para lograr reutilización, haciendo que los

componentes estén separados de los contenidos para permitir una configuración dinámica.

Para implementar estos componentes utilizamos el API DOM discutido en el subtema 3.4 del capítulo 3, que permite crear estructuras de tipo árbol; además de contar con una serie de clases y métodos que hace que se facilite el trabajo con los documentos XML dinámicos; no debemos olvidar que el objeto viene representado por *Document*. *Document* es una interfaz abstracta de fábrica utilizando el patrón de diseño de método de fábrica estipulando que al generar el nuevo objeto hace que las aplicaciones no dependan de las clases implementadas.

DOM tiene disponible un archivo comprimido con un conjunto de clases e interfaces Java (paquete org.w3c.dom) dentro de la API JAXP (Java API for XML parsing). Esto se importa con el paquete javax.xml.parsers.\*; éste, a su vez, utiliza las clases *DocumentBuilderFactory* (utilizada para la validación del documento), también cuenta con manejo de excepciones en la forma de una clase para crear objetos de excepciones; la clase se denomina *ParserConfigurationException*.

Por lo tanto los pasos para crear un XML utilizando el JAXP es instanciar un objeto *DocumentBuilderFactory*, después instanciar un objeto *DocumentBuilder* el cual utilizará un "parser" con el comportamiento deseado creando así el documento mediante *DocumentBuilder*; el código de nuestro sistema permite crear cuatro diferentes archivos XML, para ejemplificar, solo utilizaremos el código fuente que se presenta en la Figura 4.19 que genera un archivo XML llamado *slides.xml* del componente memorama.

```
//Creando el archivo slides.xml el cual es utilizado por el objeto
FLASH
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    DocumentBuilder db = dbf.newDocumentBuilder();
    Document documento = db.newDocument();
    //Creando el nodo raíz
    Element elementoSlides = documento.createElement("Slides");
    //Creando el cuerpo de nodos con información sobre las imágenes
    for(int i = 0; i < 6; i++){
        Element slideNode = documento.createElement("slideNode");
        slideNode.setAttribute("jpegURL", "images/" +
nombresArchivos[i]);
slideNode.appendChild(documento.createTextNode(fgm.getTitulo(i)));
        elementoSlides.appendChild(slideNode);
    }
    //Agregando el nodo raíz al documento
    documento.appendChild(elementoSlides);
    //Salvando el documento xml en la carpeta del material y en la carpeta del
reactivo
    XmlFile documentoXml = new XmlFile();
    documentoXml.salvar(documento, new File(rutaDirectorio +
"\slides.xml"));
```

Figura 4.19. Código fuente que utiliza JAXP para generar el archivo slides.xml del componente del memorama.

El código anterior genera el árbol jerárquico que se muestra en la Figura 4.20 mediante DOM.

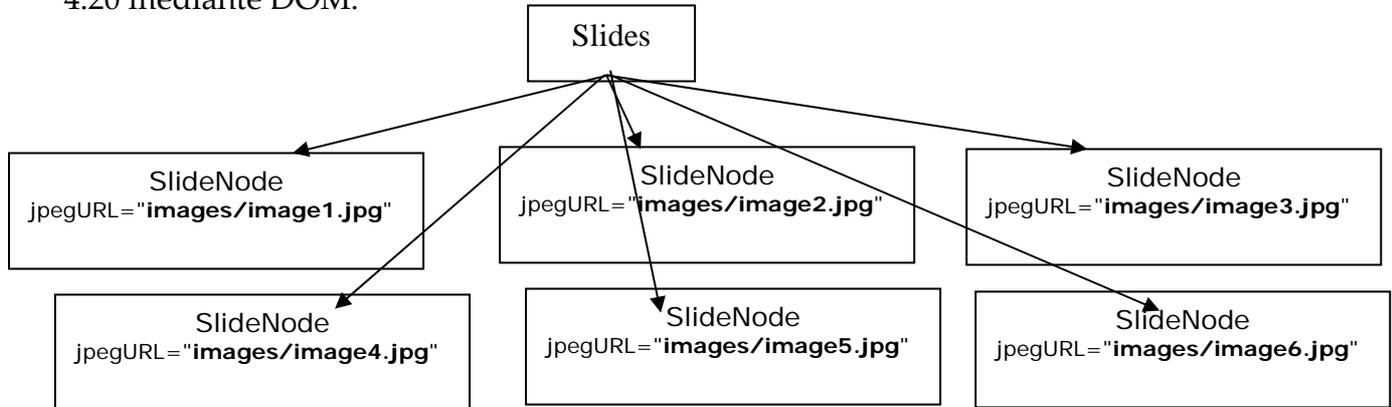


Figura 4.20. Árbol de objetos DOM para el archivo slides.XML para el componente memorama.

El proceso de transformación del árbol DOM a un archivo XML se muestra en la Figura 4.21, construyéndose mediante XSLT, por medio del *framework* XALAN.

```

<?xml version="1.0" encoding="UTF-8" ?>
<Slides>
  <slideNode jpegURL="images/image1.jpg" />
  <slideNode jpegURL="images/image2.jpg" />
  <slideNode jpegURL="images/image3.jpg" />
  <slideNode jpegURL="images/image4.jpg" />
  <slideNode jpegURL="images/image5.jpg" />
  <slideNode jpegURL="images/image6.jpg" />
</Slides>
  
```

Figura 4.21. Archivo slides.XML resultado del árbol DOM.

La generación de los distintos archivos XML para los distintos componentes del sistema se realizó utilizando *struts*. A continuación se mostrarán los distintos diagramas de *struts* que generan a los archivos XML que se utilizan con los componentes antes descritos. El strut *accionGenerarMaterial*, es el inicio para generar los archivos XML, a través de un menú para los diferentes materiales didácticos (componentes), su diagrama se puede ver en la Figura 4.22.

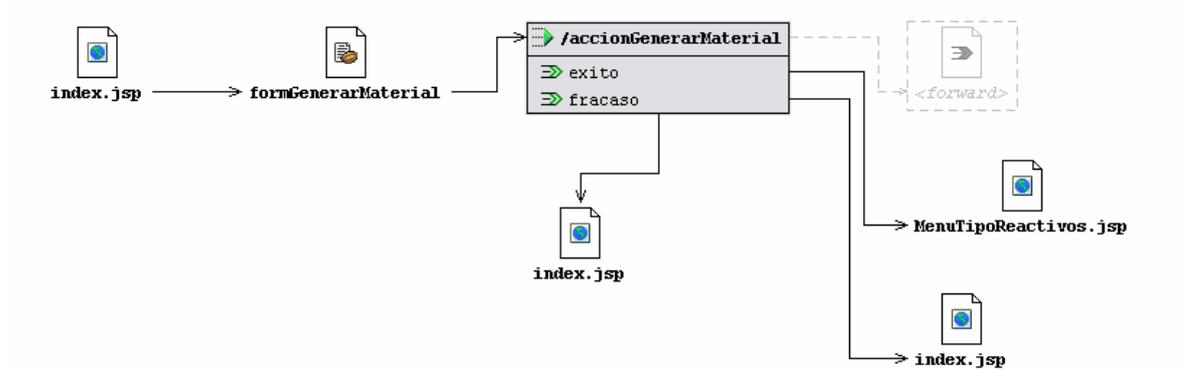


Figura 4.22. Diagrama de `strut-config.xml` del Sistema que genera los archivos XML de los distintos tipos de componentes.

El `strut` `accionGenerarMemorama` genera el archivo XML para el componente memorama, su diagrama se muestra en la Figura 4.23.

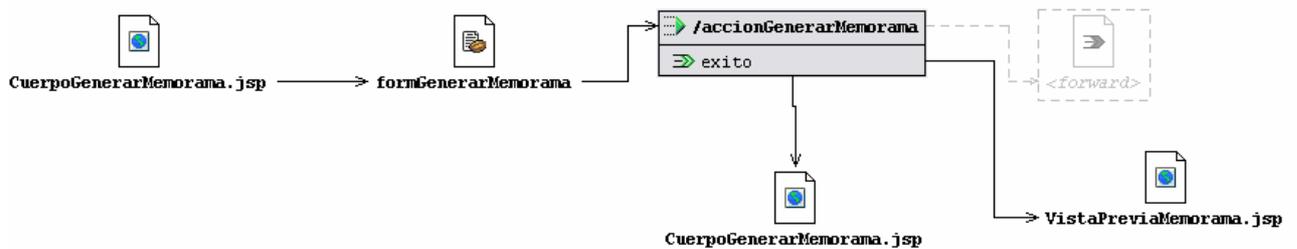


Figura 4.23. Diagrama de `strut` que genera el Archivo XML para el componente memorama.

El `strut` `accionRelacionColumnas` genera el archivo XML para el componente relación de columnas, su diagrama se muestra en la Figura 4.24.

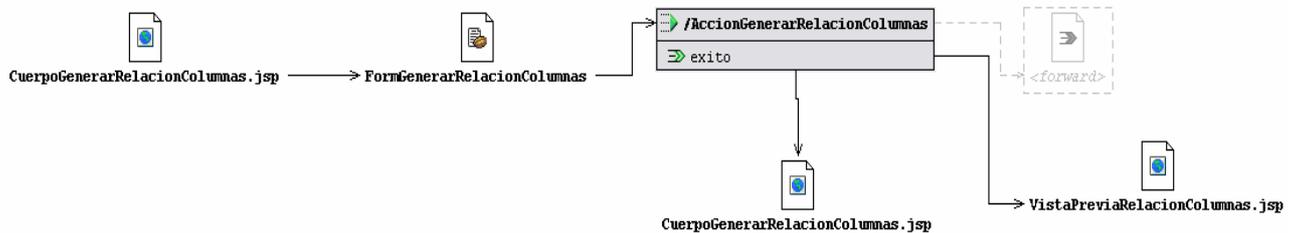


Figura 4.24. Diagrama de `Strut` que genera el Archivo XML para el componente relación columna.

El strut `accionGenerarSonidoImagen` genera el archivo XML para el componente relación de sonido con imagen, su diagrama se muestra en la Figura 4.25.

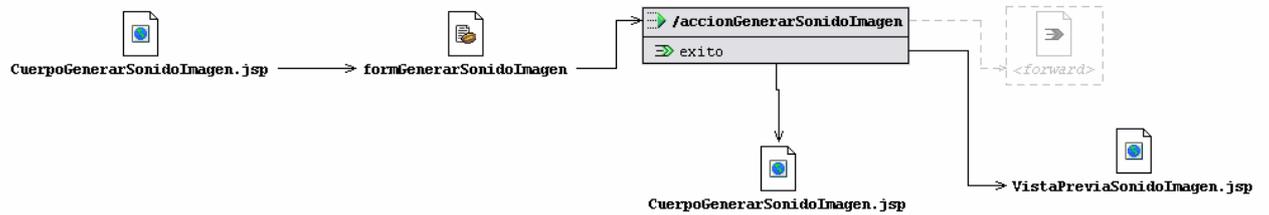


Figura 4.25. Diagrama de *strut* que genera el Archivo XML para el componente relación de sonido con imagen.

El *strut* `accionTerminarExamen` genera el archivo examen.XML que integra el número de componentes creados para visualizar todos los componentes de memoramas, relación de columnas, y relación de sonido con imagen consecutivamente según el orden elegido. Éste se muestra en un componente contenedor llamado cargapregunta y aquí se acepta o se descarta el término del material didáctico. Su diagrama se muestra en la Figura 4.26.

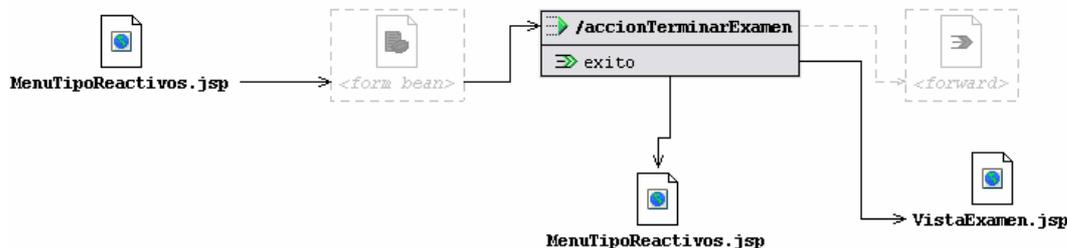


Figura 4.26. Diagrama de *strut* que genera el Archivo XML que integra los componentes de memoramas o relación de columnas o relación de sonido con imagen.

Hasta el momento ya contamos con el sistema que crea los archivos XML que logran hacer persistente la información que mostraran los componentes. A continuación se describirá el subsistema que permite almacenar y organizar los archivos en el servidor y así poderlos utilizar.

### 4.3.3 Implementación del subsistema de subida de archivos

Para implementar este subsistema se creó una clase, que obtiene los datos para los archivos multimedia creados, que se llama `StrutsUploadForm` descrita anteriormente en la Figura 3.16 y cuenta con la información de los nombres de

los archivos multimedia, ya sea imágenes o sonidos junto con los directorios en los que se encuentra ubicada. En la Figura 4.27 se muestra un segmento de código de esta implementación, en el momento que realiza la copia binaria de los archivos, subiéndolos al servidor y conformando la estructura de directorios de los recursos para su posterior utilización en los componentes.

```
//copia binaria del archivo
FormFile myFile = myForm.getTheFile();
String contentType = myFile.getContentType();
//obtiene el nombre del archivo
String fileName = myFile.getFileName();
//obtiene tamaño del archivo
int fileSize = myFile.getFileSize();
byte[] fileData = myFile.getFileData();
buffer = new byte[1024];
//ruta del archivo de origen
plantillaFlash = new File(request.getSession().getServletContext().getRealPath("c:\\\" + fileNam));
entrada = new FileInputStream(plantillaFlash);
salida = new FileOutputStream((String)request.getSession().getAttribute("c:\\\" + fileNam);
//empieza a copiar los datos del origen al destino bit a bit
while((bytesLeidos = entrada.read(buffer, 0, 1024)) != -1)
salida.write(buffer, 0, bytesLeidos);
//ruta de destino del archivo
FileOutputStream grabador = new FileOutputStream("c:\\\" + fileName);
//copia los datos
grabador.write(fileData);
grabador.
```

Figura 4.27. Segmento del código del StrutsUploadForm.

## Resumen

En este capítulo se describió el funcionamiento del sistema definiendo a la arquitectura Cliente / Servidor y las tareas que realizan las JSPs, *servlets*, Java *Beans*, *struts* para posteriormente describir paso a paso cada uno de los subsistemas que integran al proyecto: el subsistema de *login* que autentifica al usuario para enviarle el ambiente de trabajo correspondiente a su tipo ya sea administrador, profesor o alumno; el subsistema de creación de archivos XML para la reutilización de componentes que fueron elaborados previamente a través del software Flash; y el subsistema que administra y maneja la base de datos.

Por lo tanto, se utilizó la estructura de un *framework* basado en patrón MVC para conformar el sistema con: JSPs encargadas de mostrar la visualización de la información (dentro de este documento se utilizó algunas de las pantallas); Java *Beans* que llevan la lógica de la aplicación y el manejo de la base de datos; y por último los *struts* que se encargan del control y administración de las JSPs y los Java *Beans*.

En el próximo capítulo se realizarán pruebas del funcionamiento y se presentarán los resultados obtenidos.

---

---

## CAPÍTULO 5

### PRUEBAS Y RESULTADOS

---

---

En este capítulo básicamente se describen las pruebas realizadas al sistema y se dan los resultados obtenidos mediante la aplicación de una prueba de campo a algunos profesores que hicieron uso de las actividades y/o servicios del sistema AMDERV-WBE.

## 5.1 Pruebas

La prueba de los sistemas es usualmente detallada y rigurosa, ya que se requiere asegurar que cada componente del sistema esté en operación y que el sistema en su conjunto se desempeñe exactamente de acuerdo con lo propuesto.

Para realizar estas pruebas se considero un grupo de 10 profesores, dedicados a la docencia, que imparten clases a niños del primer año a nivel primaria, debido a que el objetivo es crear un apoyo para que los profesores armen sus materiales didácticos.

Se probó el sistema con equipos, bajo condiciones de operación real, un salón de clase con 25 computadoras conectadas en red local, en conjunto con la máquina que fungió de servidor.

Las características de hardware de las computadoras que fungieron como clientes son: procesador PIV a 2.13GHz., disco duro de 80 Gb., memoria de 128 Mb. en RAM., y tarjeta de Ethernet 10/100.

El hardware del servidor utilizado cuenta con las siguientes características: procesador Centrino Duo a 1.66 GHz., disco duro de 100 Gb., memoria de 1Gb. en RAM., y tarjeta de Ethernet 10/100.

Las características de Software para las computadoras clientes fue: sistema operativo Windows Xp y navegador Mozilla Firefox.

Las características de software para la computadora del servidor son: sistema operativo Windows XP, un servidor *Web*, contenedor de Servlets y JSPs llamado Apache Tomcat, mas la maquina virtual de Java quienes se encargan de implementar el protocolo HTTP, el cual está diseñado para transferir hipertexto, paginas *Web*, formularios, objetos incrustados y un DBMS que ejecuta las operaciones necesarias sobre la base de datos a través de MySQL.

Considerando las anteriores especificaciones técnicas, se colocó el sistema implementado en el servidor, para que, a través de un navegador, los profesores pudieran utilizar el sistema AMDERV-WBE.

Sin olvidar que el sistema puede ser manipulado por tres diferentes usuarios: Administrador, Profesor y Alumno, quienes darán sus datos en el subsistema Login como se ve en la Figura 4.7, después de verificar los datos y el tipo de usuario, los profesores que participaron en la prueba entraron al subsistema dedicado a los profesores.

El subsistema para los profesores les dio la posibilidad de armar nuevos materiales didácticos y ejecutar los ya armados, mediante un menú como se ve en la Figura 4.11.

En el Anexo B Manual del Usuario, se describe con mayor detalle como se usa el sistema AMDERV-WBE.

Los resultados de las pruebas del funcionamiento del sistema para profesores se presentan en la Tabla 5.1.

**TABLA 5.1** Pruebas de funcionamiento del sistema AMDERV-WBE

<b>Actividad</b>	<b>Datos de entrada</b>	<b>Resultado esperado</b>	<b>Observaciones</b>
Selección de usuario Profesor	Elección de la acción	Acceso a las acciones de profesor y selección de acción.	Ninguna
Ejecución de material didáctico (MD)	Elección del MD	Ejecución del MD seleccionado	Ninguna
Ejecución de MD	Realizar indicaciones señaladas	Obtener calificación	Ninguna
Elaboración de material didáctico	Elección del tipo de MD a elaborar	Acceso al área de elaboración del MD seleccionado	Ninguna
Elaboración de MD memorama	Llenado de los datos necesario (imágenes)	Integración de datos y creación de Archivo XML y ejecución del componente	Ninguna
Confirmar MD memorama	Elección del icono	Guarda Componente y Archivo XML en los correspondientes directorios, o borra toda información	Ninguna
Elaboración de MD Relación de Columnas	Llenado de los datos necesarios (imágenes y textos)	Integración de datos y creación de Archivo XML y ejecución del componente	Ninguna
Confirmar MD Relación de Columnas	Elección del icono	Guarda Componente y Archivo XML en los correspondientes directorios, o borra toda información	Ninguna
Elaboración de MD Relación de Sonido con Imagen	Llenado de los datos necesarios (imágenes, sonidos y textos)	Llenado de los datos necesarios (imágenes y textos)	Ninguna

Salida del servicio Profesor	Se oprime opción salida	Se regresa a página de inicio	Ninguna
Selección de usuario Alumno	Elección de la acción	Acceso al menú de MD elaborados previamente	Ninguna
Selección de MD varios	Elección del MD	Ejecución del MD seleccionado	Ninguna
Ejecución de MD	Realizar indicaciones señaladas	Obtener calificación	Ninguna

Para obtener más información acerca de estas pruebas se elaboró un cuestionario que se aplicó después de que los profesores utilizaron el sistema AMDER-WBE. El cuestionario consiste de catorce preguntas (ver el Anexo C de encuestas), que fueron contestadas por los profesores para verificar si se obtuvo el resultado de la propuesta de la Tabla 2.1.

### Cuestionario aplicado a diez profesores

Las preguntas del cuestionario son:

1.- ¿El software que observó, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican cómo manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias haría para mejorar el software?

---

---

---

## 5.2 Resultados

En el anexo C se puede confirmar que 9 profesores contestaron el cuestionario de forma afirmativa y del análisis realizado a las respuestas del cuestionario se puede derivar que el software es útil para los profesores ya que les facilita el armado de material didáctico, orientado a la estimulación visual, apoyando, la coordinación visomotriz, constancia espacial, figura a fondo, posición en el espacio y relación espacial utilizando sonidos en el idioma y acento del profesor, además adaptándose de forma flexible e interactiva a sus necesidades.

Los profesores expresaron las siguientes sugerencias para mejorar el software:

Implementar más tipos de juegos, una galería de imágenes y sonidos, incluso tener su propia herramienta para dibujo y edición de sonido.

Considerar la forma de utilización que los alumnos hacen del software para elaborar mejoras al mismo.

Es importante que el software no este dirigido sólo a reforzar esos conceptos, si no que se analice en que otros temas se puede emplear.

## **Resumen**

En este capítulo se realizaron las pruebas al sistema AMDERV-WBE utilizando el subsistema para profesores, mediante una prueba de campo con diez profesores que manipularon la herramienta propuesta; los profesores elaboraron material didáctico e incluso lo emplearon con algunos de sus alumnos. Por lo que, se puede concluir que se obtuvieron satisfactoriamente los resultados que se buscaban originalmente en los objetivos.

En el siguiente capítulo se presentan algunas de las conclusiones y se sugiere la elaboración de trabajos futuros que se desprenden de esta tesis.

---

---

## CAPÍTULO 6

### CONCLUSIONES Y TRABAJOS A FUTURO

---

---

En éste capítulo se presentan las conclusiones del desarrollo de este trabajo, así como algunas mejoras a futuro que se pueden realizar con la finalidad de enriquecer éste.

## 6.1 Conclusiones

El desarrollo de materiales educativos didácticos, necesita de las nuevas tecnologías, los cuales son instrumentos de trabajo que ayudan a la labor diaria de los profesores en el proceso de enseñanza.

Por lo que en el desarrollo del presente trabajo, se han aplicado distintas metodologías para el análisis del sistema como: programación orientada a objetos y el Lenguaje de Modelado Unificado (UML) para especificaciones. Se usó la arquitectura Cliente / Servidor, nuestra aplicación se creó utilizando el patrón de diseño MVC, y para su implementación un *framework* basado en Java denominado *struts*. Este *framework* implementa el patrón MVC a través de: *Beans*, JSPs y *servolets*. Mientras que la Implementación de los componentes para desplegar elementos multimedia fueron realizados en *ActionScript*, posibilitando el uso de imágenes, sonido, animación e interacción, utilizando archivos XML bajo el modelo DOM, y creando los archivos XML mediante la implementación de XSLT en Java, lo que permite la reutilización, ya que cada vez que se hace un nuevo material didáctico se reutiliza el componente seleccionado, el cual puede ser un memorama, relación de columnas o relación de sonido con imagen, cambiando únicamente las especificaciones de imágenes, tipo de pregunta, respuesta, textos a mostrar y sonido elegido por el usuario; así se personaliza el componente correspondiente dentro de la estructura del sistema.

Por lo que el sistema está constituido por tres subsistemas:

- Subsistema basado en *Web* para el profesor que permite elaborar materiales didácticos para estimulación visual, el resultado se puede verificar en el tema pruebas del capítulo 5 en la página 87.
- Subsistema basado en *Web* para el alumno que permite visualizar los materiales didácticos elaborados por el profesor. En él se encuentran los elementos multimedia a través de imágenes, sonido, animaciones, videos e interacciones que ayudan a estimular el reconocimiento visual en niños durante la etapa de aprendizaje de la lectura a través de la computadora.
- Subsistema basado en *Web* para el administrador que se encarga de las altas, bajas y modificaciones de los usuarios, el resultado se puede cotejar en el Anexo A manual de usuario.

Se elaboró una herramienta apoyada en el paradigma de educación basado en *Web*, utilizando la arquitectura Cliente / Servidor y haciendo uso de componentes de software los cuales permiten simplificar el desarrollo de

materiales didácticos que estimulen el reconocimiento visual en niños que estén en la etapa del aprendizaje de la lectura; obteniendo los beneficios del paradigma como: la personalización del material, que él alumno pueda avanzar a su propio ritmo, y finalmente el bajo costo por la utilización de software libre; ya que si hacemos una comparativa del software comercial (Windows Server, Internet Information Server, MS SQL Server, Internet Explorer), que se utilizaría, contra el software libre (Linux, Apache Tomcat, MySQL, Mozilla), el costo es mucho menor .

## 6.2 Trabajos a futuro

Como cualquier sistema desarrollado se encuentra inmerso en un ciclo de vida. Creo que el futuro de un sistema como el que se desarrolló es amplio si se conforma un equipo multidisciplinario constituido por: pedagogos, profesores, diseñadores, programadores, etc.

A continuación se presentan algunas mejoras que podrían realizarse a este sistema:

- Desarrollo de más componentes que agreguen grados de complejidad a los ya existentes.
- Desarrollo de más componentes utilizados en el campo del reconocimiento visual.
- Crear un manejador de interfaces para que este sistema pueda ser modificado a las necesidades de cada institución (cambiando los títulos, los colores a la portada, los letreros fijos en las diferentes JSPs, etcétera).

Además se ha pensado un conjunto de extensiones que se podrían realizar:

- Agregar una membrana "Touch screen" al monitor para evitar el uso del ratón y así las indicaciones se den a través del monitor.
- Crear un ambiente donde se tengan una mayor gama de formas de comunicación haciendo uso de las tecnologías computacionales como: la comunicación simultánea entre dos o más personas a través de Internet (Chat), videoconferencia, correo electrónico, archivos de audio que pueden bajarse de la *Web*, para posteriormente escucharlos en la computadora (*PodCast*), foros de discusión, etcétera.

## REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- [1] Azcoaga.J.E., Y Cols. Los retardos del lenguaje en el niño. Ed. Paidós, España, 1987.
- [2] Describe el software de Actividades de WIN -ABC  
<http://perso.wanadoo.es/postigoaula/winabc/winabc.htm>, agosto 2007.
- [3] Describe el software de El mono coco  
<http://perso.wanadoo.es/postigoaula/monococo/monococo.htm>, agosto 2007.
- [4] Describe el software de Jugar con  
[http://perso.wanadoo.es/postigoaula/jugar\\_con/jugar\\_con.htm](http://perso.wanadoo.es/postigoaula/jugar_con/jugar_con.htm), agosto 2007.
- [5] Describe el software de Actividades de lectoescritura  
[http://clic.xtec.net/db/act\\_es.jsp?id=3298](http://clic.xtec.net/db/act_es.jsp?id=3298), agosto 2007.
- [6] Describe el software de Los colores [http://clic.xtec.net/db/act\\_es.jsp?id=3327](http://clic.xtec.net/db/act_es.jsp?id=3327), agosto 2007.
- [7] Describe el software de En serie [http://clic.xtec.net/db/act\\_es.jsp?id=3361](http://clic.xtec.net/db/act_es.jsp?id=3361), agosto 2007.
- [8] Describe el software de Los tamaños  
[http://clic.xtec.net/db/act\\_es.jsp?id=3328](http://clic.xtec.net/db/act_es.jsp?id=3328) agosto 2007.
- [9] Describe el software de Las letras que suenan  
[http://clic.xtec.net/db/act\\_es.jsp?id=3297](http://clic.xtec.net/db/act_es.jsp?id=3297), agosto 2007.
- [10] Describe el software de Aprende los animales  
[http://clic.xtec.net/db/act\\_es.jsp?id=3278](http://clic.xtec.net/db/act_es.jsp?id=3278), agosto 2007.
- [11] Describe el software de Figuras <http://clic.xtec.net/db/jclicApplet.jsp?project=http://clic.xtec.net/jclic/figuras.jclic.zip&lang=es&title=Figuras>, agosto 2007.
- [12] Describe el software de Los primeros números por vía visual directa  
[http://clic.xtec.net/db/act\\_es.jsp?id=3155](http://clic.xtec.net/db/act_es.jsp?id=3155), agosto 2007.
- [13] Describe el software de Paco “el Chato”  
[http://clic.xtec.net/db/act\\_es.jsp?id=3156](http://clic.xtec.net/db/act_es.jsp?id=3156), agosto 2007.

- [14] Describe el software de Desberdintasunak [http://clic.xtec.net/db/act\\_es.jsp?id=2720](http://clic.xtec.net/db/act_es.jsp?id=2720), agosto 2007.
- [15] Describe el software de Adentro afuera [http://clic.xtec.net/db/act\\_es.jsp?id=2565](http://clic.xtec.net/db/act_es.jsp?id=2565), agosto 2007.
- [16] Describe el software de Jugando activo mi inteligencia [http://clic.xtec.net/db/act\\_es.jsp?id=2564](http://clic.xtec.net/db/act_es.jsp?id=2564), agosto 2007.
- [17] Describe el software de ¿Cuál es cual? [http://clic.xtec.net/db/act\\_es.jsp?id=2280](http://clic.xtec.net/db/act_es.jsp?id=2280), agosto 2007.
- [18] Describe el software de ANICLIC Método de lectura [http://clic.xtec.net/db/act\\_es.jsp?id=2241](http://clic.xtec.net/db/act_es.jsp?id=2241), agosto 2007.
- [19] Describe el software de A gaviota de A Coviña (la gaviota de A Coviña) [http://clic.xtec.net/db/act\\_es.jsp?id=2184](http://clic.xtec.net/db/act_es.jsp?id=2184), agosto 2007.
- [20] Describe el software de Jugando aprendo [http://clic.xtec.net/db/act\\_es.jsp?id=2008](http://clic.xtec.net/db/act_es.jsp?id=2008), abril 2008.
- [21] Describe el software de Jclic autor <http://clic.xtec.net>, agosto 2007.
- [22] Documentación de UML <http://www.clikear.com/manuales/uml>, agosto 2007.
- [23] Objetos de grupos dirigidos <http://www.omg.org>, septiembre 2007.
- [24] Definición de Frameworks <http://www.acm.org/crosroads/espanol/Xrds7-4/#fayad>, octubre 2007.
- [25] Significado de la siglas Object Management Group's <http://www.acm.org/crosroads/espanol/Xrds7-4/#corba>, octubre 2007.
- [26] Significado de la siglas COM+ y DCOM de Microsoft <http://www.acm.org/crosroads/espanol/Xrds7-4/#com>, octubre 2007.
- [27] Ejemplo de framwork <http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html>, octubre2007.
- [28] Jeff Tapper, James Talbot, Robin Haffner ,Object-Oriented Programming with ActionScript 2.0, Ed. New Riders Publishing, 2004.

- [29] M. Uramoto, Creación de sitios Web con XML y Java, Prentice Hall, 2000.
- [30] Michael J. Hernandez, Database Design for Mere Mortals™: A Hands-On Guide to Relational Database Design, Second Edition, Addison Wesley, 2003.
- [31] definición de modelo entidad-relación para el diseño de base de datos <http://www.alegsa.com.ar/Dic/modelo%20de%20entidad-relacion.php>, marzo 2008.
- [32] Definición de Componente <http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/javaBeans/fundamento.htm>, marzo 2008.
- [33] [Pat Niemeyer](#), [Jonathan Knudsen](#), Learning Java, 3rd Edition, O'relly, United States of America, mayo 2005.
- [34] Definición Cliente / Servidor <http://www.csi.map.es/csi/silice/Global71.html>, marzo 2008.
- [35] C. Szyperski. Component Software. Beyond Object-Oriented Programming. Addison-Wesley, 2 ed, 2002.
- [36] Definición de DDE <http://www.casadomo.com/diccionario.aspx?id=3516&idm=&pat>, abril 2008.
- [37] G. Booch, J. Rumbaugh e I. Jacobson, El lenguaje Unificado de Modelos, Addison Wesley, Madrid, 1999.
- [38] Definición de OLE <http://www.alegsa.com.ar/Dic/ole.php>, abril 2008.
- [39] Definición de VBX <http://www.alegsa.com.ar/Diccionario/dic.php?palabra=VBX&Buscar> abril 2008.
- [40] Definición de ActiveX <http://www.alegsa.com.ar/Dic/activex.php>, abril 2008.
- [41] Definición de COM <http://www.microsoft.com/com/default.mspx>, abril 2008.
- [42] Definición de Framework <http://www.alegsa.com.ar/Diccionario/dic.php?palabra=framework&Buscar=Buscar>, abril 2008.

- [43] Definición de CORBA [http://www.alegsa.com.ar/Diccionario/dic.php?palabra=CORBA](http://www.alegsa.com.ar/Diccionario/dic.php?palabra=CORBA&Buscar=Buscar) &Buscar=Buscar, abril 2008.
- [44] Definición de CCM <http://www.omg.org/docs/formal/06-04-01.pdf>, abril 2008.
- [45] Definición de aferencia <http://www.definicion.org/aferencias-sensoriales>, abril 2008.
- [46] Definición de CBT o CBI <http://www.webopedia.com/TERM/C/CBT.html>, abril 2008.
- [47] Definición de Cinestesico [http://www.lubrano.com/cgibin/glosario/mostrar\\_contenido.cgi?codigo=881&termino=sentido%20cinest%C3%A9sico](http://www.lubrano.com/cgibin/glosario/mostrar_contenido.cgi?codigo=881&termino=sentido%20cinest%C3%A9sico), abril 2008.
- [48] Definición de CMS <http://www.bitpipe.com/tlist/Content-Management-Systems.html>, abril 2008.
- [49] Definición de coordinacion-oculo-manual <http://www.definicion.org/coordinacion-oculo-manual>, abril 2008.
- [50] Definición de cognoscitivismo [www.psicopedagogia.com/definicion/cognoscitivismo](http://www.psicopedagogia.com/definicion/cognoscitivismo), abril 2008.
- [51] Definición de GLC <http://www.msglobal.org/accessibility>, abril 2008.
- [52] Definición de Gnosis <http://www.definicion.org/gnosia>, abril 2008.
- [53] [Bill Kennedy](#), [Chuck Musciano](#), HTML & XHTML: The Definitive Guide, 5th Edition, O'Reilly August 2002.
- [54] Definición de LCMS <http://www.xyleme.com/>, abril 2008.
- [55] Definición de LOM y LTSC <http://ltsc.ieee.org/wg12/>, abril 2008.
- [56] Definición de IEEE <http://mexico.ihs.com/collections/ieee/index.htm?mid=w094>, abril 2008.
- [57] Definición de Lateralidad [http://www.espaciologopedico.com/articulos2.php? Id\\_articulo=219](http://www.espaciologopedico.com/articulos2.php?Id_articulo=219) abril 2008.
- [58] Tom Pender ,UML Bible, Wiley Publishing, Inc. 2003.

[59] Definición de Percepción Visoespacial <http://www.med.univ-rennes1.fr/iidris/cache/es/39/3975>, abril 2008.

[60] Definición de Perceptual [http://www.espaciologopedico.com/articulos2.php?Id\\_articulo=108](http://www.espaciologopedico.com/articulos2.php?Id_articulo=108), abril 2008.

[61] Definición de Praxia <http://www.espaciologopedico.com/recursos/glosariodet.php?Id=234>, abril 2008.

[62] Definición de Propioceptivo [http://www.otorrinoweb.com/izquie/temas/05.1equi/propioceptivo\\_7.htm](http://www.otorrinoweb.com/izquie/temas/05.1equi/propioceptivo_7.htm), abril 2008.

[63] Definición de Psicométrico <http://www.psicopedagogia.com/definicion/psicometrica>, abril 2008.

[64] Definición de Streaming [www.webopedia.com/TERM/s/streaming.html](http://www.webopedia.com/TERM/s/streaming.html), abril 2008.

[65] Significado de las siglas The World Wide Web Consortium (W3C) <http://www.w3.org>, abril 2008.

[66] Información con respecto al manejador de base de dato MySQL <http://www.mysql.com/>

[67] Definición de Visomotor <http://intercentres.cult.gva.es/spev13/HTML/Glosario/Psicopedagogico/VISOMOTOR.html> abril 2008.

[68] Cisco System, Inc. Cisco de Networking de Cisco Systems: Guia del primer año, segunda edición Traducción: KME Sistemas, S.L. Cisco System, Madrid 2002.

[69] Alejandro Peña Ayala y Juan Humberto Sossa Azuela, Educación Basada en Web: Un Estado del Arte, Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas; Centro de Investigación en Computación del Instituto Politécnico Nacional.

[70] Definición de Courseware [www2.plymouth.ac.uk/distancelearning/course/glossary.doc](http://www2.plymouth.ac.uk/distancelearning/course/glossary.doc), mayo 2008.

[71] Definición de *e-Learning* <http://dorisaranda-diario.blogspot.com/2006/10/conceptos-e-de-electronicos.html>, mayo 2008.

[72] Jorge Vasconcelos Santillán, Introducción a la computación, 5ta edición, Publicaciones cultural, México D.F., 2000.

[73] Definición de Apache <http://apache.org> septiembre 2008.

[74] Andy Ju An Wang Kai Qian, Component-Oriented Programming, a John Wiley & Sons, inc., publication, New Jersey Published simultaneously in Canada, 2005.

[75] Diccionario de la lengua española - Vigésima segunda edición [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=elaborar y/o armar](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=elaborar_y/o_armar), septiembre 2008.

[76] Julieta Noguez Monroy, María del Carmen Ferrer Nuñez, Informática, Preuniversitario Santillana, México, 2006.

[77] Trabajo de tesis "Software gnóstico-visual de apoyo para terapias aplicadas en niños con problemas de lectura" Ing. Jeanett Figueroa Martínez. Facultad de Estudios Profesionales "Aragón",

[78] Ernesto Peñaloza Romero, Fundamentos de programación, Campus Aragón, UNAM, México D.F, 2001.

[79] Definición de Apache Tomcat <http://java.sun.com/products/jsp/tomcat/> septiembre 2008.

---

---

## ANEXO A

### MANUAL DEL USUARIO

---

---

**En este anexo se describe el funcionamiento de las actividades y/o servicios que brinda el sistema desarrollado.**

## MANUAL DEL USUARIO

A continuación se describe un material informativo en el que se recogen las actividades del servicio y las funciones del sistema desarrollado.

El primer paso es validar la información del usuario, por lo que en la pantalla que se muestra en la Figura A1, se tienen que poner el nombre y password para que el sistema identifique el tipo de usuario que va acceder al mismo, el sistema hace una búsqueda en la base de datos, si no existe el usuario se le notifica que no podrá acceder al sistema, pero, si el usuario existe, lo manda a la página que corresponde a las acciones del tipo de usuario.

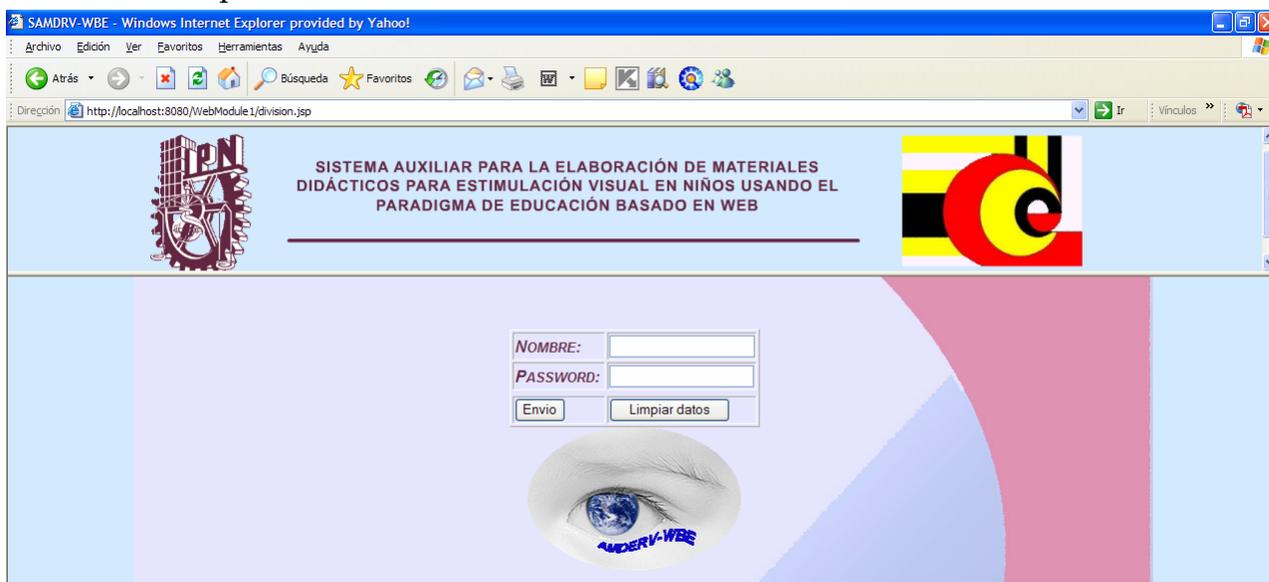


Figura A1. Página de entrada para acceso al sistema.

Según el tipo de usuario aparecerán distintos servicios.

### Servicio para el administrador

Cuando el administrador es verificado en la base de datos, envía la vista de la Figura A2. En esta página el administrador tiene las posibilidades de dar de alta, baja o cambios en los datos de otros usuarios, para ello hay que dar clic en las ligas de lado izquierdo que se ven en la Figura A2.

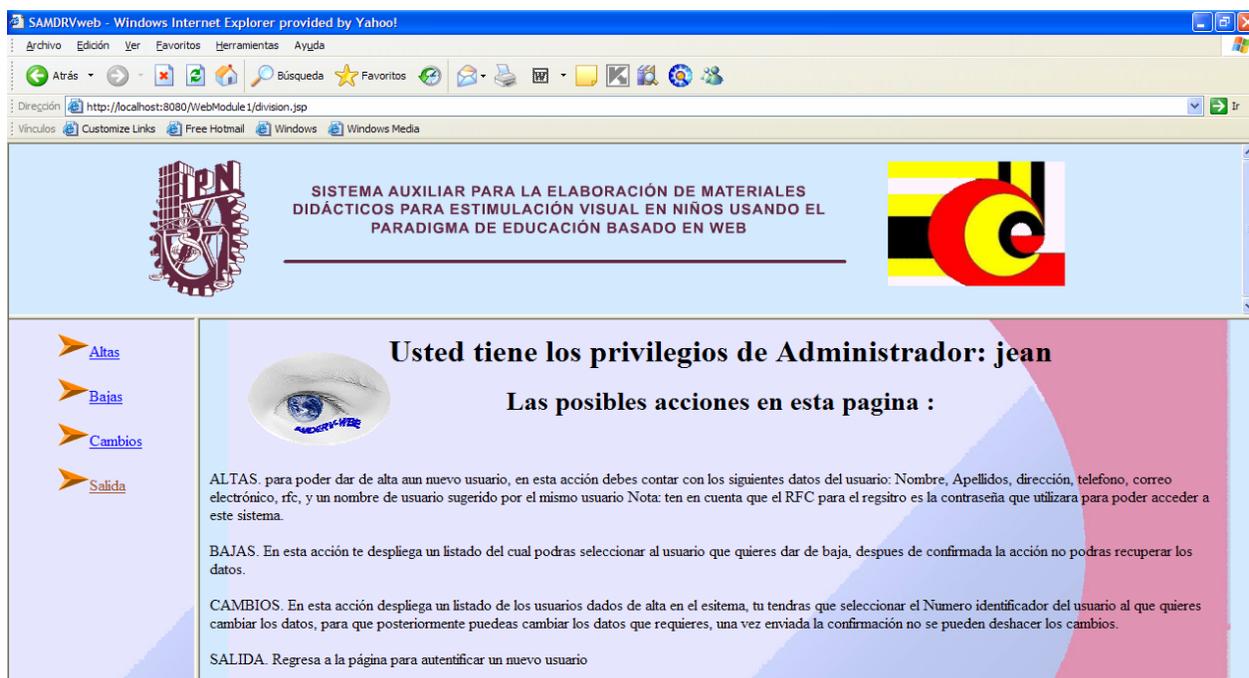


Figura A2. Página de acciones para el administrador

Al hacer clic en Altas se desplegará la página de la Figura A3 en la que se proporcionan los datos para un nuevo usuario, en esta acción se debe contar con los siguientes datos del usuario: Nombre, Apellidos, dirección, teléfono, correo electrónico, rfc, y un nombre de usuario sugerido por el mismo usuario. Hay que tener en cuenta que el RFC para el registro es la contraseña que se utilizara para que se pueda acceder a este sistema.

Si se hace clic en Bajas se presenta la pagina de la Figura A4. En esta acción se despliega un listado en el cual se podrá seleccionar al usuario que se quiere borrar, ya que se selecciono el usuario dando un clic sobre el id\_usuario se envía una página que confirma que el usuario fue borrado, después de confirmada la acción no se podrán recuperar los datos.

Figura A3. Página que muestra el formato de altas.

id_usuario	Nombre	Apellido Paterno	Apellido Materno	tipo de usuario
1	Marisol	Figuroa	Zepeda	alumn
2	Manuel	Rivera	Suarez	alumn
3	Ruben	Peredo	Valderrama	profe
4	jeanett	Figuroa	Martinez	admin
5	israel	arriaga	fuentes	admin
6	Fabiola	Figuroa	Martinez	profe
7	Antonio	ruiz	ramirez	alumn
8	Sergio	Sandoval	Reyes	profe
9	Julissa	Figuroa	Martinez	profe
10	Alma Patricia	Lopez	Hernández	profe
11	Adan	Ramos	Bautista	profe

Figura A4. Página que muestra el formato de bajas de los usuarios.

Al acceder a Cambios se despliega un listado de los usuarios dados de alta en el sistema, aquí se seleccionará el id\_usuario al que se le quiere cambiar los datos, para que posteriormente se despliegue la información del usuario que se requiere cambiar, como se ve en la Figura A5, una vez enviada la confirmación no se pueden deshacer los cambios.

SAMDRV-WBE - Windows Internet Explorer provided by Yahoo!

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://localhost:8080/WebModule1/division.jsp

SISTEMA AUXILIAR PARA LA ELABORACIÓN DE MATERIALES DIDÁCTICOS PARA ESTIMULACIÓN VISUAL EN NIÑOS USANDO EL PARADIGMA DE EDUCACIÓN BASADO EN WEB

Modifica los datos del usuario por favor

Nombre : jeanett RFC : fmj750529

Ap. Paterno : Figueroa Ap. Materno : Martinez

Direccion : valle

C.P. : 55280 Municipio : ecatepec

Telefono : 57803755 E-mail : jeanettfm@yahoo.com

Datos de Usuario de SGP

Usuario : jean Contraseña : .....

Listo Intranet local

Figura A5. Página que muestra el formato para modificar la información del usuario.

## Servicio para el profesor

Como se muestra en la Figura A6 aparece un menú con dos ligas, la primera muestra un listado con los materiales elaborados, para poder ser ejecutados por el profesor, mientras que la segunda permite el armado de un nuevo material didáctico.

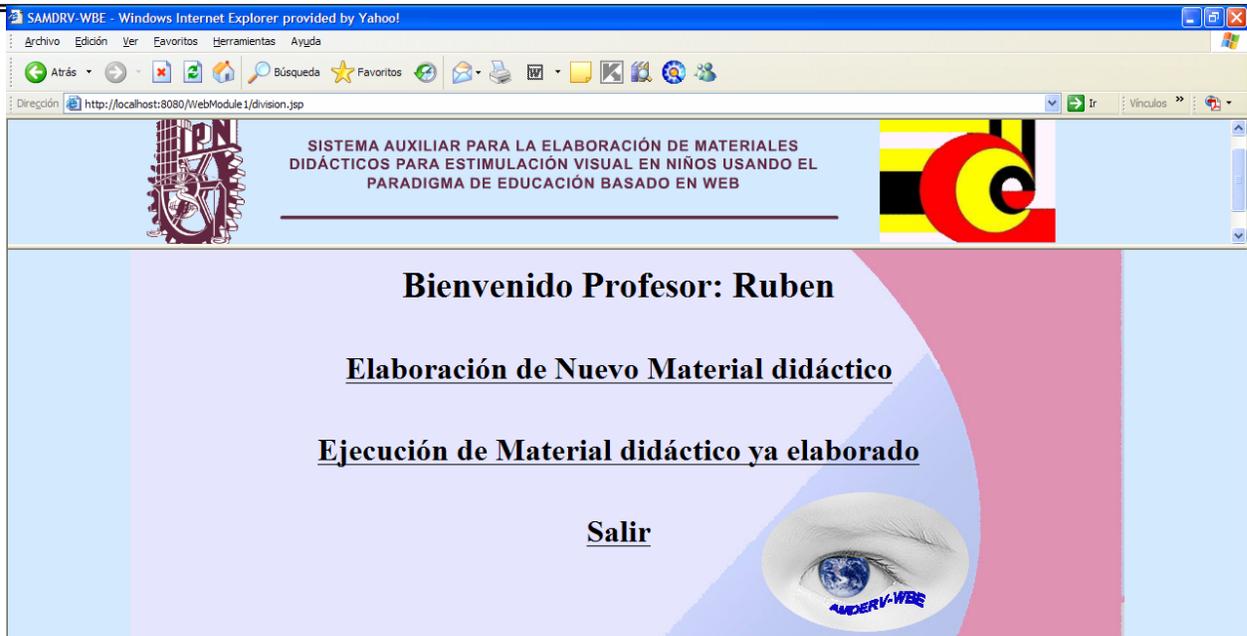


Figura A6. Página que muestra el menú de opciones para el usuario profesor.

### Elaboración de Nuevo Material didáctico

Para la elaboración de material didáctico se despliega la pantalla de la Figura A7, aquí lo primero que se coloca es el nombre o título del material didáctico, el cual se guardará en la base de datos, para que posteriormente, si es creado, se pueda visualizar para su ejecución.

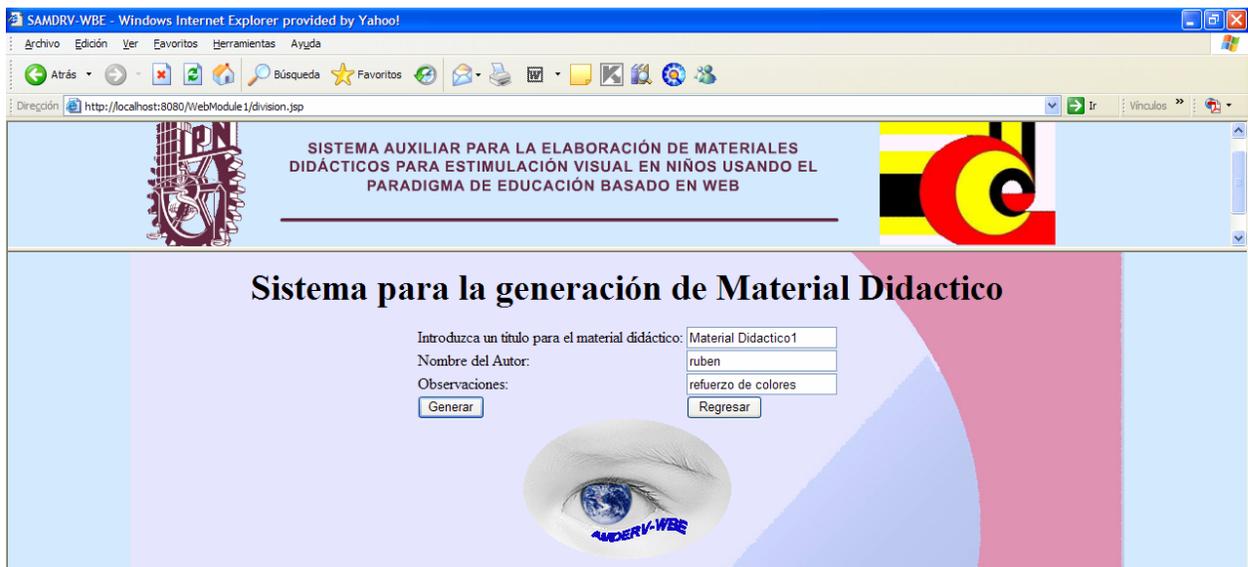


Figura A7. Página que muestra la opción para introducir el nombre al material didáctico.

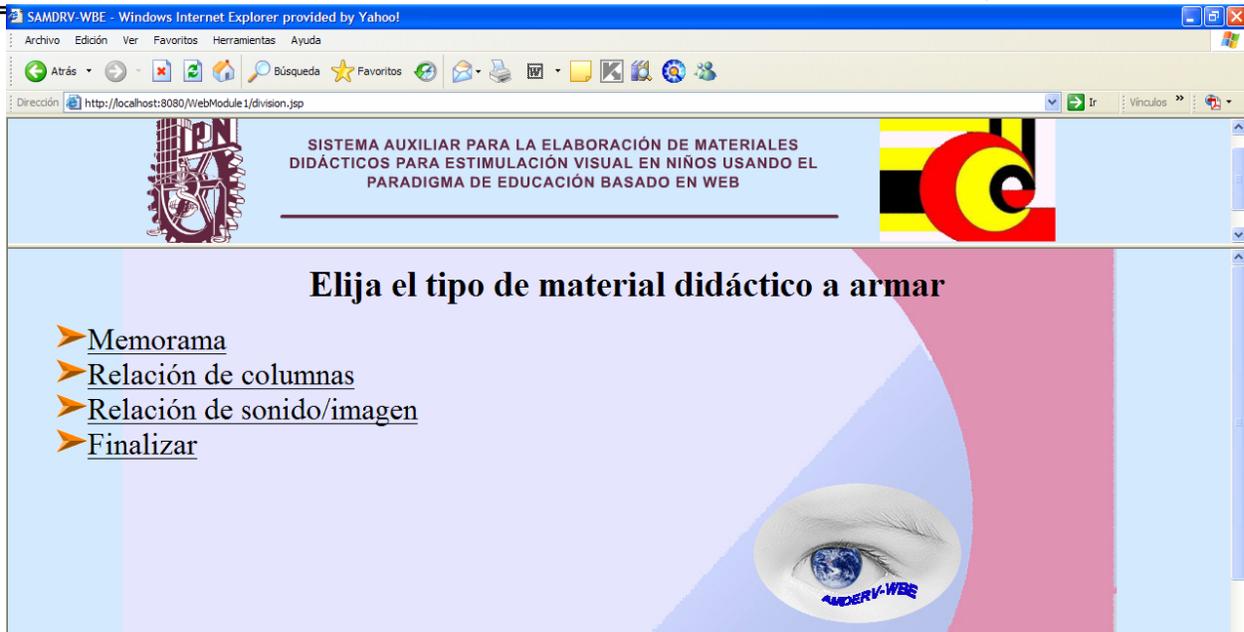


Figura A8. Menú para la creación de los diferentes materiales didácticos.

### Creación del Material Didáctico Memorama

Este material didáctico tiene como finalidad presentar un conjunto de imágenes tapadas, para que el usuario encuentre los pares de imágenes iguales. Estas imágenes tienen formato jpg; se tienen que elegir las diferentes imágenes que lo van a conformar, por lo que se despliega la página de la Figura A9. Las cajas de texto tienen la acción de explorar para buscar en las diferentes carpetas los archivos de imagen, una vez que son llenados estos datos se oprime el botón guardar; el cual nos mostrará la pagina con una vista preliminar del memora con las figuras elegidas como se muestra en la Figura A10.

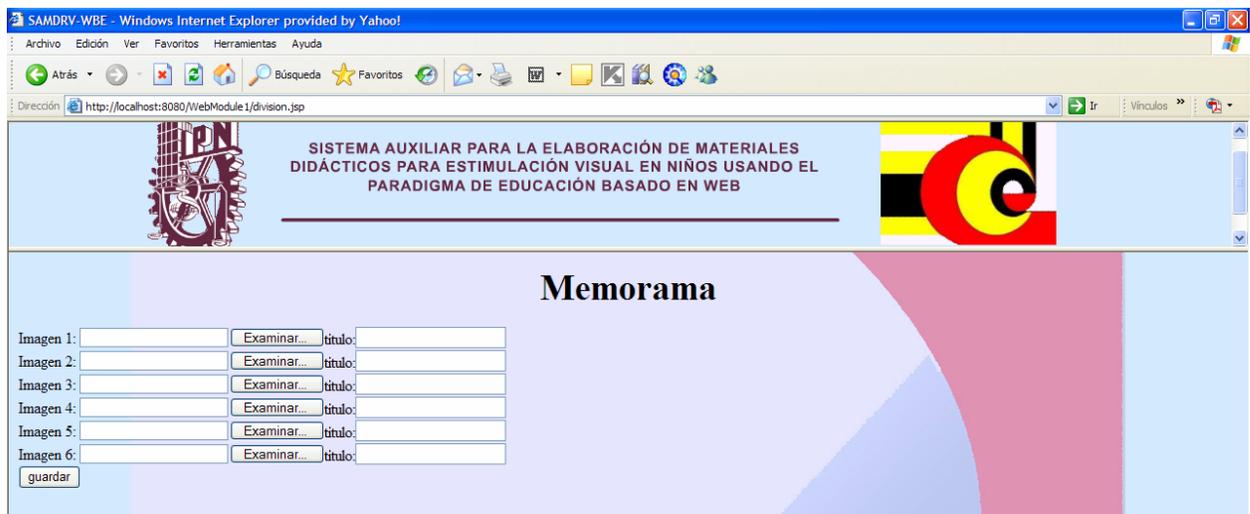


Figura A9. Muestra el formato para elegir las imágenes para el memorama.

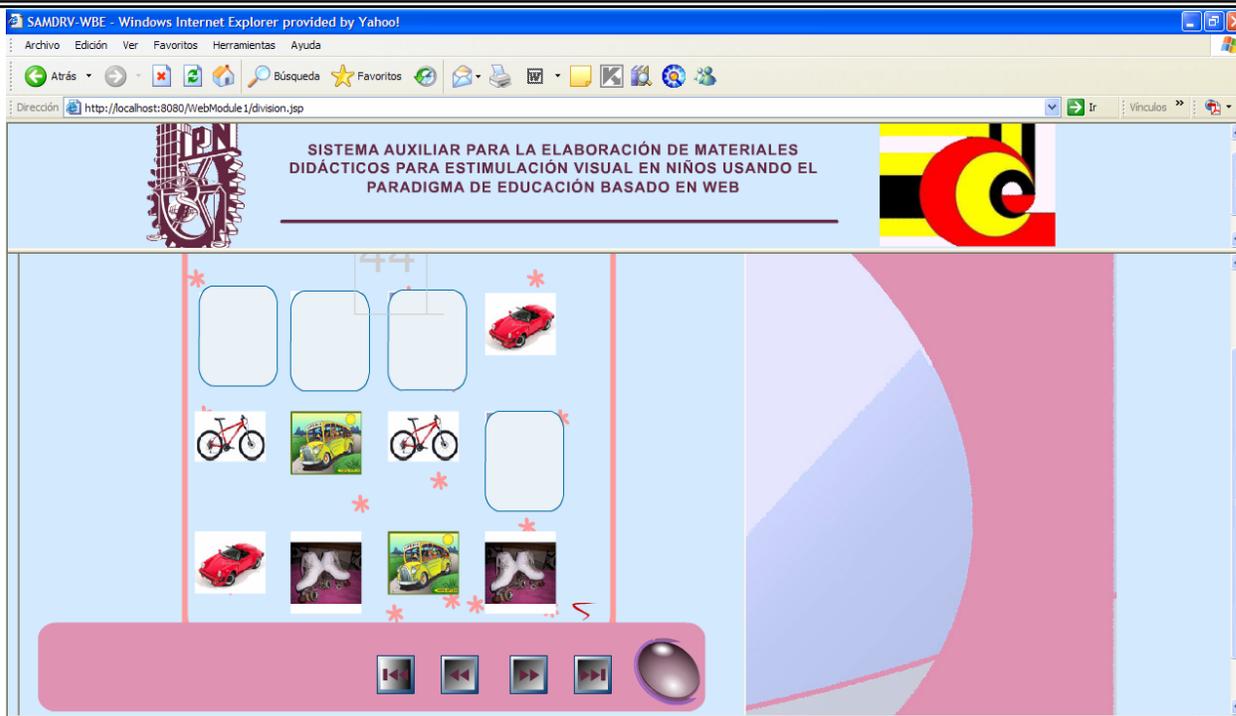


Figura A10. Página que muestra la vista preliminar del memorama con las figura elegidas.

### Creación del Material Didáctico Relación de Columnas

columnas de imágenes para que relacionar la imágenes de la columna de la izquierda trasladando estas con la figura de la columna derecha depositándolas en el lugar correspondiente de esta manera se tienen que colocar el texto de la pregunta para esta sección, después se seleccionaran 8 imágenes que se guardaran utilizando las cajas de dialogo, con la posibilidad de buscarla a través del explorador de Windows a cada imagen se le puede colocar un texto que describa a la imagen como se visualiza en la Figura A11.

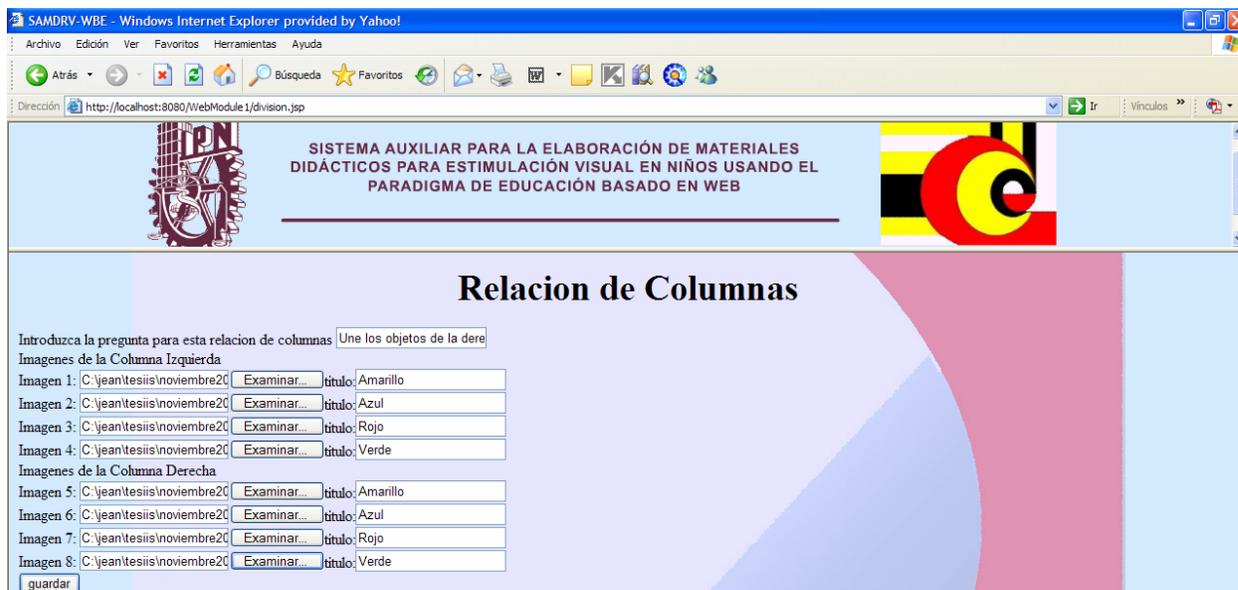


Figura A11. página que muestra el formato para llenar la pregunta, las imágenes, su descripción para el componente relación columna.

Al dar clic en el botón guardar se obtiene la página de vista previa del material didáctico como se visualiza en la Figura A12, posteriormente si presiona aceptar el Material didáctico será guardado y nos regresa al menú de creación de materiales didácticos, regresar nos vuelve a presentar la página de la Figura A8.

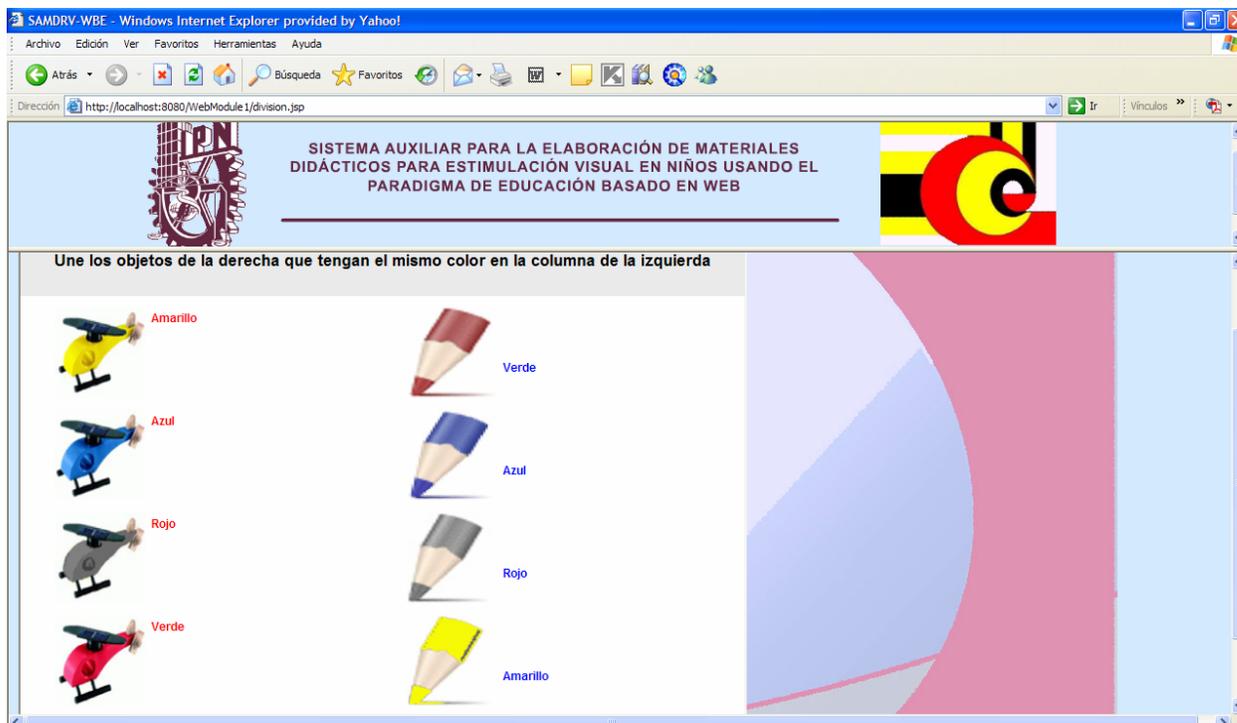


Figura A12. Página que muestra la vista previa del componente relación columna.

## Creación del Material Didáctico Relación de Sonido con Imagen

El material didáctico relación de sonido con imagen tiene el objetivo de presentar una serie de imágenes entre las cuales hay que seleccionar, la indicada por la pregunta, que esta relacionada con el sonido, la página que se despliega para poder llenar la información para este material esta descrita por la Figura A13 en donde se teclea el texto para la pregunta, se selecciona el sonido que se quiere escuchar con la opción de buscar el archivo de sonido a través del explorador de Windows, en conjunto se tienen que seleccionar seis imágenes, una tiene que estar relacionada con la pregunta y el sonido, para saber que imagen es la correcta se tiene una sección de botones de radio en la que se indica cual es la imagen que concuerda con la pregunta y el sonido. Ya que se lleno este formato al oprimir el boton guardar se despliega la vista previa del componente como se ve en la Figura A14. Si se presiona aceptar el Material didáctico será guardado y nos regresa al menú de creación de materiales didácticos, regresar nos vuelve a presentar la página de la Figura A8.

The screenshot shows a web browser window with the following content:

**SISTEMA AUXILIAR PARA LA ELABORACIÓN DE MATERIALES DIDÁCTICOS PARA ESTIMULACIÓN VISUAL EN NIÑOS USANDO EL PARADIGMA DE EDUCACIÓN BASADO EN WEB**

### Relación de sonido / imagen

Texto de la pregunta:

Archivo de sonido:

Imagen 1:

Imagen 2:

Imagen 3:

Imagen 4:

Imagen 5:

Imagen 6:

Seleccione la imagen que corresponde al sonido:  Imagen1  Imagen2  Imagen3  Imagen4  Imagen5  Imagen6

titulo:

titulo:

titulo:

titulo:

titulo:

titulo:

Figura A13. Página que muestra el formato para llenar los datos del componente relación de sonido con imagen.



Figura A14. Página que muestra la vista previa del componente relación de sonido con imagen.

### Servicio para el alumno

El servicio que brinda el sistema en la sección de alumnos es únicamente para ejecutar materiales didácticos previamente elaborados por profesores, por lo que se despliega un listado con los materiales didácticos para que el alumno seleccione el que desee resolver dando un clic sobre el nombre del material seleccionado, como se ve en la Figura A15.

Posteriormente, se desplegará el material didáctico con toda la serie de ejercicios que este contenga, y se navegará a través de ellos oprimiendo los botones de desplazamiento que aparecen en la parte inferior, el ovalo muestra la calificación obtenida de este material didáctico, como se ve en la Figura A16.

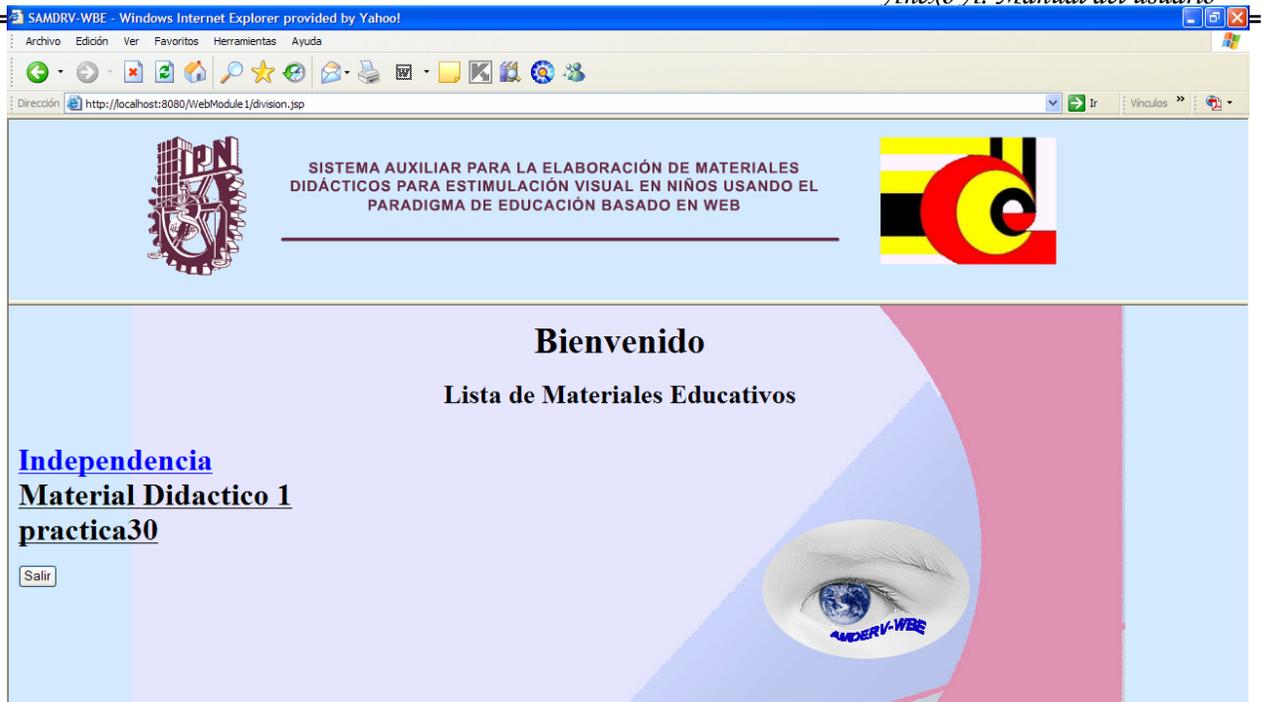


Figura A15. Página que despliega en forma de lista todos los materiales didácticos que pueden ejecutar los alumnos.

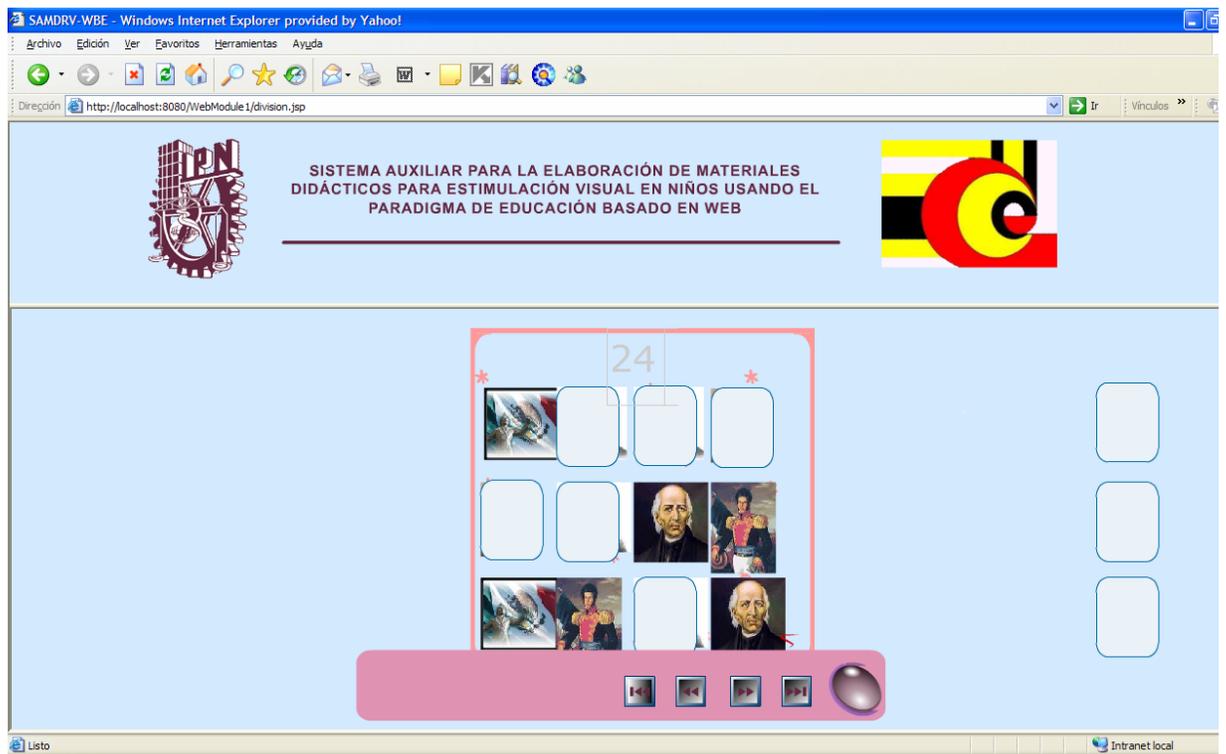


Figura A16. Página que despliega la ejecución de un material didáctico.

---

---

## ANEXO B

### ESPECIFICACIONES DE HARDWARE Y SOFTWARE

---

---

**Como un software se ejecuta mediante un hardware es necesario saber sus características para que el programa o sistema trabaje adecuadamente.**

## REQUERIMIENTOS DE HARDWARE Y SOFTWARE

Con antelación se mencionó que el sistema esta basado en la arquitectura Cliente / Servidor motivo por el cual se señalan las características para el Servidor y para el Cliente, y de igual forma los requerimientos de hardware que se utilizan en el Servidor para la implementación del sistema AMDERV-WBE.

Características de hardware mínima para el Servidor:

- Computadora Centrino Duo velocidad de los procesadores a 1.66GHz.
- Memoria 1 Gb.
- Disco Duro 100 Gb.
- Tarjeta de Red 10/100/1000.
- Conexión a Internet.

Características de software para el Servidor:

- Sistema Operativo Windows XP.
- Servidor de aplicaciones Apache Tomcat ver 5.1.
- Manejador de Base de datos MySQL 5.0.41 –community-nt via TCP/IP.

Características de hardware mínima para el Cliente:

- Tarjeta de red 10/100.
- Procesador.
- Memoria.
- Disco Duro (puede o no tener).
- Conexión a Internet.

Características de software para el Cliente:

- Puede utilizar desde terminales hasta computadoras personales, lo indispensable es que pueda ejecutar un programa de navegación por Internet, ya sea Mozilla, Internet Explore, Konqueror, etc., por lo que, no depende de un sistema operativo específico.

---

---

## ANEXO C

### CUESTIONARIOS CONTESTADOS POR PROFESORES

---

---

**Para llegar a los resultados del sistema AMDERV-WBE implementado se aplicó un cuestionario a 10 profesores que imparten clases al primer año, nivel primaria, y sus respuestas se encuentran en este anexo.**

Datos Generales

Nombre: ABACUC CONTRERAS JAIMES  
Edad: 60 AÑOS Nivel de estudios:  
Lugar donde labora: Esc. Prim. "Profr. Mario Rojas Aguilar"

I.- En el siguiente cuestionario, por favor conteste las preguntas de acuerdo con su experiencia docente

1.- ¿El software que observo, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias haría para mejorar el software?

Es un material muy bueno que nos ayuda a tener creatividad, para estimular a los niños. Sobretodo a niños con estilo de aprendizaje visual.

Datos Generales

Nombre: EVA MARIA GARCIA AVARADO

Edad: 48 Nivel de estudios: LICENCIATURA

Lugar donde labora: ESCUELA PARTICULAR "DELTA"

I.- En el siguiente cuestionario, por favor conteste las preguntas de acuerdo con su experiencia docente

1.- ¿El software que observo, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias haría para mejorar el software?

REALMENTE SERIA BUENO QUE UN SOFTWARE ASI ESTUBIERA EN  
LAS ESCUELAS PARA AYUDARNOS A REAFIRMAR CONOCIMIENTOS,  
AUNQUE SE VE SENCILLO SERIA EXLENTE QUE NOS CAPACITARAN  
EN COMPUTO

### Datos Generales

Nombre: Nónica Dianney León Favela.

Edad: 20 años. Nivel de estudios: Licenciatura en Pedagogía.

Lugar donde labora: Escuela Primaria Lic. Benito Juárez.

### I.- En el siguiente cuestionario, por favor conteste las preguntas con respecto a su experiencia docente

1.- ¿El software que observo, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias harías para mejorar el software?

Sería bueno adaptar un instructivo para el uso de dicho software  
para luego en ocasiones se complica la forma de usarlo y en  
caso de que así fuese pues ya esta la "AYUDA" para no  
perder la secuencia de lo que se esta haciendo.

Datos Generales

Nombre: Yessica Monserrat Contreras Torres.  
Edad: 32 años Nivel de estudios: Lic. en Educ. Prim.  
Lugar donde labora: Escuela Primaria "José Vasconcelos".

I.- En el siguiente cuestionario, por favor conteste las preguntas de acuerdo con su experiencia docente

1.- ¿El software que observo, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias haría para mejorar el software?

Es un material muy bueno que nos ayuda a tener creatividad, para estimular a los niños. Sobretudo a niños con estilo de aprendizaje visual.

Datos Generales

Nombre: Ana María Torres y Martínez  
Edad: 61 años Nivel de estudios: Profesora Educ. Prim.  
Lugar donde labora: Lse. Prim Urb. "Maestro de América"

I.- En el siguiente cuestionario, por favor conteste las preguntas de acuerdo con su experiencia docente

1.- ¿El software que observo, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias haría para mejorar el software?

Lo importante de este software es que el profesor lo practique con sus alumnos y con ellos enriquezca su contenido con valiosos ejercicios de acuerdo donde vivan y se inicie.

## Datos Generales

Nombre: FABIOLA MORENO GARCIA  
Edad: 29 AÑOS Nivel de estudios: LIC. EN PEDAGOGIA  
Lugar donde labora: ESCUELA PRIMARIA PAIDEIA

---

### I.- En el siguiente cuestionario, por favor conteste las preguntas con respecto a su experiencia docente

1.- ¿El software que observo, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias harías para mejorar el software? Encuentro muy completo el software presenta ejercicios y actividades diseñados para el aprendizaje y desarrollo de habilidades para asimilar nuevos conocimientos, la cual se apoya en la moderna pedagogía operatoria. Solamente recomendaría ampliar la galería de imágenes y sonidos para facilitar la elaboración de material y ahorrar tiempo al buscar por medio de Internet.

---

---

---

---

---

## Datos Generales

Nombre: ALMA

Edad: 29 AÑOS

Nivel de estudios: PASANTE MAESTRÍA

Lugar donde labora: CCH AZCAPOTZALCO, CENTRO CULTURAL Y RECREATIVO LOMAS

### I.- En el siguiente cuestionario, por favor conteste las preguntas de acuerdo con su experiencia docente

1.- ¿El software que observó, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias haría para mejorar el software?

Desconozco si en alguna parte del software se hacen sugerencias de las variantes que se pueden hacer con la combinación de ejercicios como las que me mostraste, considero que sería conveniente aludir ideas o temáticas específicas, partir de ejemplos que desarrollen la creatividad.

**¡¡SUERTE!!!**

### Datos Generales

Nombre: Ed Garcia Jacmin Hayder

Edad: 22 Nivel de estudios: Lic Pedagogia

Lugar donde labora: Escuela Primaria Luz Cabrer G

### I.- En el siguiente cuestionario, por favor conteste las preguntas con respecto a su experiencia docente

1.- ¿El software que observo, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias harías para mejorar el software?

Esta completo solo faltaria a que tuviera mas juegos para que no se aburrieran los niños de siempre ver los mismo juegos.

Datos Generales

Nombre: Israel Arriaga Fuentes  
Edad: 38 Nivel de estudios: Licenciatura  
Lugar donde labora: Escuela primaria Dr. Gustavo Baz Prada

I.- En el siguiente cuestionario, por favor conteste las preguntas de acuerdo con su experiencia docente

1.- ¿El software que observo, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias haría para mejorar el software?

Sería conveniente agregar más tipos de materiales didácticos

## Datos Generales

Nombre: Julio Cesar Castillo Cañedo

Edad: 33 AÑOS Nivel de estudios: candidato a Maestro En Ciencias

Lugar donde labora: Escuela Primaria Antonieta Rivas Mercado, Oriente 127 No. 208 Col. Moctezuma 2da. Secc. México D.F.

**I.- En el siguiente cuestionario, por favor conteste las preguntas de acuerdo con su experiencia docente**

1.- ¿El software que observó, le sirve a un profesor para armar material didáctico, orientado a la estimulación visual?

Si  No

2.- ¿El software le facilita a un profesor el armado de material didáctico, orientado a la estimulación visual?

Si  No

3.- ¿Con este software, el profesor puede armar material didáctico para estimular Coordinación Visomotriz?

Si  No

4.- ¿Con este software, el profesor puede armar material didáctico para estimular Constancia Perceptual?

Si  No

5.- ¿Con este software, el profesor puede armar material didáctico para estimular Figura a Fondo?

Si  No

6.- ¿Con este software, el profesor puede armar material didáctico para estimular Posición en el Espacio?

Si  No

7.- ¿Con este software, el profesor puede armar material didáctico para estimular Relación Espacial?

Si  No

8.- ¿El profesor puede armar material didáctico en su idioma y con su acento?

Si  No

9.- ¿El software le da flexibilidad para adaptar los materiales didácticos, por ejemplo: cambiando los textos, imágenes y sonidos?

Si  No

10.- ¿El software realiza un proceso entre el profesor y la computadora de tal forma que haya estímulo y respuesta?

Si  No

11.- ¿El software le proporciona las instrucciones que le indican como manejarlo?

Si  No

12.- ¿Usted utilizaría este software para estimular el reconocimiento visual?

Si  No

13.- ¿Usted recomendaría este software a otro profesor?

Si  No

14.- ¿Que sugerencias haría para mejorar el software?

El inconveniente es que no cuento con computadora y en la escuela que laboro a nivel primaria no cuenta con computadoras, por lo que no habria forma de utilizar el software, y creo que el software si lo podrian utilizar otros profesores siempre y cuando cuenten con computadora.

|