



UNIVERSIDAD DE LAS CIENCIAS
INFORMÁTICAS

MÓDULO CUENTO DEL PRODUCTO “MIS MEJORES CUENTOS”

TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS.

AUTORES:

ANA MARGARITA CLARO SÁNCHEZ
ARIEL RODRÍGUEZ ABREU

TUTOR:

ING. YASMANI CEBALLOS IZQUIERDO

CO-TUTOR:

ING. LUIS FELIPE DÍAZ BARRIOS

CIUDAD DE LA HABANA, ENERO DE 2011.

HAY UNA FUERZA MOTRIZ MÁS PODEROSA QUE EL VAPOR, LA
ELECTRICIDAD Y LA ENERGÍA ATÓMICA: LA VOLUNTAD.

ALBERT EINSTEIN



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo Módulo Cuento del producto “Mis Mejores Cuentos” y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores:

Ana Margarita Claro Sánchez

Ariel Rodríguez Abreu

Autores:

Ing. Yasmani Ceballos Izquierdo

Ing. Luis Felipe Díaz Barrios



AGRADECIMIENTOS

Q mis padres por su absoluto apoyo en estos años de universidad, por estar siempre orgullosos de sus hijos, por aceptar y apoyar mis decisiones y por dejarme ser yo sin presiones. Gracias por su confianza.



A mi hermanito lindo Geysler, gracias por ser tan exigente conmigo, tu papel de hermano mayor me ha convertido en mejor persona.

Un especial agradecimiento a mi novio Julio César, por ser siempre incondicional. Por ser amigo por encima de todo y por su inmensa paciencia. Gracias por quererme, por ayudarme en todo, sabes que sin ti no estuviera graduándome ahora. Gracias también por estar siempre en los momentos difíciles, cuando todo estaba en contra mía, tú fuiste el único que supo comprenderme y ayudarme a levantar la cabeza sin vergüenza. Nada de lo que pueda escribir será suficiente para agradecer todo lo que haces por mí a diario. Te amo “cosi”. Gracias a su familia porque para ellos soy un miembro más.

A mi familia por estar siempre dispuesta a complacerme y hacer más feliz mi estancia en la universidad. Gracias porque sé que están muy orgullosos de mí. Nani, Maricelis, Claudia y Yosvi, mis chiquiticos lindos que siempre me brindan hasta lo que no tienen, para que yo esté a gusto.

A mis mejores amigos de siempre Darlén, Lily, Alejandro, Lisi y Lía, los quiero mucho, gracias por permitirme colarme en sus corazones.

A mi compañero de tesis Ariel, por animarme en días difíciles y por hacerme ver la parte buena de todas las cosas malas que nos sucedió durante la realización de esta investigación.

A los que quedan del 8102 y a los que no están también, siempre van a ser mi grupo favorito. A las chicas de mi apartamento y a los chicos también, por ser una gran y divertida familia disfuncional. A todas las personas que de alguna manera han formado parte de mi vida universitaria y han hecho que estos cinco años no los pueda olvidar nunca.



Q mi novia linda, por ser quien es y sobre todo por ser quien soy junto a ella; por pasar junto a mi mis tormentos y noches de estudios, por aguantar y saber llevar mis pesadeces y malos genios. Por llenar todos mis días de amor y chicha con solo estar ahí, y ser esa personita importante que no te puede faltar y quieres siempre junto a ti. También a mis suegros y Yaimi por permitirme ser parte de ellos.

A mis hermanos Mabel y Ariel por apoyarme como hermano, como persona, por demostrar lo mucho que me quieren y lo que valgo para ellos y saber que para lo que sea estarán ahí. A mis otros hermanitos Jose, Pedro, Yadira y Yahima por ser mis hermanos de corazón y alma, aunque no llevemos la misma sangre, pero si el mismo amor los uno por los otros.

A mi tíos, Idalmis, Aurora y Jorge, por confiar y sentirse orgullosos de mi, y ser uno más de sus hijos, así como anhelar y disfrutar de mi logro como mis padres. A mis primos Jorgito y Yoli, por ser compañeros de vida, por compartir conmigo alegrías y reveses, por hacer suyos mis padres cuando yo no estaba, por ser como mis hermanos. A mi prima por ser mi meta como docente y profesional, e impulsarme a romper sus logros. A toda mi familia en general.

A mi compañera de tesis por ser insistente y preocupada, pasar juntos momentos de triunfos y reveses, y ser una genial persona, aunque no me invitara a probar su rico sazón. Por compartir y lograr conmigo este triunfo decisivo en nuestras carreras.

A Sita por ser mi mamita linda en la UCI, y por una amiga como ya no vienen, en las buenas y las malas, por entender mis pesares y locuras, por compartir y vivir conmigo sus logros, y hacerme feliz con solo estar ahí. A mis compañeros de un aula que no se separa, Adrian, Yerandi, Ifigenia, Pachi, a todos los del 4105 que fueron y serán mi grupo.

A los Paz por estar ahí y ser una etapa inolvidable en la universidad, por las muchas horas de amistad, por ser diferentes pero unidos, Fredy, Zami, Hismel, etc.



DEDICATORIA

A mi abuelito Pipo, para que, donde estés, te sientas orgulloso de tu nieta, que te quiere mucho, y no se acostumbra a estar sin ti.

A mi hermanito Geysler, mi carrera te la debo a ti, y sé que si estuvieras aquí disfrutarías mucho este momento. Aquí está tu sueño, tu hermanita chiquita se hace Ingeniera. Te quiero mucho.

A mis padres por su sacrificio durante estos 5 años. Ma, eres genial, así con tus cosas, que te hacen ser la madre más preocupada y dulce del mundo. Pa, es fantástico ser para ti todavía tu niña pequeña, eres el mejor dando aunque casi nunca recibes nada. Este es mi regalo por todo lo que haces por mí a diario. Los quiero mucho.

Anita

A mis padres Ariel Rodríguez Leyva y Rosa María Abreu Gamboa por haber sido los padres que fueron, son y serán por siempre; por haber cursado junto a mi 5 años de sacrificio, preocupación, desvelo, estudio, ternura y amor; por haber confiado en mí y hacer mío su sueño de un hijo universitario, cuando todo empieza por complacer a tus padres y terminas haciendo tuyo el sueño de ser alguien en la vida por ti mismo. Por ser todos uno, sin primeros, ni segundos, un logro de familia. Por ser este el primer paso para un día ser como ellos y hacer sentir a mis hijos tan orgullosos como yo lo estoy de ellos.

Ariel



RESUMEN

En la actualidad la utilización de multimedia como método de enseñanza a nivel mundial ha demostrado ser de gran ayuda como material de apoyo para el desarrollo de habilidades en los estudiantes, favoreciendo los procesos de enseñanza y aprendizaje grupales e individuales. El sistema de enseñanza cubano no cuenta con la cantidad de productos multimedia suficientes para el desarrollo de habilidades de los niños de 3 a 5 años de edad. Debido a esto el proyecto “Producción de Recursos Didácticos” de la Universidad de las Ciencias Informáticas (UCI) asume el reto de desarrollar el producto “Mis Mejores Cuentos”, con la utilización de herramientas libres, que contenga los cuentos tradicionales más gustados por los niños de la edad de 3 a 5 años.

En el producto multimedia se presentarán los cuentos infantiles los cuales deben ser de dominio público y organizados por autores, de ahí la necesidad de desarrollar un módulo que contenga la información que se desea presentar.

En el presente trabajo se hace un estudio de las tendencias actuales para el desarrollo de este tipo de aplicaciones. Se realiza un análisis y selección de las tecnologías adecuadas para el desarrollo del módulo, se identifican las funcionalidades que debe cumplir este, y se realiza el diseño del mismo. A partir de los artefactos obtenidos se procede a su implementación y validación.



ABSTRACT

Nowadays the use of multimedia applications as a teaching globally method has proved very helpful as a support for the development of abilities in students, encouraging the group and individual teaching-learning process. The Cuban education system lacks the sufficient amount of multimedia products for the development of skills for children from 3 to 5 years old. As a result, the project “Producción de Recursos Didácticos” of the University of Informatics Sciences (UCI) has accepted the challenge of developing the product “Mis Mejores Cuentos”, using of free tools that contains the most beloved fairy tales by children between 3 and 5 years old.

The multimedia product will present free copyrighted children's stories organized by author, for this reason, raise the need to develop a module that contains this information.

In this investigation we study the current trends for developing multimedia applications. An analysis and selection of appropriate technologies for the implementation of the module, the identification of the functional requirements, as well as the design, was made. From the artifacts obtained are applicable to its implementation and validation.

INTRODUCCIÓN.....	1
CAPÍTULO I FUNDAMENTACIÓN DEL TEMA	5
1.1 Introducción.....	5
1.2 Software educativo.....	5
1.3 Tecnología multimedia.....	6
1.4 Metodologías de desarrollo de software	8
1.4.1 Rational Unified Process (RUP).....	9
1.5 Lenguajes de modelado.....	12
1.5.1 Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L) 1.5.....	12
1.6 Herramientas CASE.....	15
1.6.1 Visual Paradigm 6.4	15
1.7 Lenguajes de programación web.....	16
1.7.1 Lenguaje Javascript.....	16
1.8 Sistema de Control de Versiones	17
1.8.1 Subversion (SVN).....	17
1.9 Servidor web.....	18
1.9.1 Servidor Apache 2.2.....	18
1.10 Entorno de Desarrollo Integrado (IDE)	19
1.10.1 Aptana Studio 2.0.5.....	19
1.11 Conclusiones.....	20
CAPÍTULO II ANÁLISIS Y DISEÑO DEL MÓDULO	21
2.1 Introducción.....	21
2.2 Modelo de Dominio.....	21
2.2.1 Identificación de los conceptos del dominio.....	22
2.3 Requerimientos del Sistema	23
2.3.1 Requerimientos Funcionales.....	23
2.3.2 Requerimientos No Funcionales	24
2.4 Vista de Gestión del Modelo y sus subsistemas	26
2.5 Vistas de Presentación.....	27
2.5.1 Descripción textual de las Vistas de Presentación.....	27
2.5.2 Diagramas de Estructura de Presentación.....	34

ÍNDICE DE CONTENIDOS

2.5.3	Diagramas de Estructura de Navegación.....	36
2.6	Diagrama de clases	37
2.7	Diagramas de Secuencias.....	40
2.8	Conclusiones.....	41
CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA.....		43
3.1	Introducción.....	43
3.2	Modelo de Implementación	43
3.2.1	Diagrama de Componentes	44
3.2.2	Diagrama de Despliegue	45
3.3	Validación de la solución propuesta	46
3.3.1	Métodos de Prueba	46
3.3.2	Pruebas aplicadas al software	48
3.4	Conclusiones.....	49
CONCLUSIONES GENERALES		50
RECOMENDACIONES.....		51
REFERENCIAS BIBLIOGRÁFICAS		52
GLOSARIO DE TÉRMINOS		57

Figura 1. RUP en dos dimensiones.....	11
Figura 2. Lenguajes básicos de ApEM-L.....	13
Figura 3. Modelo Conceptual o Modelo de Dominio	23
Figura 4. Representación de la Vista de Gestión del Modelo	26
Figura 5. DEP Vista de Presentación <i>Cuentos</i>	35
Figura 6. DEP Vista de Presentación <i>Común</i>	36
Figura 7. DEN del módulo Cuento	37
Figura 8. Diagrama de Clases de la Vista de Presentación <i>Cuentos</i>	39
Figura 9. Diagrama de Clases de la Vista de Presentación <i>Común</i>	40
Figura 10. Diagrama de Secuencia de la Vista de Presentación <i>Cuento</i>	41
Figura 11. Diagrama de Secuencia de la Vista de Presentación <i>Común</i>	41
Figura 12. Diagrama de Componentes General	44
Figura 13. Diagrama de Componentes del paquete Clases JS	45
Figura 14. Diagrama de Despliegue.....	46
Figura 15. Método de Caja Blanca	47
Figura 16. Método de Caja Negra	48

ÍNDICE DE TABLAS

Tabla 1 Características principales de los programas educativos.	6
Tabla 2. Descripción textual de la Vista de Presentación <i>Cuentos</i>	30
Tabla 3. Descripción textual de la Vista de Presentación <i>Común</i>	34



INTRODUCCIÓN

Los cambios acelerados y el desarrollo en el ámbito de las tecnologías de la información y comunicación (TICs) en la actualidad, de una u otra forma han ocasionado fuertes reestructuraciones en los espacios de trabajo de los diversos campos de la actividad humana.

Cuba ha defendido siempre el concepto de que el uso masivo de las TICs no es un fin sino una herramienta poderosa para lograr el desarrollo, por esta razón hoy en día la informatización forma parte de la lucha por elevar la calidad de vida del pueblo, y lograr una sociedad cada vez más preparada y con acceso pleno al conocimiento (1).

La “sociedad de la información” en general y las nuevas tecnologías en particular inciden de manera significativa en la educación (2), por esto a partir del año 2001, en el sistema educativo cubano se comenzó a introducir la computación a todos los niveles de educación, logrando de esta manera hacer uso de nuevos medios de enseñanza que vinculen los contenidos que transmiten los mismos con la formación informática del escolar.

En la actualidad la utilización de multimedia como método de enseñanza a nivel mundial ha demostrado ser de gran ayuda como material de apoyo para el desarrollo de habilidades en los estudiantes, favoreciendo los procesos de enseñanza y aprendizaje grupales e individuales (3). Por sus características interactivas y versátiles sus principales aportaciones en este sentido son las siguientes: proporcionar información haciendo uso de imágenes, sonido y videos e intensificar el interés de los estudiantes.

El bloqueo económico impuesto a Cuba limita el acceso a las herramientas informáticas para el desarrollo de multimedia, a pesar de esto nuestro país continuó perfeccionando la introducción de este medio de enseñanza en todos los centros educacionales con el objetivo de enriquecer de forma didáctica el aprendizaje de los estudiantes. Para ello se llevó a cabo una serie de acciones que permitieron realizar aplicaciones de este tipo utilizando software libre y sin la necesidad de pagar licencias, pero que no siempre fueron consideradas soluciones óptimas.

El sistema de enseñanza cubana cuenta ya con tres colecciones: “Multisaber”, dedicado a la enseñanza primaria, “El Navegante”, para la enseñanza media y “Futuro”, diseñado para el bachillerato, la enseñanza

técnica-profesional y la educación de adultos. El nivel preescolar al contrario, no cuenta con la cantidad de productos multimedia suficientes para el desarrollo de habilidades de los niños de 3 a 5 años de edad (4). Debido a esto surge la idea de realizar un producto interactivo que contenga cuentos infantiles y que contribuya como material de apoyo para el desarrollo de habilidades de los niños de temprana edad.

El desarrollo de productos multimedia hoy día está liderado por un conjunto de herramientas propietarias tales como Adobe Flash, Flex y Macromedia Director, que facilitan el desarrollo de este tipo de aplicaciones, pero nuestro país no tiene permisos para el acceso a dichas herramientas.

Frente a esta realidad, esta clara la necesidad de alternativas para la construcción de multimedia con tecnologías libres y sin embargo es también la principal problemática. En la actualidad no se cuenta con sistemas de autor que se ejecuten sobre GNU/Linux para el desarrollo de aplicaciones multimedia y las herramientas que existen no cumplen en su totalidad con los requerimientos que implican estos productos (5). Por otra parte la web brinda grandes ventajas en este sentido, pues existen estándares y herramientas libres que facilitan el desarrollo de aplicaciones multimedias debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales (6).

Como alternativa a esta situación el proyecto “Producción de Recursos Didácticos” de la Universidad de las Ciencias Informáticas (UCI) asume el reto de desarrollar el producto “Mis Mejores Cuentos”, en una plataforma web con la utilización de herramientas libres, que contenga cuentos tradicionales para niños de la edad de 3 a 5 años, como “La Caperucita Roja”, “El burro ostentoso”, “El Rey Midas”, etc.

En la multimedia se presentarán los cuentos infantiles, los cuales deben ser de dominio público y deben mostrarse organizados por sus autores, de ahí la necesidad de desarrollar un módulo que contenga la información que se desea presentar.

Por lo anteriormente expuesto el **problema** a resolver queda resumido en la siguiente interrogante: ¿Cómo presentar de forma interactiva los cuentos infantiles en una plataforma web con herramientas libres, para los niños de la edad de 3 a 5 años?

El **objeto de estudio** de este trabajo lo constituye el proceso de desarrollo de software educativo mientras que el **campo de acción** está enfocado en el análisis, diseño e implementación del módulo Cuento del producto “Mis Mejores Cuentos”.

Con este trabajo se **defiende la idea** de que el desarrollo del módulo Cuento del producto “Mis Mejores Cuentos” garantizará la presentación de forma interactiva de los cuentos infantiles más gustados por los niños de la edad de 3 a 5 años.

Objetivo General: Realizar el análisis, diseño, e implementación del módulo Cuento del producto “Mis Mejores Cuentos”.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

1. Investigar el estado del arte.
2. Desarrollar el análisis y diseño del módulo Cuento.
3. Implementar el módulo Cuento del producto “Mis Mejores Cuentos”.
4. Validar la propuesta presentada.

Como **posible resultado** se espera obtener el módulo Cuento del producto “Mis Mejores Cuentos”.

Entre las **tareas a desarrollar** para darle cumplimiento a los objetivos planteados se tienen las siguientes:

- ✓ Elaboración del diseño teórico de la investigación.
- ✓ Selección de la metodología, lenguaje de modelado y herramientas más convenientes para el desarrollo de la propuesta de solución.
- ✓ Identificación de las funcionalidades requeridas para desarrollar el módulo del producto “Mis Mejores Cuentos” detallando los requisitos funcionales y no funcionales detectados mediante la ingeniería de requerimientos.
- ✓ Diseño de los submódulos Presentación y Narración del módulo Cuento del producto “Mis Mejores Cuentos”.
- ✓ Diseño de los submódulos Texto y Autor del módulo Cuento del producto “Mis Mejores Cuentos”.
- ✓ Implementación de los submódulos Presentación y Narración.
- ✓ Implementación de los submódulos Texto y Autor.

- ✓ Integración del módulo Cuento a partir de los submódulos desarrollados.
- ✓ Realización de pruebas de Aceptación utilizando el método de Caja Negra.
- ✓ Conformación del documento de tesis.

En el cumplimiento de las tareas se utilizan los siguientes **métodos científicos**:

- ✓ El método teórico histórico-lógico para llevar a cabo una mejor comprensión del estado actual de los productos multimedia desarrollados con herramientas libres, se investigarán las técnicas y procedimientos adecuados para llevar a cabo una ingeniería de requisitos, además del profundo estudio a las metodologías que existen y técnicas de recopilación de información para la captura de requisitos.
- ✓ El método teórico de la modelación mediante la utilización del lenguaje de modelado ApEM-L, permitirá reflejar la estructura, relaciones internas y características de la solución a través de diagramas a la hora de modelar el negocio y especificar los artefactos a generar, se especificarán además los requisitos funcionales y no funcionales del módulo. Este método facilitará también el diseño de clases, así como otros artefactos necesarios para la implementación y validación del módulo Cuento.

Para una mejor comprensión del documento del trabajo de diploma, el contenido ha sido estructurado en 3 capítulos:

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA Se tratan varios elementos teóricos que soportan la investigación. Además se presenta la arquitectura de software establecida para el proyecto “Producción de Recursos Didácticos”, la cual define las tecnologías y las herramientas informáticas a utilizar.

CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL MÓDULO Se realiza el flujo actual de los procesos a través del modelo del dominio, el cual propicia el análisis necesario para definir las funcionalidades del módulo a desarrollar. Estas características se detallan mediante la especificación de requerimientos, descripción de casos de uso y representación gráfica. Se definen paquetes, dentro de los cuales se agrupan los casos de uso de acuerdo a sus funcionalidades y las clases del análisis. Se describen los patrones utilizados en el diseño del módulo, se define la estructura y los elementos del diseño, se muestran los diagramas de clases de diseño por cada caso de uso.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA En este capítulo se muestra cómo los elementos del modelo de diseño se implementan en términos de componentes, para esto se presenta el diagrama de componentes y el diagrama de despliegue detallado. Se describen también las pruebas realizadas a la solución implementada, la validación de la misma, con el objetivo de asegurar la calidad y eficiencia del sistema.



CAPÍTULO I FUNDAMENTACIÓN DEL TEMA

1.1 Introducción

A lo largo de la última década y como consecuencia del avance tecnológico los medios de enseñanza han evolucionado facilitando la inclusión de la informática en las aulas, la cual ha favorecido el rápido aprendizaje de los estudiantes. La necesidad de introducir la tecnología en los procesos docentes se ha hecho evidente por los logros alcanzados en la rápida fijación del conocimiento por parte de los estudiantes utilizando estas tecnologías, dentro de los cuales desempeña un papel fundamental el software educativo (7).

En el desarrollo del software educativo a nivel mundial ha influido en una tecnología capaz de presentar de forma interactiva la información y de atraer al usuario logrando así una mayor motivación y retención del conocimiento: la tecnología multimedia (4).

En el presente capítulo se introduce al lector en los términos asociados a los conceptos de software educativo y tecnología multimedia. Además, se describen algunos de los productos que existen a nivel nacional e internacional desarrollados bajo esta tecnología. Se presentarán también las características principales de las metodologías, tecnologías y herramientas seleccionadas que permitirán desarrollar la solución correcta al problema de este trabajo investigativo.

1.2 Software educativo

Existen varias definiciones de software educativo o programa educativo (8); (9); (10); (11); entre otros). En este trabajo investigativo se considera la formulada por Cataldi (8) como la más completa, la cual define al software educativo como “los programas de computación realizados con la finalidad de ser utilizados como facilitadores del proceso de enseñanza” y consecuentemente de aprendizaje, con algunas características particulares tales como: la facilidad de uso, la interactividad y la posibilidad de personalización de la velocidad de los aprendizajes.

Marques citado en Cataldi (8) sostiene que se pueden usar como sinónimos de software educativo los términos “programas didácticos” y “programas educativos”, centrandó su definición en “aquellos

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

programas que fueron creados con fines didácticos, en la cual excluye todo software del ámbito empresarial que se pueda aplicar a la educación aunque tengan una finalidad didáctica, pero que no fueron realizados específicamente para ello”. Las características principales de los programas educativos, según Marques, se muestran en la siguiente tabla.

Características	Descripción
Facilidad de uso.	En lo posible autoexplicativos y con sistemas de ayuda.
Capacidad de motivación.	Mantener el interés de los alumnos.
Relevancia curricular.	Relacionados con las necesidades del docente.
Versatilidad.	Adaptables al recurso informático disponible.
Enfoque pedagógico.	Que sea actual: constructivista o cognitivista.
Orientación hacia los alumnos.	Con control del contenido del aprendizaje.
Evaluación.	Incluirán módulos de evaluación y seguimiento.

Tabla 1 Características principales de los programas educativos.

1.3 Tecnología multimedia

Multimedia es uno de los términos relacionados con la informática y de uso más frecuente en los últimos años, por lo que en muchas ocasiones es considerada una tecnología totalmente nueva con la que todos quieren interactuar. Una de las principales definiciones que se utiliza en la actualidad precisa que multimedia es la integración de dos o más medios de comunicación que pueden ser controlados o manipulados por el usuario en una computadora, o sea, es un sistema informático interactivo, que integra diferentes medias digitales como texto, vídeo, imagen, sonido y animaciones (4).

Internacionalmente la utilización de multimedia como fuente de conocimiento para el aprendizaje ha alcanzado grandes resultados aunque partiendo de herramientas propietarias. Ejemplo de esto se puede mencionar a: “Boaki”, multimedia interactiva destinada a niños de 6 y 12 años de edad; “Pelayo y su

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

Pandilla”, recurso interactivo que introduce la unidad didáctica por medio de un cuento con ilustraciones, dirigido a niños de 4 y 5 años de edad; entre otras.

En la actualidad el desarrollo de herramientas libres para la construcción de aplicaciones multimedia no es lo suficientemente sólido como para saciar las necesidades que estos productos implican. La comunidad de software libre ha empleado todo su empeño en desarrollar dichas herramientas, aunque todavía no se cuenta con sistemas a la altura de Macromedia. Entre los competidores de Macromedia están en sus versiones libres OpenLaszlo, BlueFlish y Synfig (5).

Por muchos años la herramienta propietaria de animación Flash fue el dominante en la web para la presentación de elementos multimedia; hoy en día la conjunción de HTML5 + CSS3 + Canvas + Javascript ha logrado por fin encontrar un reemplazo casi perfecto para Flash. En los últimos meses se han realizado animaciones 2D y 3D espectaculares, comparables con productos desarrollados con Flash (com.per.1).¹

En Cuba existe una Red de Centros de Estudio de Software Educativo a lo largo de todo el país, encargada de desarrollar aplicaciones educativas que contribuyan a elevar el nivel del aprendizaje, entre las que se destacan: la “Colección Multisaber”, la “Colección El Navegante”, la “Colección Futuro”, entre otras.

Actualmente en la Universidad de las Ciencias Informáticas en colaboración con el Ministerio de Educación (MINED) se está implementando un proceso de migración a versiones multiplataforma de las colecciones Multisaber, actualmente “La Caja Mágica”, y El Navegante propiciado por un acuerdo que persigue la utilización de estas colecciones en la República Bolivariana de Venezuela. También se está desarrollando Bachelor, versión multiplataforma de la colección Futuro, para la comercialización con México. Estas versiones se desarrollan con herramientas y tecnologías libres como son los lenguajes Javascript y PHP (Hypertext Preprocessor).

La Colección Multisaber tiene un enfoque multidisciplinario por su relación con los contenidos de los programas de todas las asignaturas del currículo de estudio de este nivel de enseñanza. Esta colección agrupa varias aplicaciones que tributan a la formación de una cultura general integral de los estudiantes

¹ Gregorio Espadas. Licenciado en Ciencias de la Computación de la Universidad Autónoma de Yucatán. México. gespadas@gmail.com.

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

así como una interfaz estandarizada que proporciona un ambiente de trabajo amigable e intuitivo con un alto nivel de interactividad para acceder a la información existente en el software (4).

El Navegante es una colección compuesta por 10 aplicaciones educativas que integran todos los contenidos del nivel secundario presentando un carácter curricular extensivo, esto significa que constituye un soporte informático pleno para el proceso docente para cada una de las asignaturas y grados hacia los cuales va dirigido el programa (4).

También la UCI desarrolló el software educativo “A Jugar” para niños de 1 a 5 años de edad, que presenta una interfaz amigable y permite el desarrollo de habilidades en la computadora pero siempre a través de un intermediario, pues presenta una compleja navegación para un niño de esta edad (4). Es válido destacar que este producto está sustentado sobre tecnología multimedia, aunque desarrollado con herramientas propietarias por lo que su comercialización con otros países se hace costosa.

Con el desarrollo del módulo Cuentos del producto “Mis Mejores Cuentos” se pretende diseñar una interfaz amigable para los niños en las edades de 3 a 5 años, donde puedan interactuar con cada una de las funcionalidades sin la necesidad de ayuda por parte de terceras personas, desarrollando así habilidades con la computadora. Este módulo será desarrollado rompiendo los esquemas del uso de herramientas propietarias para la implementación de productos multimedia.

1.4 Metodologías de desarrollo de software

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar y documentar un software (12). De acuerdo con Figueroa (12) el papel preponderante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo. En la actualidad existen dos grandes tendencias de metodologías en las cuales los desarrolladores se pueden basar para realizar su trabajo: metodologías tradicionales y metodologías ágiles.

Las metodologías ágiles se caracterizan por tener un desarrollo incremental para producir tempranamente pequeñas entregas en ciclos rápidos y predisposición para el cambio y la adaptación continua. Estas

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

metodologías por lo general se centran en desarrollar productos funcionales, más que en conseguir una buena documentación (13).

Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Lo proponen porque para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significa reducir su coste (12). Entre las principales metodologías ágiles se encuentran eXtreme Programming (XP), Scrum, Iconix, Crystal Methods, Agil Unified Process (AUP), entre otras.

Las metodologías tradicionales o pesadas se centran en el control del proceso y establecen una rigurosa planificación de tareas y responsabilidades, generando gran cantidad de documentos y artefactos. Estas se distinguen por especificar las herramientas y notaciones que deben ser utilizadas durante la construcción del producto (14). Estas metodologías imponen una disciplina de trabajo sobre el proceso de desarrollo del software con el objetivo de conseguir un software más eficiente y predecible. Para ello, se hace especial énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto de software (13). Entre las metodologías tradicionales o pesadas se encuentra RUP (Rational Unified Process), MSF (Microsoft Solution Framework), entre otras.

En este trabajo investigativo se analiza la metodología tradicional RUP porque es la que define el diseño arquitectónico realizado para el producto “Mis Mejores Cuentos”, además su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto) (15).

1.4.1 Rational Unified Process (RUP)

RUP provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. RUP es el resultado de la integración de diferentes metodologías de desarrollo, donde se unifican los mejores elementos de cada una de las anteriores. De acuerdo con

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

Jacobson y otros autores (15), RUP presenta un enfoque orientado a objetos, contando con las mejores prácticas del modelo de desarrollo de un software, como son:

- ✓ Desarrollo de software iterativo e incremental.
- ✓ Administración de requerimientos.
- ✓ Arquitectura basada en componentes.
- ✓ Modelar visualmente.
- ✓ Verificar la calidad del producto.
- ✓ Controlar los cambios durante el ciclo de vida del software.

Las cuatro fases del ciclo de vida según la página oficial de RUP (16) son:

- ✓ **Inicio:** se define la idea, la visión del producto, como se enmarca en el negocio y el alcance del proyecto.
- ✓ **Elaboración:** se planifican las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura.
- ✓ **Construcción:** se construye el producto, la arquitectura y los planes, hasta que el producto esté listo para ser enviado a la comunidad de usuarios.
- ✓ **Transición:** se realiza la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto, hasta que el cliente esté satisfecho.

El proceso propuesto por RUP posee dos dimensiones: la primera, representa el aspecto dinámico del proceso, y está expresado en términos de ciclos, fases, iteraciones e hitos, la segunda, representa el aspecto estático, que se describe en términos de componentes, actividades, flujos de trabajo, artefactos y actores (17) como se muestra en la figura 1.

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

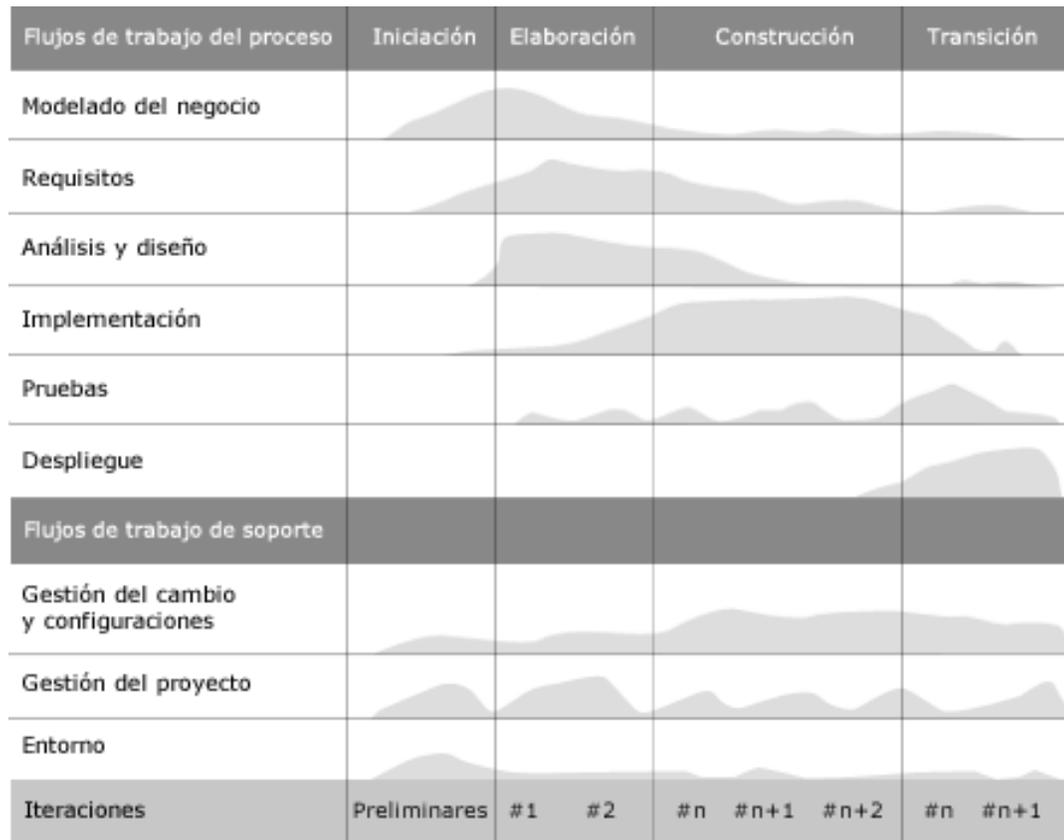


Figura 1. RUP en dos dimensiones

En resumen, las metas de RUP según Catalá (18) son:

- ✓ Asegurar la producción de un software de alta calidad que reúna las necesidades de los usuarios finales dentro de un plan y un presupuesto predecible.
- ✓ Proveer un enfoque disciplinado para asignar tareas y responsabilidades dentro del desarrollo del sistema.
- ✓ Proveer un camino metódico, sistemático para desarrollar, diseñar y validar una arquitectura.

Durante el ciclo de vida se genera gran documentación permitiendo que si varía el equipo de desarrollo se puede comprender el proceso. Es utilizada en proyectos de larga duración y se puede ajustar a cualquier tamaño de equipo de trabajo.

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

RUP resulta ser la metodología más indicada para guiar el proceso actual, debido a las múltiples ventajas que ofrece. Como metodología robusta se ajusta a dicho proceso ya que RUP a pesar de utilizar los Casos de Uso para guiar el proceso, y definir ciclos de trabajo, presta especial atención al establecimiento temprano de una arquitectura fuerte, además genera gran cantidad de documentación que es muy útil para el control, la organización y la detección de errores.

1.5 Lenguajes de modelado

Un lenguaje de modelado es un conjunto estandarizado de símbolos dispuestos para modelar parte de un diseño de software orientado a objetos. Estos se combinan con las metodologías de desarrollo para avanzar de una especificación inicial a un plan de implementación, para comunicar dicho plan a todo un equipo de desarrolladores (19). Los modelos proporcionan un mayor nivel de abstracción, permitiendo trabajar con sistemas mayores y más complejos, y facilitando el proceso de codificación e implementación del sistema de forma distribuida y en distintas plataformas (18).

1.5.1 Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L) 1.5.

ApEM-L es una extensión de UML (Lenguaje Unificado de Modelado) que incorpora a este los elementos fundamentales del proceso productivo UCI. ApEM-L no modifica la semántica de UML, sino que trabaja en estereotipos restrictivos, por lo que a su vez produce modificaciones descriptivas y decorativas en la representación de los componentes del lenguaje base (20). En ApEM-L están incluidos los elementos más significativos de OMMMA-L (Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia), para de esta forma lograr una extensión consistente para la modelación de aplicaciones educativas.

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA



Figura 2. Lenguajes básicos de ApEM-L

ApEM-L tiene entre sus principales objetivos (20) :

- ✓ Desarrollar una extensión del lenguaje de modelado UML, tomándolo como base e incorporando a este, a través de sus mecanismos de extensión, los elementos fundamentales del proceso productivo UCI.
- ✓ Incorporar los elementos más significativos de extensiones anteriores como OMMMA – L (2001) y a su vez respetar lo establecido por el estándar OCL (Lenguaje de Modelación de Objetos) 2003, para de esta forma lograr una extensión consistente y escalable en el tiempo.
- ✓ No complicar ApEM – L con elementos que lo convirtieran o abarcaran un método de desarrollo de aplicaciones educativas, sino sólo el área de la representación y la documentación de este tipo de aplicaciones.
- ✓ No circunscribir ApEM – L a un proceso de desarrollo en específico, sino expresarlo de manera tal que pueda ser utilizado con cualquiera de los existentes, aunque se sugiere la utilización de procesos de desarrollo iterativo, incremental y basado en prototipos, que permitan la modelación de sistemas orientados a objetos.

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

Los conceptos y modelos de ApEM-L están agrupados en las siguientes áreas conceptuales (20):

- ✓ **Estructura lógica:** está compuesta por la Vista Estática y la Vista de Arquitectura. Cualquiera de los modelos presentados por ApEM-L define los conceptos clave de la aplicación que modela, las propiedades internas de estos y sus relaciones. Estos conceptos son modelados como clases, describiendo cada una un conjunto de objetos que almacenan información y se comunican para implementar su comportamiento. La información almacenada se representa como atributos de estas clases y las operaciones a través de los métodos de dichas clases. A su vez la vista de arquitectura la componen el diagrama de componentes y el diagrama de despliegue.
- ✓ **Comportamiento dinámico:** el comportamiento de la aplicación está descrito por la vista de comportamiento, la cual está compuesta por los diagramas: de actividad, de secuencia, de colaboración y de estados, donde solo ha sido modificado el segundo de los listados anteriormente; adicionando una variable de tiempo donde quiera que sea necesario su especificación para un mejor entendimiento.
- ✓ **Gestión del modelo:** esta área es la que ha sufrido grandes cambios tanto en su carácter semántico como sintáctico, con la incorporación de estereotipos restrictivos en todos los diagramas a partir de nuevos conceptos incorporados a los diagramas de clases originales básicos de UML. Se crean dos nuevos diagramas: el de estructura de la presentación y el de estructura de la navegación.
- ✓ **Presentación:** esta área está compuesta por la Vista de Presentación vista de presentación que ha sido incorporada completamente sobre el lenguaje base UML, para permitir utilizar la semántica original de dicho lenguaje en la construcción de estructuras lógicas de presentación y navegación, construyendo nuevos diagramas dentro de la misma: el diagrama de estructura de navegación y el diagrama de estructura de presentación.

ApEM-L está dividido en 4 vistas: Vista Estática, Vista de Arquitectura, Vista de Comportamiento y Vista de Presentación, cada una de estas modela un aspecto del sistema. La división ha sido sobre la base de las áreas conceptuales ya presentadas: estructura lógica, comportamiento dinámico y gestión del modelo. Una descripción detallada de cada una de estas vistas se encuentra en Febe (20).

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

ApEM-L no pretende convertirse en un método de desarrollo de aplicaciones con tecnología multimedia, sino en el área de la representación y la documentación de este tipo de aplicaciones (20).

Se selecciona como el lenguaje de modelado APEM-L 1.5 porque es un lenguaje orientado a la descripción de las vistas del software educativo, además responde a una mejor modelación de aplicaciones educativas ya que en él se incorporan nuevos diagramas y estereotipos que permiten una mejor comprensión del funcionamiento de aplicaciones con estas características.

1.6 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering) son utilizadas con el fin de automatizar los aspectos clave de todo un proceso de desarrollo de software de un sistema (21). Las herramientas CASE permite modelar procesos de desarrollo de software, y tienen como objetivo fundamental solucionar y afrontar los problemas de una mala calidad de software y una documentación inadecuada (22).

1.6.1 Visual Paradigm 6.4

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite graficar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (23).

Una de las características más importantes de Visual Paradigm es que es multiplataforma. Esta herramienta es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario. Su notación es muy parecida a la estándar, permite configurar las líneas de redacción, el modelado de base de datos, el modelado de requerimientos, el modelado del proceso de negocio, la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP además de permitir la integración con herramientas de desarrollo (24).

Se selecciona Visual Paradigm 3.4 como herramienta CASE por ser una herramienta multiplataforma que permite llevar una buena organización y planificación de los diagramas.

1.7 Lenguajes de programación web

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas (25). Se pueden clasificar en lenguajes del lado del servidor como PHP, ASP (Active Server Pages), entre otros y tecnologías del lado del cliente como Javascript, VBScript (Visual Basic Script), etc. En esta investigación se analizan los lenguajes del lado del cliente pues estos permiten manejar los eventos del navegador y ganar una mayor rapidez e interactividad en las respuestas al cliente.

VBScript es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de Javascript (26). Por las facilidades que presenta el uso de Javascript y además por estar definido en la arquitectura diseñada para el producto “Mis Mejores Cuentos” como lenguaje de programación a utilizar, se exponen en la presente investigación las características y ventajas que favorecen el desarrollo del módulo Cuento.

1.7.1 Lenguaje Javascript

Javascript es un lenguaje de scripts basado en objetos que permite hacer aplicaciones incrustadas en un documento HTML, que pueden manejar eventos según acciones de parte del usuario. Es un recurso muy potente para lograr mejorar las páginas web así como lograr un óptimo funcionamiento de sus proyectos web (27).

Según Bonilla (27) las principales ventajas de Javascript son:

- ✓ Es un código “interpretado” por el cliente.
- ✓ Es un lenguaje abierto.
- ✓ Es un código que soporta objetos.
- ✓ Es un código integrado a HTML.
- ✓ Reutilización de código de programación.

Javascript supera a otros lenguajes del lado del cliente en cuanto a portabilidad, rendimiento, facilidad de uso y documentación (28).

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

Para el módulo se decidió usar Javascript debido a la gran cantidad de animaciones que contiene y este lenguaje se caracteriza por brindar una serie de funcionalidades en este sentido. Otra ventaja que brinda el lenguaje es la conexión asincrónica con el servidor web, aumentado con esta la velocidad de respuesta del mismo.

1.8 Sistema de Control de Versiones

Un sistema de control de versiones es un sistema de gestión de archivos y directorios cuya principal característica es que mantiene el registro de los cambios y modificaciones que se han realizado sobre estos a lo largo del tiempo. De esta forma, el sistema es capaz de recordar las versiones antiguas de los datos, lo que permite examinar el historial de cambios o recuperar versiones anteriores de un fichero (29).

1.8.1 Subversion (SVN)

Subversion es un sistema de control de versiones libre y de código fuente abierto. Es decir, Subversion maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos (30).

Esta herramienta permite el acceso a los archivos de un proyecto, observar el trabajo, realizar cambios y guardar estos cambios en su repositorio, donde se almacenan todos los archivos involucrados. Al finalizar una acción de guardar cambios, se considera que se ha creado una nueva "revisión" (31).

De acuerdo con Vindas (31) entre las ventajas que proporciona el uso de Subversión como Sistema de control de Versiones se encuentran:

- ✓ **Visibilidad del cambio:** facilita mantener involucrada a cada persona de su equipo en el proyecto, ya que los cambios son visibles para todos una vez que son realizados.
- ✓ **Versión a versión:** guardará únicamente las diferencias de una versión a otra en el servidor.
- ✓ **Administración:** permite al administrador del proyecto observar la productividad de su equipo de trabajo, así como visualizar el historial de cambios de un producto, permitiendo una mejor

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

administración. Esto también le da la opción de trabajar en múltiples secciones de forma paralela, dentro del mismo proyecto, mediante las ramificaciones que maneja.

De acuerdo con las políticas de configuración establecidas por el proyecto “Producción de Recursos Didácticos” se decide utilizar como sistema de control de versiones SVN.

1.9 Servidor web

Un servidor web es un programa que implementa el protocolo Hypertext Transfer Protocol (HTTP, por sus siglas en inglés). Este protocolo pertenece a la capa de aplicación del modelo Open System Interconnection (OSI) y está diseñado para transferir hipertextos, páginas web o páginas HyperText Markup Language (HTML), textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. De acuerdo con una encuesta realizada por la empresa inglesa Netcraft, en septiembre de 2009, los servidores web más usados en la actualidad son Internet Information Services (IIS) y Apache (32).

1.9.1 Servidor Apache 2.2

Un servidor de aplicaciones web es el encargado de recibir las peticiones emitidas por los navegadores, atienden únicamente los aspectos del cliente HTML, Javascript y HTTP. Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1. Es fiable, eficiente y fácilmente extensible.

Según Marques Díaz (33) las ventajas que brinda Apache son:

- ✓ Es una tecnología gratuita, de código abierto.
- ✓ Es un servidor altamente configurable, de diseño modular.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.
- ✓ Es robusto y seguro.

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

De acuerdo a la arquitectura de software definida para el desarrollo del producto “Mis Mejores Cuentos” se decide utilizar como servidor web el servidor Apache 2.2.

1.10 Entorno de Desarrollo Integrado (IDE)

Un IDE es un conjunto de herramientas utilizadas por los programadores, que incluye por lo general, un editor de código, administrador de proyectos y archivos, enlace a compiladores e integración con sistemas controladores de versiones o repositorios, además de brindar facilidades para la construcción de interfaces gráficas de usuario (34).

Las características principales de un IDE de acuerdo con Bradshaw y otros autores (34) son las siguientes:

- ✓ Permite trabajar con varios proyectos a la vez, encontrar código duplicado y personalizar el entorno.
- ✓ Incorpora completamiento de código.
- ✓ Permite la integración con aplicaciones de control de versiones.
- ✓ Permite comparar archivos.
- ✓ Mantiene un historial local de los archivos.
- ✓ Incorpora funcionalidades a través de extensiones.
- ✓ Permite el trabajo con múltiples lenguajes de programación.

1.10.1 Aptana Studio 2.0.5

Aptana Studio es un IDE que permite mediante una moderna y agradable interfaz, trabajar con diferentes lenguajes y tecnologías de programación web como HTML, DOM (Document Object Model), Javascript y CSS (Cascading Style Sheets) (35).

Además, este IDE permite la diferenciación de código, chequeo de sintaxis, organización estructural del código fuente, depuración, autocompletitud integrada con el DOM (Document Object Model) e integra diferentes librerías (Extjs, jQuery, Scriptaculous, Dojo etc.) entre una lista bastante grande de opciones, herramientas y perspectivas (36).

CAPÍTULO I

FUNDAMENTACIÓN DEL TEMA

Aptana Studio es gratuito, de código abierto y multiplataforma, por lo que puede instalarse en cualquier estación de trabajo con sistemas operativos Windows, Linux o Mac OSX en sus versiones más recientes, además permite comprobar la compatibilidad de las funciones con los diferentes navegadores (35).

Se decide utilizar Aptana Studio 2.0.5 como entorno de desarrollo integrado para la implementación del módulo Cuento pues basado en las características mencionadas lo define la arquitectura diseñada para el desarrollo del producto “Mis Mejores Cuentos”.

1.11 Conclusiones

Las características propias de la metodología RUP hacen que esta sea una guía completa y apropiada para desarrollar el módulo.

El estudio realizado sobre los lenguajes de modelado determinó que ApEM-L 1.5 es el lenguaje más adecuado para modelar aplicaciones educativas.

La selección del IDE Aptana Studio 2.0.5 en conjunto con el lenguaje de programación Javascript facilitará en gran medida el desarrollo del módulo en la fase de implementación.



CAPÍTULO II ANÁLISIS Y DISEÑO DEL MÓDULO

2.1 Introducción

El resultado final de un proyecto de software es un producto que toma forma a lo largo del desarrollo del proyecto. La calidad del producto final, está estrechamente ligada a la calidad del proceso de desarrollo de software, entre otros aspectos, donde se requieren de etapas de modelamiento que permitan experimentar y visualizar el sistema que se construirá (37).

Este capítulo se centra en la descripción de la solución propuesta para resolver el problema planteado en la investigación, teniendo como guía de desarrollo la metodología RUP. Se identifican los conceptos asociados al dominio y los requisitos funcionales y no funcionales del sistema. También se modela el dominio a partir de los conceptos identificados, la Vista de Gestión del Modelo, los Diagramas de Estructura de Navegación (DEN), los Diagramas de Estructura de Presentación (DEP) y se describen las vistas de presentación identificadas en el producto.

2.2 Modelo de Dominio

Según Rosales (38), para modelar un sistema con tecnología multimedia no es necesario definir los procesos de negocio sino los conceptos relacionados con el mismo, se propone la elaboración de un modelo de dominio que represente los tipos de objetos más importantes en el sistema. El modelo de dominio representa los conceptos (objetos) más importantes en el contexto del sistema, donde los objetos son entidades que existen en el mundo real y no de los componentes de software, además tienen identidad y son distinguibles entre sí (39). El modelo de dominio contiene clases de objetos, las que a su vez incluyen asociaciones entre sí y cada una posee atributos pero no métodos, ya que en el modelo de dominio no se muestra comportamiento. Al no encontrarse definidos claramente los procesos para el módulo Cuento del producto “Mis Mejores Cuentos” se propone el modelo de dominio (Ver figura 3). El modelo de dominio permite una mejor comprensión de los conceptos del entorno donde se desarrolla el problema planteado.

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

2.2.1 Identificación de los conceptos del dominio

El modelo de dominio sirve para la comprensión y descripción de las clases más significativas del módulo. Con la aplicación de este modelo se identificaron las siguientes entidades y conceptos fundamentales:

Niño: Usuario que interactúa con la aplicación.

Cuento: Narración breve de hechos imaginarios o reales, protagonizada por un grupo reducido de personajes y con un argumento sencillo.

Narración: Se relatan los cuentos presentes en la aplicación mediante sonidos y animaciones.

Texto: Texto donde se muestra el contenido del cuento seleccionado por el usuario.

Autor: Se especifican los datos más importantes del autor del cuento seleccionado por el usuario.

Entretenimiento: Distintas opciones de juego que el sistema le brinda al usuario.

Idioma: Distintas opciones que el sistema le brinda al usuario para que el cuento se muestre en el idioma seleccionado.

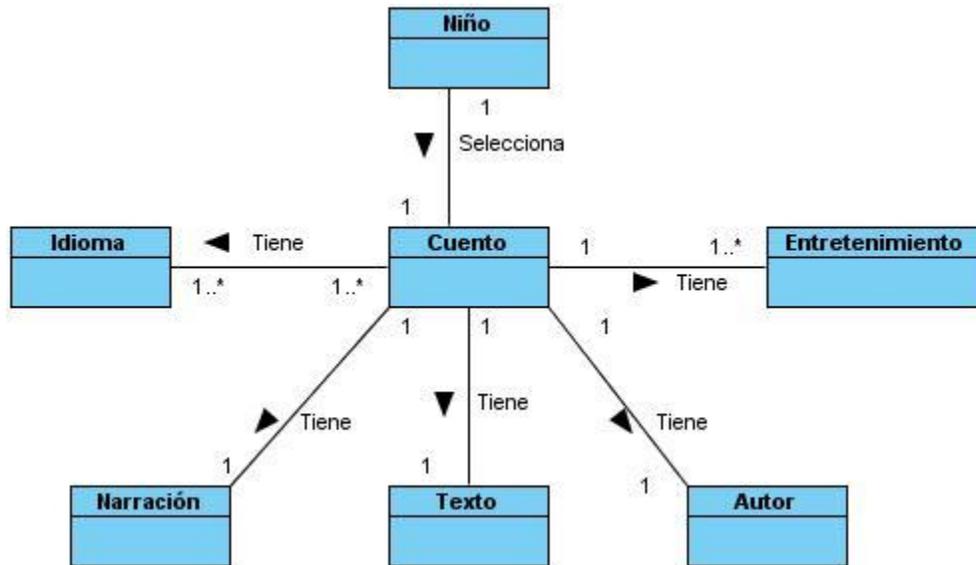


Figura 3. Modelo Conceptual o Modelo de Dominio

2.3 Requerimientos del Sistema

Los requerimientos de un sistema definen qué es lo que este debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen (40). Un requerimiento es una característica de diseño, una propiedad o un comportamiento de un sistema. Estos constituyen la descripción de los deseos o de las necesidades de un producto. De manera general se clasifican en requerimientos funcionales y no funcionales.

Requerimientos funcionales: son capacidades o condiciones que el sistema debe cumplir.

Requerimientos no funcionales: son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

2.3.1 Requerimientos Funcionales

RF1 Mostrar presentación.

RF1.2 Cargar imágenes de presentación.

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

RF1.3 Reproducir presentación.

RF2 Mostrar Cuento Infantil.

RF2.1 Mostrar narración de un cuento infantil.

RF2.2 Mostrar texto de un cuento infantil.

RF3 Mostrar los Entretenimientos asociados a un cuento infantil.

RF4 Mostrar datos de la biografía del autor de un cuento infantil.

RF5 Permitir al usuario volver a la pantalla principal del producto.

RF6 Permitir cambiar de idioma el texto y la narración de un cuento infantil.

RF7 Mostrar despedida de la multimedia y salir de la aplicación.

2.3.2 Requerimientos No Funcionales

Lograr cumplir las expectativas del usuario, implica, además de lograr que el software funcione, que lo haga de la mejor manera posible; en base a los requisitos no funcionales que especifican a continuación.

RNF 1 Requerimiento de software

RNF 1.2 Debe ser multiplataforma

RNF 1.3 El módulo debe estar disponible a través del servidor web Apache 2.2

RNF 1.4 El módulo debe visualizarse de igual forma en todos los navegadores.

RNF 2 Requerimiento de Hardware

RNF 2.1 Memoria RAM de 128 Mb (Mínimo).

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

RNF 2.2 Espacio disponible en disco 5gb (Mínimo)

RNF 3 Requerimiento de Usabilidad

RNF 3.1 El módulo deberá permitir el acceso a cualquier usuario. Los usuarios podrán acceder libremente al módulo.

RNF 3.2 La información contenida en el módulo debe estar libre de derecho de autor.

RNF 4 Requerimiento de Soporte

RNF 4.1 El módulo será probado, instalado y configurado por los especialistas, que también se ocuparán de su mantenimiento.

RNF 5 Requerimiento de apariencia o interfaz externa

RNF 5.1 El diseño de la interfaz del módulo deberá ser sencillo, no estará sobrecargado de contenido, pero debe tener elementos visuales que capten la atención y que de forma metafórica representen una idea visual, para lograr un mayor entendimiento a los usuarios finales a los que va dirigido este producto: niños entre 3 y 5 años.

RNF 6 Restricciones en el diseño y la implementación

RNF 6.1 El módulo debe ser implementado en lenguaje de programación Javascript.

RNF 6.2 Se utilizarán como herramienta de modelado Visual Paradigm 6.4, y como IDE de desarrollo Aptana 2.0.5.

RNF 6.3 Las imágenes tendrán formato: .jpeg o .gif.

2.4 Vista de Gestión del Modelo y sus subsistemas

El principio de organización que propone ApEM-L para la construcción de un sistema se basa en la división de la aplicación por subsistemas. Cada subsistema va a estar compuesto por una o más Vistas de Presentación. Las relaciones entre los subsistemas van a ser representados mediante la Vista de Gestión del Modelo (VGM) (20). Para la construcción del módulo Cuento del producto “Mis Mejores Cuentos” se han identificado cuatro subsistemas, definidos por las características y funcionalidades de las vistas que lo conforman.

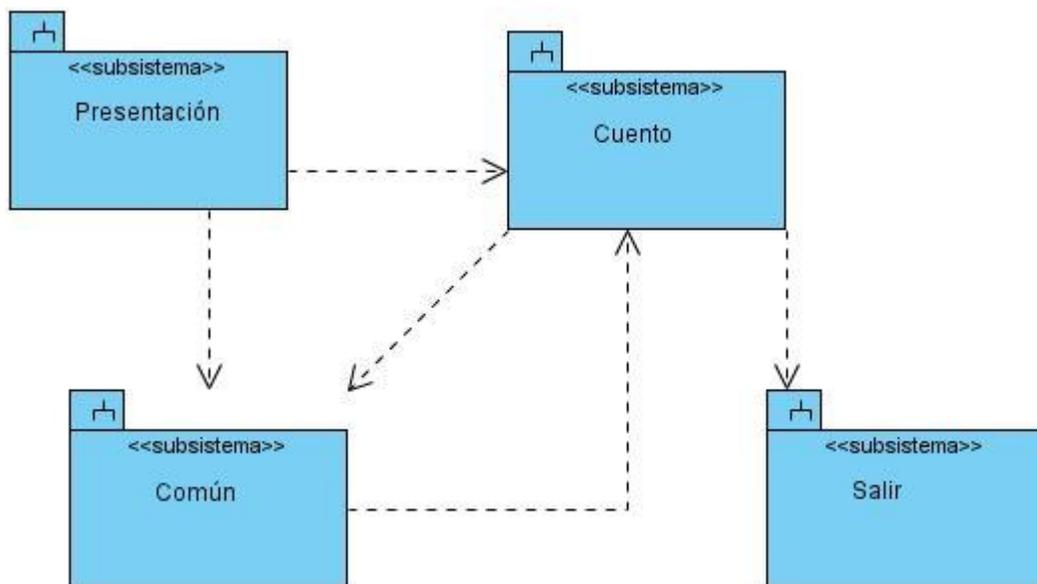


Figura 4. Representación de la Vista de Gestión del Modelo

A continuación se exponen cada uno de los subsistemas y las vistas que los componen, así como una breve explicación de la funcionalidad de estos.

Subsistema Presentación: Es el módulo encargado de mostrar la presentación de la aplicación.

Subsistema Cuento: Es el módulo encargado de manejar los elementos más importantes de los cuentos, está compuesto por las siguientes vistas: Narración, Cuento, Texto y Autor.

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

Común: Representa un conjunto de opciones comunes que permiten modelar de una manera más entendible el módulo y son de gran importancia a la hora de navegar por el mismo, está compuesto por la vista Común y la vista Idioma.

Salir: Es el encargado de mostrar los elementos de despedida cuando el usuario desea abandonar la aplicación.

2.5 Vistas de Presentación

La Vista de Presentación es el artefacto que recoge la descripción de los elementos y su relación para el cumplimiento de una determinada función del sistema. Ha sido incorporada completamente al lenguaje base de UML, para permitir utilizar la semántica original de dicho lenguaje en la construcción de estructuras lógicas de presentación y navegación (20).

2.5.1 Descripción textual de las Vistas de Presentación

La Descripción Textual de las Vistas de Presentación surge para una mejor descripción del contexto productivo del software educativo, bien explícito, incluyendo objetivos pedagógicos, elementos asociados a las vistas y medias a utilizar. Su objetivo principal es dejar claro en detalle todo el flujo de evento así como todas las características esenciales y necesarias de la vista de presentación (38).

Descripción textual de la vista de Presentación <i>Cuentos</i>	
Actores de la vista de presentación	Usuario
Propósito	Mostrar al usuario los diferentes cuentos infantiles que se encuentran en el sistema.
Objetivos pedagógicos	Permitir conocer un cuento infantil.
Resumen	La vista se inicia cuando se termina de ejecutar la vista de presentación, se muestra los íconos de los cuentos, permitiendo seleccionar uno de

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

	ellos. Seguidamente el actor selecciona el cuento de su interés y el sistema muestra la Vista Común, terminando así la ejecución de esta vista.	
Vistas asociadas	Presentación, Común.	
Referencias	RF1	
Precondiciones	Debe haberse ejecutado la Vista de Presentación <i>Presentación</i> .	
Poscondiciones	Se ejecuta la vista Cuentos.	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	Elementos de la Vista
1. La vista se inicia cuando se termina de ejecutar la vista de presentación.	2. Se muestran los íconos (botones) de todos los cuentos.	2.1. Los íconos se ubicarán en toda la pantalla.
-	3. Se muestra el ícono Salir.	3.1. El ícono se mostrará en la parte inferior derecha de la pantalla.
-	4. Se reproduce un sonido indicando las opciones que se brindan en la vista.	4.1. El sonido indicando las opciones que se brindan en la vista se reproducirá durante 10 s.
5. Selecciona el cuento de su interés.	6. Muestra la vista Común, terminando así la vista.	-
Curso Alternativo de los eventos		
Acción	Curso Alternativo	Elementos de la Vista
4. Se reproduce un sonido indicando las opciones que se brindan en la vista.	4. a.1 El actor selecciona Salir, terminando así la ejecución de la vista.	4. a.1.1 Se ilumina el botón Salir y se reproduce el sonido

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

		explicando la acción del botón.	
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I-C-001	Imagen cuento1.	Existente.
	I-C-002	Imagen ícono cuento2.	Existente.
	I-C-003	Imagen ícono cuento3.	Existente.
	I-C-004	Imagen ícono cuento4.	Existente.
	I-C-005	Imagen ícono cuento5.	Existente.
	I-C-006	Imagen ícono cuento6.	Existente.
	I-C-007	Imagen ícono cuento7.	Existente.
	I-C-008	Imagen ícono cuento8.	Existente.
	I-C-009	Imagen ícono cuento9.	Existente.
	I-C-010	Imagen ícono cuento10.	Existente.
	I-C-011	Imagen ícono cuento11.	Existente.
	I-C-012	Imagen ícono cuento12.	Existente.
	I-BS-001	Imagen de botón Salir.	Existente.
Video o animación	-	-	-
Sonido	S-BS-001	Sonido botón Salir.	Existente.
	S-C-001	Sonido ícono cuentos1	Existente.
	S-C-002	Sonido ícono cuentos2	Existente.
	S-C-003	Sonido ícono cuentos3	Existente.
	S-C-004	Sonido ícono cuentos4	Existente.
	S-C-005	Sonido ícono cuentos5	Existente.
	S-C-006	Sonido ícono cuentos6	Existente.
	S-C-007	Sonido ícono cuentos7	Existente.

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

	S-C-008	Sonido ícono cuentos8	Existente.
	S-C-009	Sonido ícono cuentos9	Existente.
	S-C-010	Sonido ícono cuentos10	Existente.
	S-C-011	Sonido ícono cuentos11	Existente.
	S-C-012	Sonido ícono cuentos12	Existente.
	S-V-C	Sonido de la vista Cuentos	Existente.
Texto	-	-	-
Reglas Pedagógicas	-		

Prototipo de Interfaz de Usuario de la Vista

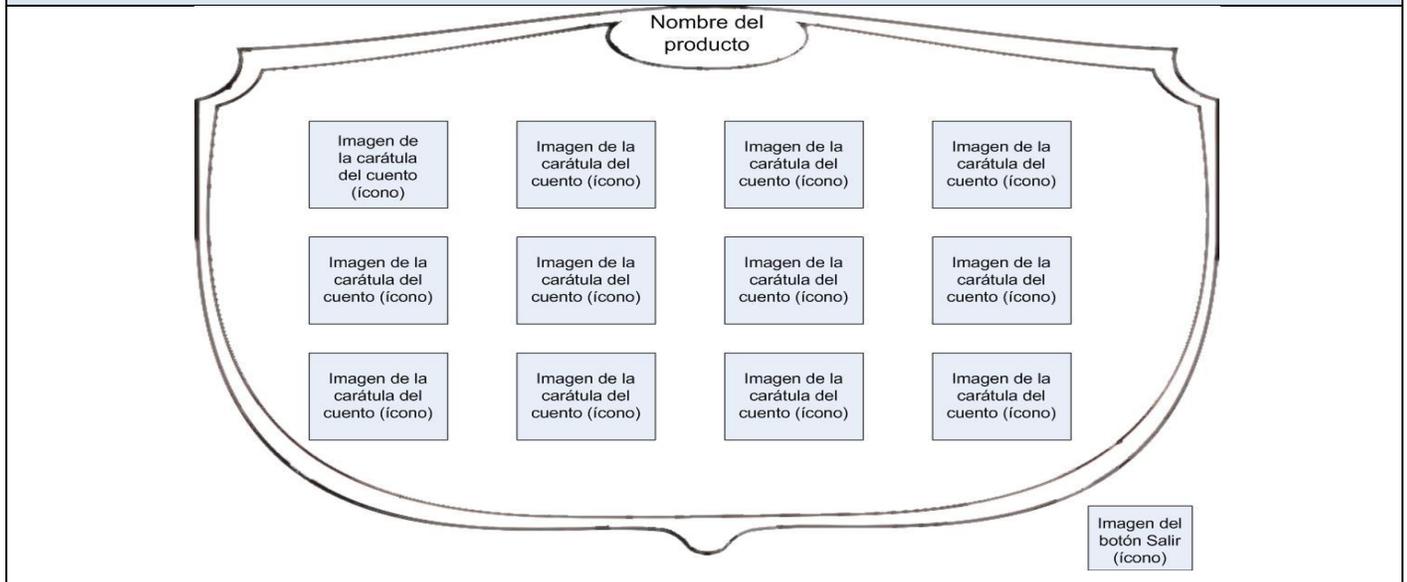


Tabla 2. Descripción textual de la Vista de Presentación *Cuentos*

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

Descripción textual de la vista de Presentación <i>Común</i>		
Actores de la vista de presentación	Usuario	
Propósito	Presentar al usuario diferentes opciones de trabajo.	
Objetivos pedagógicos	-	
Resumen	La vista se inicia cuando el usuario selecciona una vista que la contiene a ella, se muestran varias opciones y se termina cuando se interactúa con uno de sus botones levantando siempre otra vista.	
Vistas asociadas	Narración, Texto, Autor, Entretenimiento, Inicio, Cuentos, Salir.	
Referencias	RF21, RF23, RF3, RF4, RF5, RF6, RF7	
Precondiciones	Debe haberse ejecutado una vista que la contenga a ella.	
Poscondiciones		
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	Elementos de la Vista
1. Levanta una vista que contenga la misma.	2. Se muestran 9 íconos (Entretenimiento, Narración, Autor, Texto, Inicio, Salir, Idioma1, Idioma2 e Idioma3)	2.1 Los íconos se encontrarán de la siguiente forma: tres en la parte derecha de la pantalla, y tres en la parte izquierda de la pantalla.
3. Selecciona un botón.	4. Si el usuario seleccionó el ícono Entretenimiento, se ejecuta la vista Entretenimiento y se culmina la vista.	4.1 Se ilumina el ícono Entretenimiento y se reproduce el sonido explicando la acción del ícono.

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

	5. Si el usuario seleccionó el ícono Narración se ejecuta la vista Narración y se culmina la vista.	5.1 Se ilumina el ícono Narración y se reproduce el sonido explicando la acción del ícono.
	6. Si el usuario seleccionó el ícono Autor se ejecuta la vista Autor y se culmina la vista.	6.1 Se ilumina el ícono Autor y se reproduce el sonido explicando la acción del ícono.
	7. Si el usuario seleccionó el ícono Texto se ejecuta la vista Texto y se culmina la vista.	7.1 Se ilumina el ícono Texto y se reproduce el sonido explicando la acción del ícono.
	8. Si el usuario seleccionó el ícono Inicio se ejecuta la vista Cuento y se culmina la vista.	8.1 Se ilumina el ícono Inicio y se reproduce el sonido explicando la acción del ícono.
	9. Si el usuario seleccionó el ícono Salir se ejecuta la vista Salir y se culmina la vista.	9.1 Se ilumina el ícono Salir y se reproduce el sonido explicando la acción del ícono.
	10. Si el usuario seleccionó el ícono Idioma1 el cuento seleccionado se mostrará en el idioma correspondiente al botón.	10.1 Se ilumina el ícono Idioma1.
	11. Si el usuario seleccionó el ícono Idioma2 el cuento seleccionado se mostrará en el idioma correspondiente al botón.	11.1 Se ilumina el ícono Idioma2.
	12. Si el usuario seleccionó el ícono Idioma3 el cuento seleccionado se mostrará en el idioma correspondiente al botón.	12.1 Se ilumina el ícono Idioma3.
Curso Alterno de los eventos		
Acción	Curso Alterno	

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

-	-		
Prioridad	Crítica		
Mejoras			
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I-BE-001	Imagen botón Entretenimiento.	Existente.
	I-BN-001	Imagen botón Narración.	Existente.
	I-BA-001	Imagen botón Autor.	Existente.
	I-BT-001	Imagen botón Texto.	Existente.
	I-BI-001	Imagen botón Inicio.	Existente.
	I-BS-001	Imagen botón Salir.	Existente.
	I-BI-001	Imagen botón Idioma1.	Existente.
	I-BI-002	Imagen botón Idioma2.	Existente.
	I-BI-003	Imagen botón Idioma3.	Existente.
Video o animación	-	-	-
Sonido	S-BE-001	Sonido botón Entretenimiento.	Existente.
	S-BN-001	Sonido botón Narración.	Existente.
	S-BA-001	Sonido botón Autor.	Existente.
	S-BT-001	Sonido botón Texto.	Existente.
	S-BI-001	Sonido botón Inicio.	Existente.
	S-BS-001	Sonido botón Salir.	Existente.
Texto	-	-	-
Reglas Pedagógicas	-		
Prototipo de Interfaz de Usuario de la Vista			

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

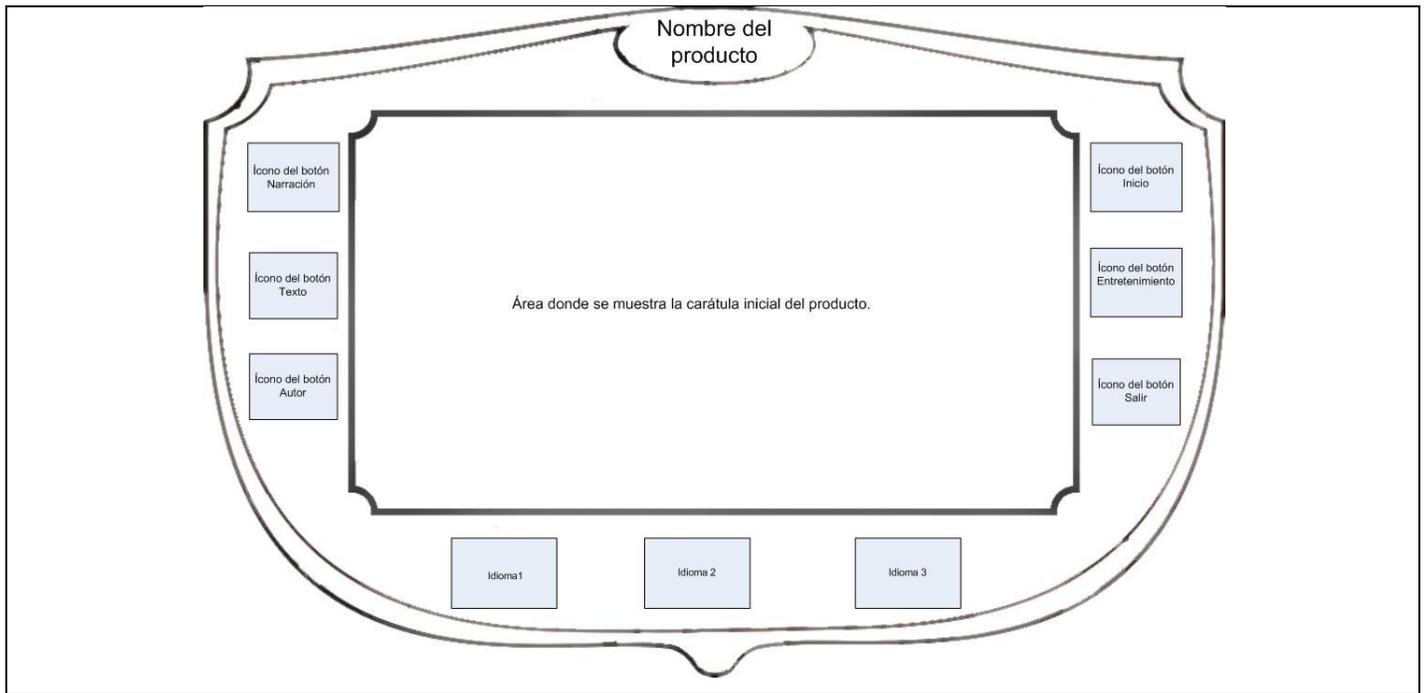


Tabla 3. Descripción textual de la Vista de Presentación *Común*

Las descripciones textuales del resto de las vistas de la aplicación pueden ser consultadas en los Anexos 1-6.

2.5.2 Diagramas de Estructura de Presentación.

Los Diagramas de Estructura de Presentación (DEP) permiten representar la estructura que tendrán las interfaces de usuario, estableciendo una organización lógica de los elementos conformantes de las mismas y dejándoles a los diseñadores gráficos la función de decidir dónde y cómo serán en términos visuales dichos elementos (38).

Para la mejor estructuración del modelo ApEM-L define dos nuevos tipos de clases: la clase Estáticos y la clase Interacción, las cuales seccionarán los elementos que cumplan con cada una de las características que denotan los propios nombres. La clase Estáticos agrupará los componentes que solo tiene como función visualizar información, pero que no permiten interacción con el usuario. Todo lo contrario con los agrupados bajo la clase Interacción, los cuales serán los elementos de la vista que permiten interacción

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

del usuario con el sistema informático modelado. Para cada una de las vistas de presentación se realiza DEP (20). A continuación se muestran los DEP de las diferentes vistas de presentación del módulo Cuento del producto “Mis Mejores Cuentos”:

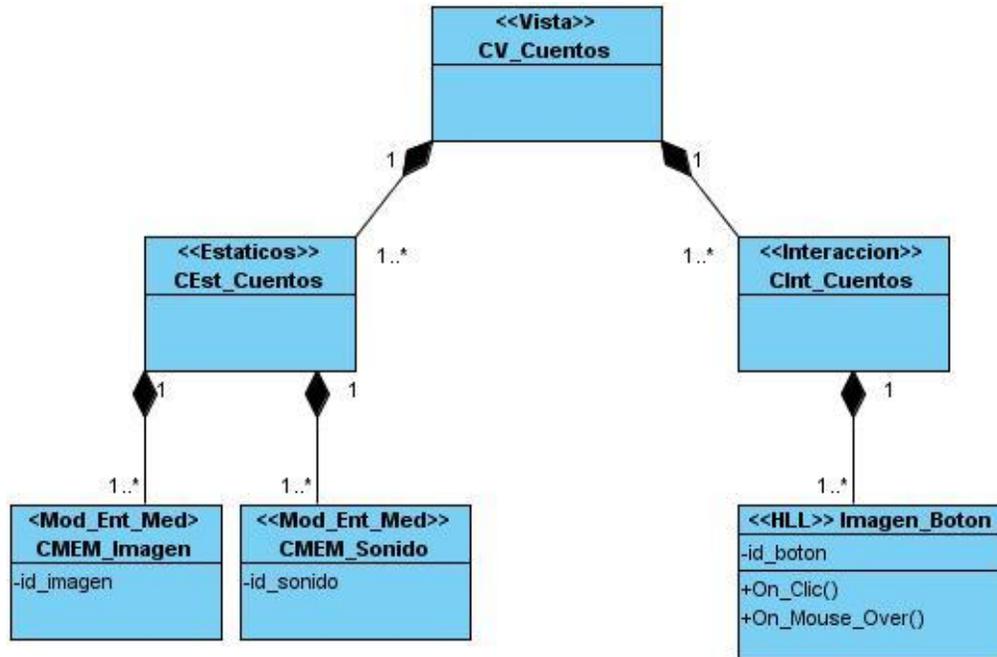


Figura 5. DEP Vista de Presentación *Cuentos*

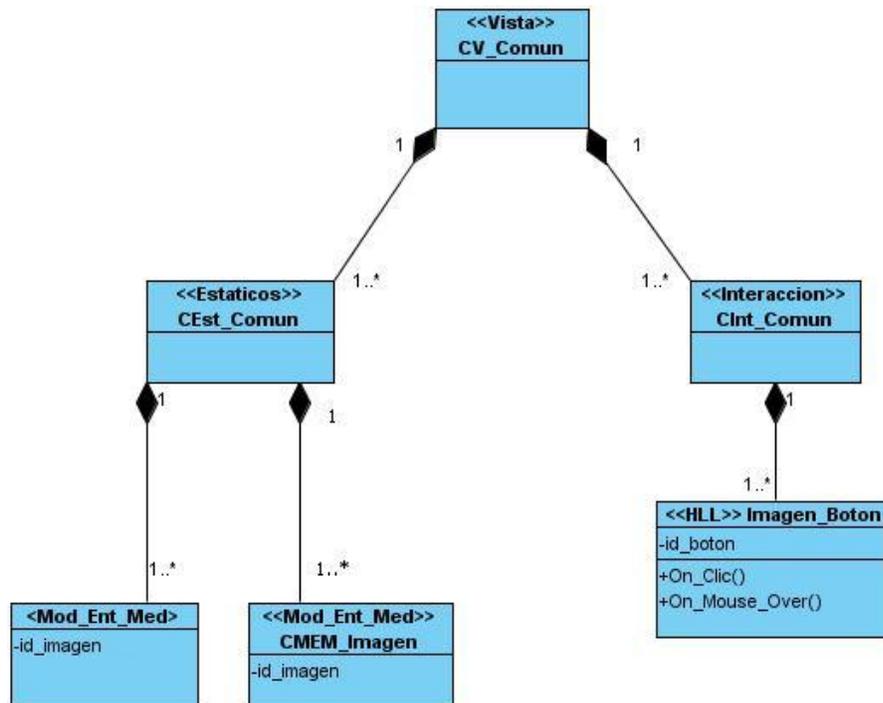


Figura 6. DEP Vista de Presentación *Común*

Los Diagramas de Estructura de Presentación del resto de las vistas del módulo pueden ser consultados en los Anexos 7-12.

2.5.3 Diagramas de Estructura de Navegación

Debido a la ineficiencia del lenguaje base UML para modelar la navegación en el software educativo se ha definido el Diagrama de Estructura de Navegación (DEN) como un artefacto de ApEM-L. El DEN representa la estructura y organización de los elementos de navegación de sistemas multimedia incorporando nuevos estereotipos restrictivos que facilitan la comprensión del modelo como son: la clase botón, la clase consulta y la clase menú. Estos estereotipos son elementos de composición de la clase vista que permiten el enlace con otras clases vistas del modelo (20).

Este diagrama forma parte de una de las áreas estructural llamada gestión de modelo que propone el lenguaje ApEM-L. Es aplicable de manera general a todo el módulo.

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

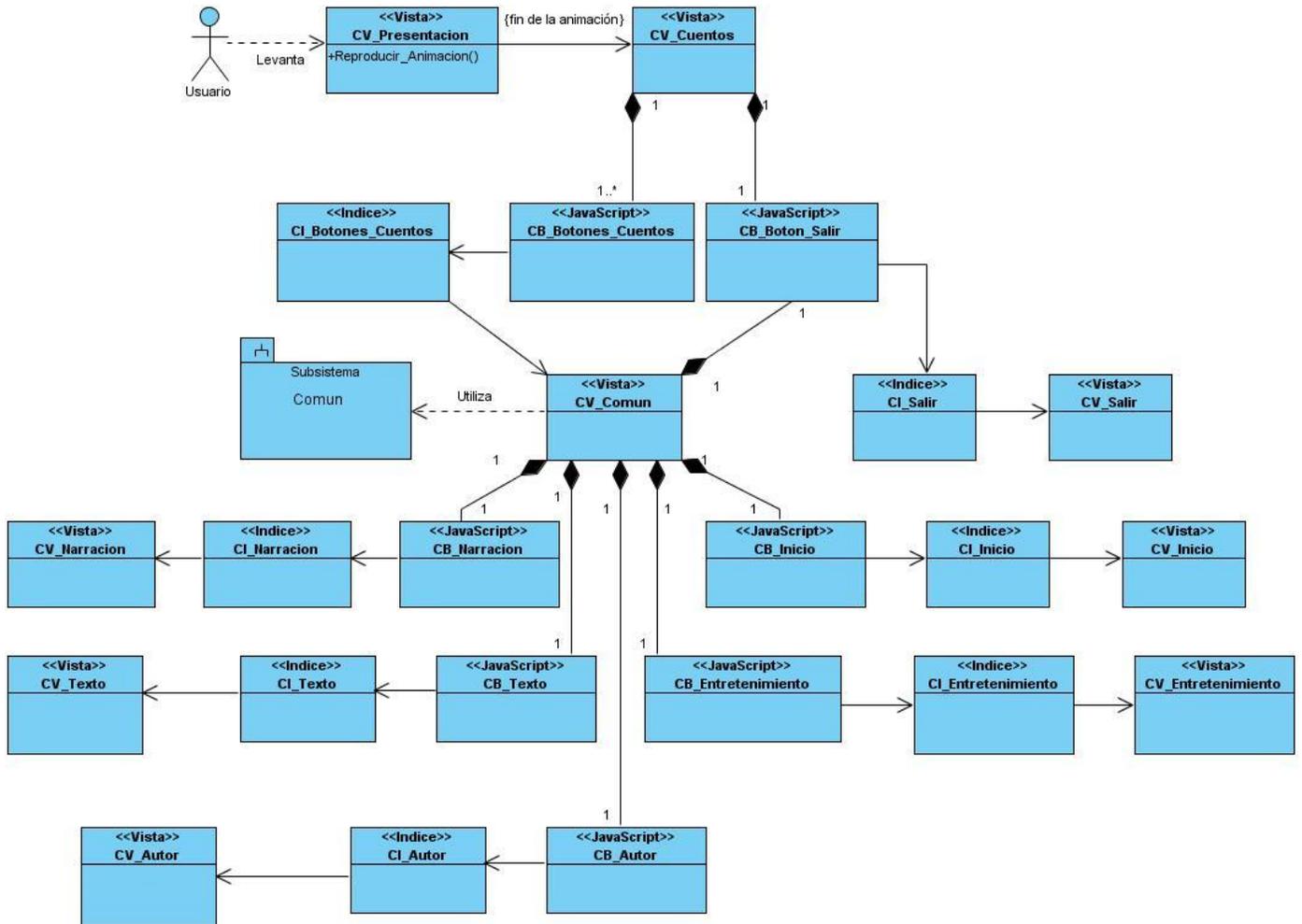


Figura 7. DEN del módulo Cuento

2.6 Diagrama de clases

El diagrama de clases no es más que conceptos que son modelados como clases describiendo cada una un conjunto de objetos que almacenan información y se comunican para implementar su comportamiento. La información almacenada se representa como atributos de estas clases y las operaciones a través de los métodos de dichas clases. Son los diagramas más comunes en el modelado de sistemas orientados a objetos (41).

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

ApEM-L propone algunas modificaciones para este tipo de diagrama, que consisten en dividirlo en dos grandes zonas, la de la izquierda dedicada al árbol jerárquico de las clases modelo entidad medias que representan los recursos mediáticos de la aplicación y en la zona de la derecha del diagrama las clases que controlan la lógica del negocio de la aplicación propiamente dicha. La zona de la derecha se divide en 4 zonas, la primera dedicada a las clases vista, la contigua a esta y en el extremo superior derecho dedicada a las clases controladoras, inmediatamente debajo de esta sección, la destinada a las clases modelo, quedando una banda inferior derecha dedicada en su extremo derecho a las clases modelo entidad persistentes para el tratamiento de la información persistente de la aplicación; y en el extremo izquierdo las clases correspondientes al Lenguaje de Alto Nivel (HLL) (20).

Este tipo de diagrama se representa para cada una de las vistas del módulo del producto “Mis Mejores Cuentos”. Seguidamente se muestran los diagramas de clases de las vistas más importantes del módulo a construir.

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

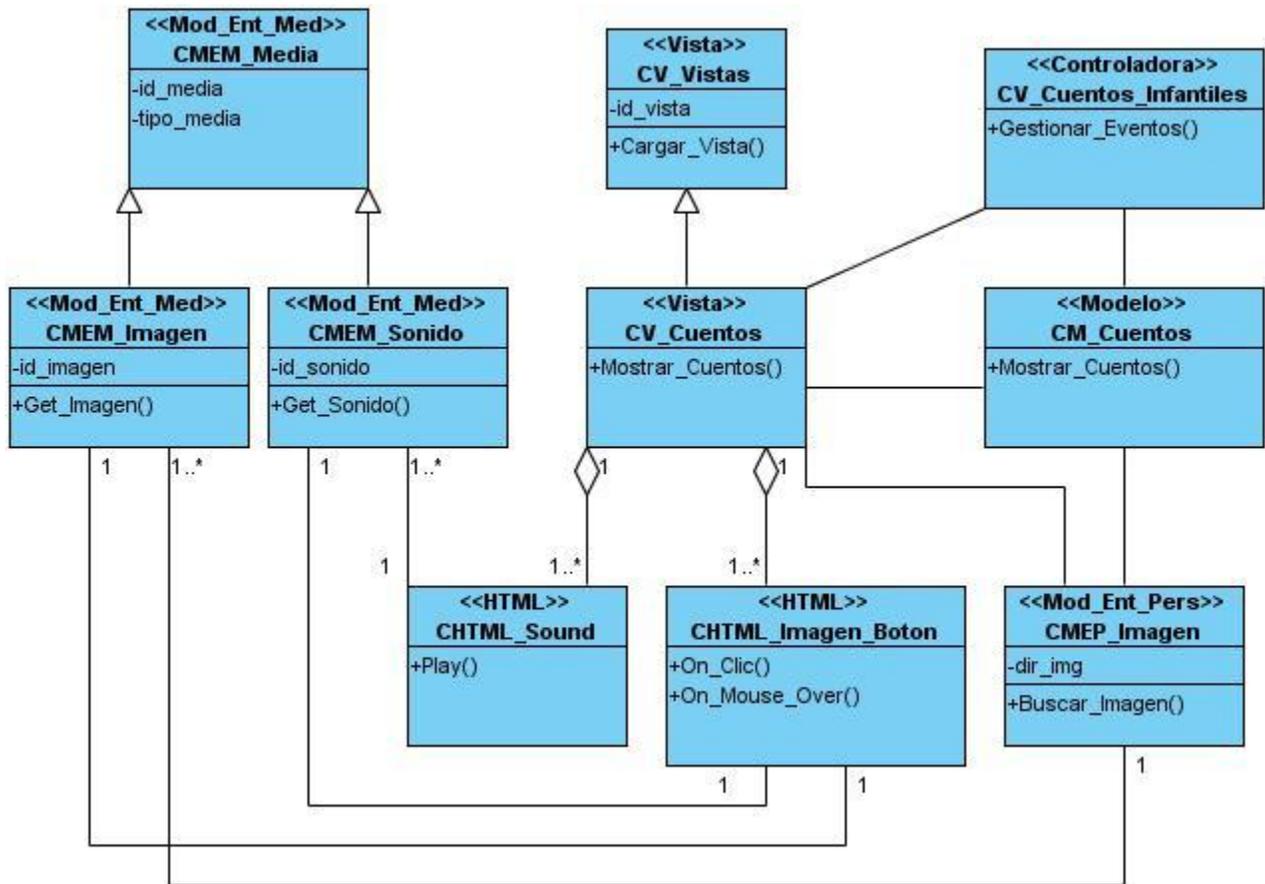


Figura 8. Diagrama de Clases de la Vista de Presentación *Cuentos*

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

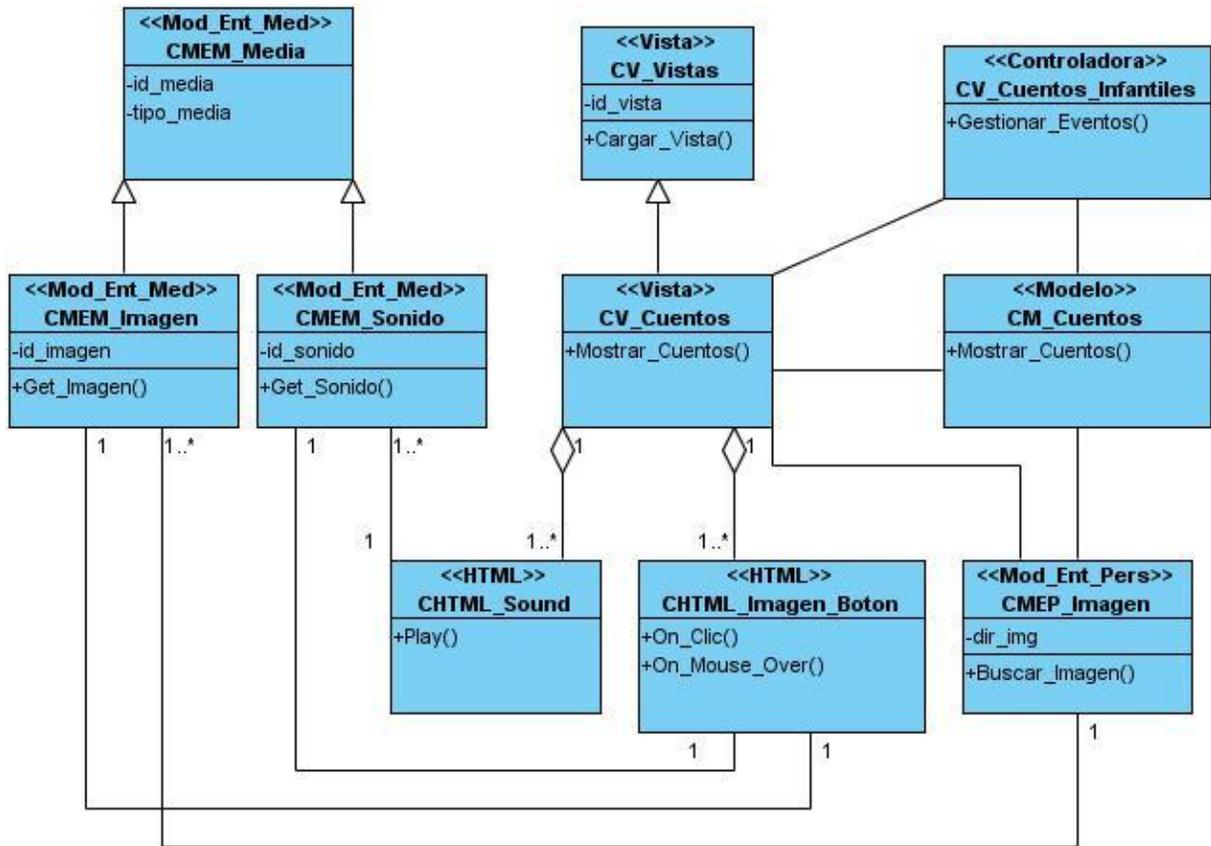


Figura 9. Diagrama de Clases de la Vista de Presentación *Común*

Los Diagramas de Clases del resto de las vistas del módulo pueden ser consultados en los Anexos 13-18.

2.7 Diagramas de Secuencias

Un Diagrama de Secuencias es un tipo de diagrama de interacción que muestra los objetos con sus interacciones en el tiempo representadas como mensajes (38).

ApEM-L propone una modificación para este tipo de diagrama, que consisten en la representación del tiempo cuando su incorporación sea necesaria, por la importancia del control de esta variable en las aplicaciones educativas y en especial en el trabajo con las medias continuas como sonido, video y animaciones (20).

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

Este tipo de diagrama se representa para cada una de las vistas del módulo. Seguidamente se muestran los diagramas de secuencias de las vistas más importantes del módulo a construir.

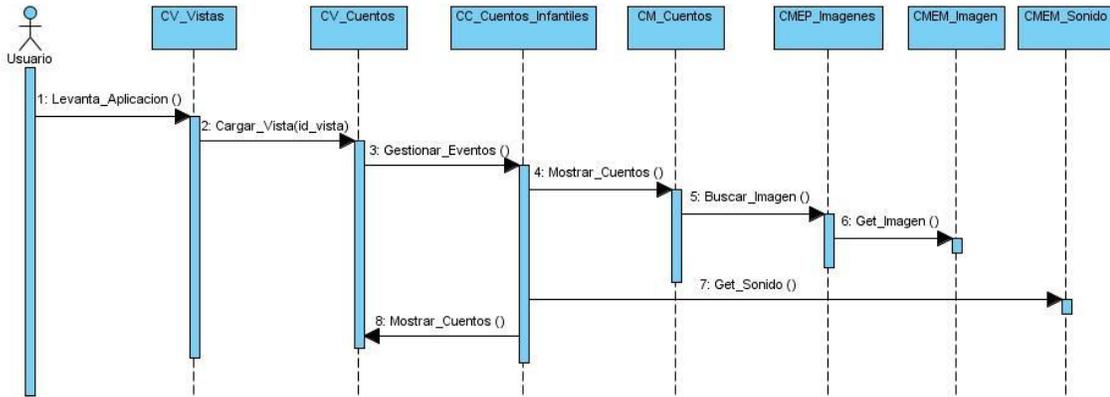


Figura 10. Diagrama de Secuencia de la Vista de Presentación *Cuento*

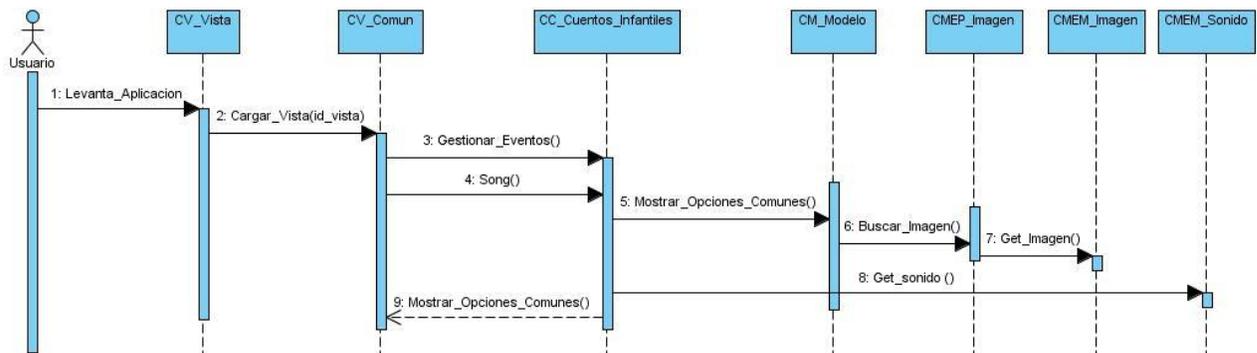


Figura 11. Diagrama de Secuencia de la Vista de Presentación *Común*

Los Diagramas de Secuencia del resto de las vistas del módulo pueden ser consultados en los Anexos 19-24.

2.8 Conclusiones

En este capítulo se identificaron y describieron los principales conceptos del entorno donde se usará el software, se realizó el modelamiento del dominio y se especificaron seis requerimientos funcionales y siete no funcionales que debe cumplir el módulo a desarrollar. Además se modeló la Vista de Gestión del Modelo, compuesta por cuatro subsistemas. También se hizo una descripción textual de dos de las

CAPÍTULO II

ANÁLISIS Y DISEÑO DEL MÓDULO

principales Vistas de Presentación y se mostraron los Diagramas de Estructura de Presentación de estas vistas así como el Diagramas de Estructura de Navegación correspondiente al módulo en general. Se obtuvieron además, los artefactos fundamentales correspondientes al diseño de la solución propuesta, se representaron dos diagramas de clases del diseño de los ocho modelados y dos de los ocho diagramas de secuencia que se obtuvieron.



CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

3.1 Introducción

El flujo de trabajo de Implementación en la fase de Elaboración va encaminado a implementar la arquitectura que se ha definido. Está fuertemente determinado por el lenguaje de programación. También describe cómo los elementos del Modelo del Diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo con los nodos específicos en el Modelo de Despliegue. Los Diagramas de Despliegue y Componentes conforman lo que se conoce como un Modelo de Implementación al describir los componentes, su organización y la dependencia entre los nodos físicos en los que funcionará el módulo (42).

En este capítulo se documenta el proceso de implementación de los elementos identificados durante la realización del diseño. Para ello se modela el diagrama de despliegue, donde se muestra la distribución física de los elementos de hardware que conformarán el módulo y las relaciones entre ellos; y el diagrama de componentes, donde se representa la organización y las dependencias lógicas que existen entre los componentes del módulo. Además se realizará la validación de la propuesta de solución planteada para agrupar y dar respuesta a los Requisitos Funcionales obtenidos en el proceso inicial. Se dará un pequeño bosquejo teórico de los temas a tratar, una explicación de los tipos de pruebas y niveles que se aplican, así como la evaluación de los resultados esperados y obtenidos luego de aplicarle un conjunto de pruebas a la propuesta, para de esta forma fundamentar su correcta implementación y funcionamiento.

3.2 Modelo de Implementación

El Modelo de Implementación representa la composición física de la implementación en términos de subsistemas de implementación y elementos de implementación. Describe cómo los elementos de diseño se implementan en componentes (15).

Según los mismos autores (15), se considera el artefacto más significativo del flujo de trabajo de Implementación, debido a la importancia que tiene para los desarrolladores comprender el funcionamiento del sistema desde el punto de vista de componentes y sus relaciones. Los Diagramas de Despliegue y

CAPÍTULO III

IMPLEMENTACIÓN Y PRUEBA

Componentes conforman lo que se conoce como un Modelo de Implementación al describir los componentes, su organización y la dependencia entre los nodos físicos en los que funcionará el módulo.

3.2.1 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo (38).

ApEM-L extiende la semántica de UML para este tipo de diagrama al incorporar los elementos de organización en paquetes asociados al patrón arquitectónico MVC-E (Modelo – Vista – Controlador – Entidad) y sus relaciones de funcionamiento (20).

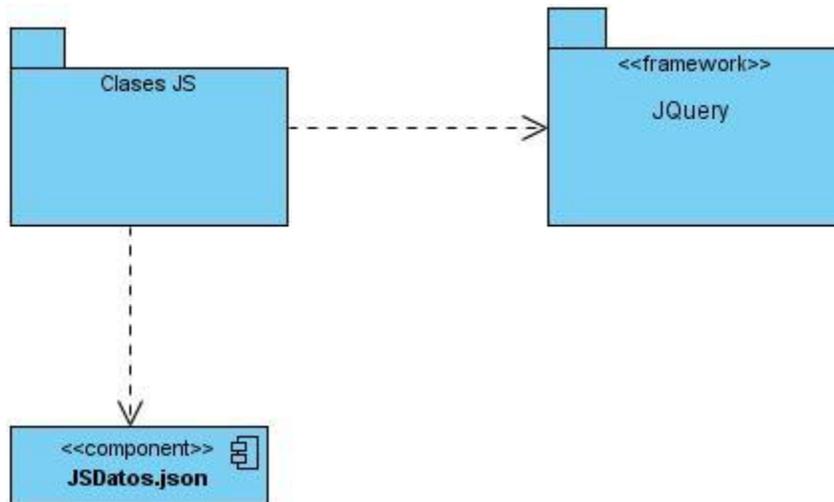


Figura 12. Diagrama de Componentes General

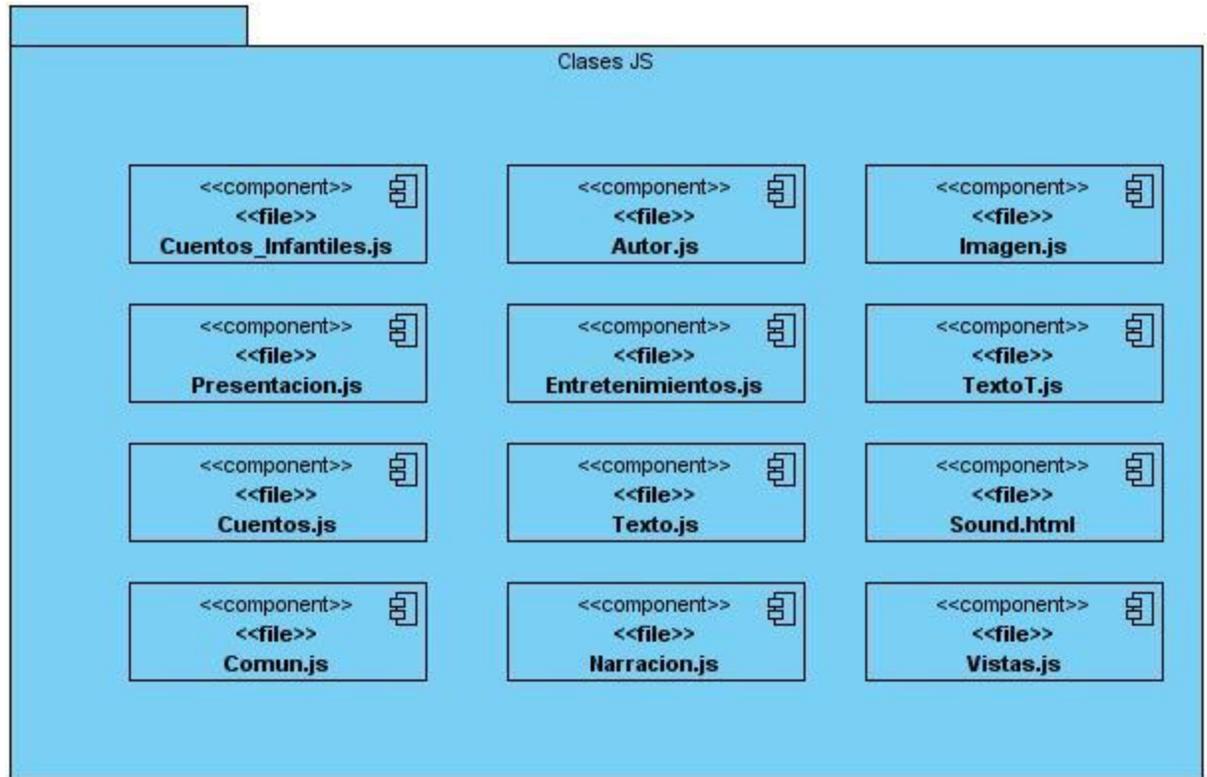


Figura 13. Diagrama de Componentes del paquete Clases JS

3.2.2 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas entre los componentes del hardware y el software, es decir, la configuración de los elementos de procesamiento entiendo de ejecución y los componentes del software. También un Diagrama de Despliegue es un grafo de nodos, unidos por conexiones de comunicación, donde un nodo puede contener instancias de componentes, un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria (43).

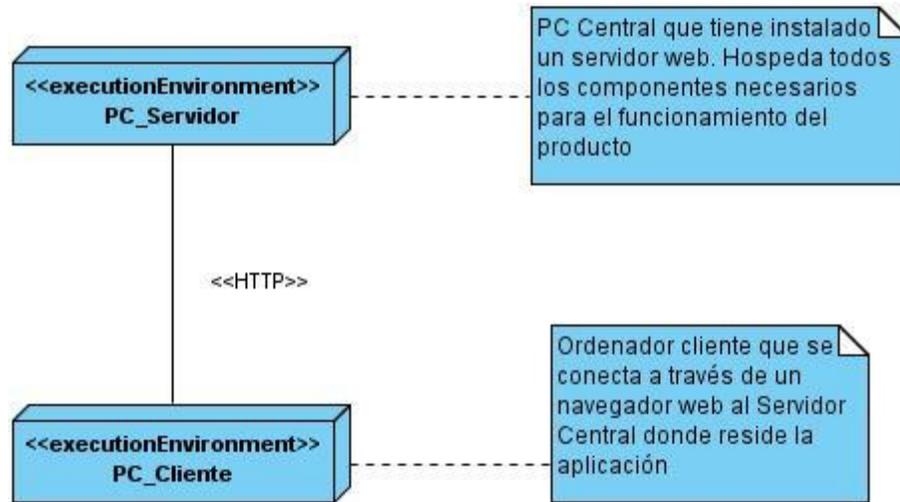


Figura 14. Diagrama de Despliegue

3.3 Validación de la solución propuesta

El proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Las aplicaciones de software han crecido en complejidad y tamaño, y por consiguiente también en costos (44), por lo que se hace necesario que el proceso de pruebas se lleve a cabo con calidad y eficiencia, logrando la aceptación del producto por parte del usuario final.

3.3.1 Métodos de Prueba

Los métodos de prueba definen qué estrategia seguir en cuanto a la verificación y validación del sistema, ya que están diseñados con el propósito de descubrir fallos y no para demostrar que el software funciona, siendo más razonable diseñar pruebas en aquellas partes donde la probabilidad de fallo es mayor (45). Las pruebas se enfocan sobre la lógica interna del software y las funciones externas, para lo cual se definen los siguientes métodos:

Método de Caja Blanca: Se basa en el examen de los detalles procedimentales. Se comprueban los caminos lógicos proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado (46).

CAPÍTULO III

IMPLEMENTACIÓN Y PRUEBA

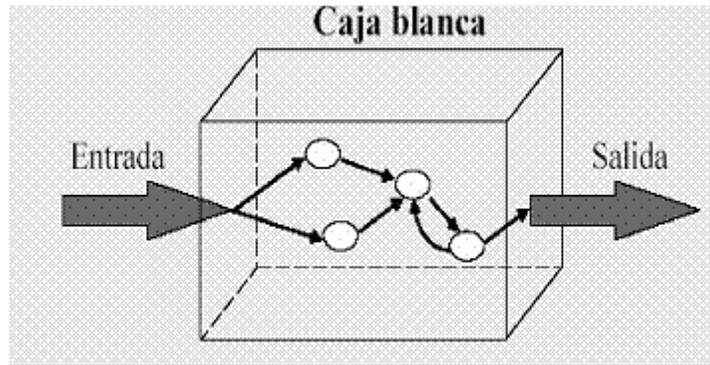


Figura 15. Método de Caja Blanca

Método de Caja Negra: Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software. Estas pruebas permiten encontrar:

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y terminación.

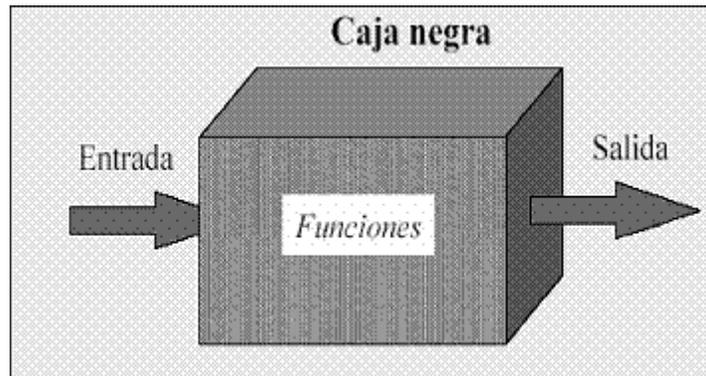


Figura 16. Método de Caja Negra

3.3.2 Pruebas aplicadas al software

Para encontrar la mayor cantidad de errores y hacer una buena validación de software se decide realizar pruebas de Aceptación y de Funcionalidad. Dichas pruebas pertenecen al método de caja negra, las cuales están regidas por los casos de pruebas, que fueron generados a partir de la especificación de casos de uso. Los casos de prueba representan los requisitos que se esperan que cumpla la aplicación, por lo que al escribirlos también se documenta el sistema.

Seguidamente se muestra una explicación de las pruebas utilizadas:

Pruebas de Aceptación: Prueba de aceptación del usuario es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido (47).

Pruebas de Funcionalidad: Pruebas orientadas a determinar el cumplimiento de las funcionalidades del sistema respecto a los requerimientos de los usuarios (48).

- ✓ **Función:** Pruebas fijando su atención en la validación de las funciones, métodos, servicios, caso de uso.
- ✓ **Seguridad:** Asegurar que los datos o el sistema solamente es accedido por los actores deseados.
- ✓ **Volumen:** Enfocada en verificar las habilidades de los programas para manejar grandes cantidades de datos, tanto como entrada, salida o residente en la BD.

CAPÍTULO III

IMPLEMENTACIÓN Y PRUEBA

Los resultados de las pruebas pueden ser consultados en los anexos del 25 al 32.

3.4 Conclusiones

Para detallar descriptivamente la solución propuesta, se realizaron los diagramas de despliegue y componentes, quedando confeccionada la vista estática del sistema lo que proporciona tanto la distribución física como la representación lógica de los elementos del diseño, comportándose estos últimos como componentes de un modelo que llevan implícito las relaciones de dependencias que existe entre ellos, los paquetes y los subsistemas. Para evaluar la solución desarrollada se abarcaron los métodos, niveles y tipos de prueba expuestos anteriormente, arrojando resultados visibles, los cuales validan la calidad del producto construido.

CONCLUSIONES GENERALES

Con el propósito de darle cumplimiento al objetivo general y a la problemática planteada en el presente trabajo, se han llevado a cabo satisfactoriamente cada una de las tareas que fueron trazadas al comienzo del mismo.

- ✓ El estudio realizado de productos multimedia con propósitos educativos, en el ámbito nacional e internacional demostró la necesidad de realizar un software educativo multiplataforma con herramientas libres para niños de 3 a 5 años de edad, rompiendo los esquemas del uso de herramientas propietarias para la implementación de productos multimedia
- ✓ El análisis de las tecnologías, herramientas y metodologías posibilitaron identificar las funcionalidades de la aplicación y diseñar el software orientado a la tecnología multimedia, con una interfaz agradable capaz de captar la atención de los niños y una fácil navegabilidad sin la ayuda de terceras personas.
- ✓ Con la implementación del módulo Cuento del producto “Mis Mejores Cuentos” en plataforma web y con tecnología multimedia, se garantiza la presentación de forma interactiva de los cuentos infantiles para niños de la edad de 3 a 5 años, posibilitando el desarrollo de las habilidades educacionales ajustadas a la edad, tales como: enseñanza transmitida por el cuento, reconocimiento de colores y sonidos, identificación y asociación de figuras, ejercitación de la memoria a largo plazo, destreza del infante con el mouse, así como la familiarización con la computación.
- ✓ La posibilidad de selección de idioma facilita la comercialización del producto “Mis Mejores Cuentos”.

RECOMENDACIONES

A partir de los resultados o beneficios que proporciona este trabajo de diploma, se proponen las siguientes recomendaciones.

- ✓ Utilizar la documentación generada como estándar en otros proyectos con características similares, permitiendo que sirva como base para el desarrollo del modelado con APEM – L.
- ✓ Se recomienda además lograr la distribución del producto “Mis Mejores Cuentos” en un paquete de instalación suficientemente sencillo como para poder ser distribuido en las escuelas primarias, Círculos Infantiles, y Joven Club del país.
- ✓ Integrar el módulo Administración al producto “Mis Mejores Cuentos” para facilitar el cambio de los cuentos en versiones posteriores del producto sin necesidad de rediseñar toda la colección.



REFERENCIAS BIBLIOGRÁFICAS

1. Librinsula La Isla de Los Libros. [En línea] 18 de Noviembre de 2005. [Citado el: 18 de Enero de 2010.] <http://librinsula.bnjm.cu/1-205/2005/noviembre/98/dossier/dossier180.htm>. ISSN: 1810-4479.
2. **Graells, Dr. Pere Marquès.** IMPACTO DE LAS TIC EN EDUCACIÓN: FUNCIONES Y LIMITACIONES. [En línea] 8 de Agosto de 2010. [Citado el: 20 de Enero de 2011.] <http://peremarques.pangea.org/siyedu.htm>.
3. **DELGADO SIERRA, GUSTAVO ADOLFO.** Educación y Tecnología. *Blog de Educación y Tecnología de Víctor Romero.* [En línea] Enero de 2010. [Citado el: 12 de Febrero de 2011.] <http://educacionytecnologiaromero.blogspot.com/2010/03/educacion-y-tecnologia-ii-enero-abril.html>.
4. **Romero, Leudis.** *Análisis y diseño del software educativo cuentos infantiles para niños en edad preescolar.* C.Habana : s.n., 2009.
5. **Membrides Espinosa, Antonio.** SLAM-C++. *Soluciones Libres para Aplicaciones Multimedia basadas en C++.* [En línea] Cordoba. [Citado el: 18 de Enero de 2011.] <http://slam-c.nireblog.com/cat/multimedia>.
6. **DIMAGIN.** imagin Web Development. [En línea] [Citado el: 18 de Febrero de 2011.] <http://www.dimagin.net/es/main.php>.
7. **Autores, Colectivo de.** *Tendencias pedagógicas en la realidad educativa actual.* Tarija-Bolivia : Editorial Universitaria, 2008.
8. **Cataldi, Zulma.** *Metodología de diseño, desarrollo y evaluación.* s.l. : Facultad de Informática. UNLP, 2000. ISBN 960-34-0204-2.
9. **Gutiérrez Topete, Angelina.** Maestría en Comunicación y Tecnologías Educativas. *Maestría en Comunicación y Tecnologías Educativas.* [En línea] 18 de Marzo de 2007. [Citado el: 15 de Enero de 2011.] <http://tecno-educativa.blogspot.com/2007/03/software-definicion-y-caractersticas.html>.
10. **Alvares Alfonso, Karenia.** Informática Educativa. *Informática Educativa.* [En línea] 23 de Junio de 2009. [Citado el: 14 de Enero de 2011.] <http://blogs.rimed.cu/infoedu/2009/06/23/sobre-el-concepto-de-software-educativo/>.

REFERENCIAS BIBLIOGRÁFICAS

11. **Ramos Pérez, Lourdes, y otros.** *Revistas Médicas Cubanas. Revistas Médicas Cubanas.* [En línea] [Citado el: 10 de Enero de 2011.] http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm.
12. **Figueroa, Roberth G., Solís, Camilo J. y Cabrera, Armando A.** *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES.* Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación : s.n., 2006.
13. **Guerra Anaya, Ivonne Isabel y Casanova Alvarez, Alexey.** *Sistema de Control de Estado de las Especificaciones de Casos de Uso y Manuales de Usuario del SIIPOL.* La Habana : s.n., 2010.
14. **Cabrera Medina, Andy.** *Ingeniería de Requisitos aplicada al Módulo de Sustancias Químicas del Sistema de Investigación Policial del CICPC .* La Habana : s.n., 2010.
15. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software. El Proceso Unificado de Desarrollo de Software.* La Habana : Félix Varela, 2004.
16. **IBM.** *Página oficial de RUP. IBM.* [En línea] 2010. [Citado el: 12 de Enero de 2011.] <http://www-01.ibm.com/software/awdtools/rup/index.html>.
17. **Catalá Matienzo, Ivette.** *Ingeniería de Requerimientos aplicados a la plataforma de Infodrez 2.0.* La Habana : s.n., 2010.
18. —. *Ingeniería de Requerimientos aplicados a la plataforma de Infodrez 2.0.* La Habana : s.n., 2008.
19. **Velázquez Corona, Dianick.** *Análisis y Diseño del Módulo Resultados de la Colección Multisaber.* La Habana : s.n., 2008.
20. **Febe, Ricardo Ángel.** *ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedias en la UCI.* Ciudad de La Habana : s.n., 2007.
21. **Pérez Chirino, Yosbel.** *Herramienta para la gestión de torneos ajedrecísticos, versión 2.0.* La Habana : s.n., 2010.
22. *Ciencia yTécnica Administrativa . Ciencia yTécnica Administrativa .* [En línea] [Citado el: 10 de Enero de 2011.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
23. *Free Download Manager. Free Download Manager.* [En línea] 2007. [Citado el: 10 de Diciembre de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(Iglesia_Anglicana\)_%5BMac_OS_X_cuenta_14717_p/..](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/..)
24. **Hdez. Peña, Dennys, y otros.** *SACCEM: SISTEMA AUTOMATIZADO CUBANO PARA EL CONTROL DE EQUIPOS MÉDICOS.* La Habana : s.n.

REFERENCIAS BIBLIOGRÁFICAS

25. **Valdés, Damián Pérez.** Maestros del Web. *Maestros del Web*. [En línea] 2010. [Citado el: 17 de Enero de 2011.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
26. **de la Torre, Aníbal.** PHPNuke. *PHPNuke*. [En línea] 2006. [Citado el: 19 de Enero de 2011.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
27. **Bonilla, Luis.** Cursos de Multimedia. *Cursos de Multimedia*. [En línea] 2009. [Citado el: 19 de Enero de 2001.] <http://www.cursopaginasweb.net/curso-paginas-web/introduccion-javascript.htm>.
28. **Martínez Padrón, Jorge.** *Desarrollo de un módulo para la gestión de contenidos en la versión multiplataforma de la*. La Habana : s.n.
29. Observatorio Tecnológico. *Observatorio Tecnológico*. [En línea] Instituto de Tecnologías Educativas, Enero de 2011. [Citado el: 16 de Enero de 2011.] <http://recursostic.educacion.es/observatorio/web/ca/software/software-general/548-luis-garcia>.
30. **Collins-Sussman, Ben, Fitzpatrick, Brian W. y Pilato, C. Michael.** Control de versiones Open Source de siguiente generación. *Control de versiones Open Source de siguiente generación*. [En línea] [Citado el: 19 de Enero de 2011.] <http://svnbook.red-bean.com/nightly/es/svn-book.html#svn-ch-1-sect-1>.
31. **Vindas, Rolando.** Intergraphic Designs. *Intergraphic Designs*. [En línea] 2010. [Citado el: 19 de Enero de 2011.] <http://www.intergraphicdesigns.com/blog/2008/09/29/resumen-sobre-ventajas-de-utilizar-subversion/>.
32. **Sánchez, Emilio A., Letelier, Patricio y Conos, José H.** *Mejorando la gestión de Historias de Usuario en eXtreme programming*. Valencia : s.n.
33. **Márquez Díaz, José y Sampedro, Leonardo and Vargas, Félix.** *ciruelo.uninorte.edu.co*. *ciruelo.uninorte.edu.co*. [En línea] 2002. [Citado el: 12 de Diciembre de 2010.] http://ciruelo.uninorte.edu.co/pdf/ingenieria_desarrollo/12/instalacion_y_configuracion_de_apache.pdf..
34. **Bradshaw Venzant, Maikel y Rodríguez Martini, Jorge Luis.** *Análisis, diseño e implementación del Portal WAP del SIIPOL Móvil*. La Habana : s.n., 2010.
35. **Vera, Pablo.** Adictos a la red. *Adictos a la red*. [En línea] 2009. [Citado el: 12 de Enero de 2011.] <http://www.adictosalared.com/aptana-studio-excelente-entorno-gratuito-para-desarrollo-web/>.
36. **BrotherSoft.com.** Brothersoft. *Brothersoft*. [En línea] 2007. [Citado el: 15 de Enero de 2011.] <http://es.brothersoft.com/Aptana-136930.html>.

REFERENCIAS BIBLIOGRÁFICAS

37. **Delgado Dapena., MSc. Martha D.** *Definición del modelo del negocio y del dominio utilizando.* La Habana : Centro de Estudios de Ingeniería de Sistemas.
38. **Rosales Velazquez, Norlen.** *SOFTWARE EDUCATIVO PICO MESTRO.* La Habana : s.n., 2009.
39. **Galán Garcés, Yaritza.** *ANÁLISIS Y DISEÑO DEL MÓDULO TEMAS DE LA COLECCIÓN MULTISABER.* La Habana : UCI, 2008.
40. **E.V.A. .** ECURED. *ECURED.* [En línea] 2010. [Citado el: 1 de Marzo de 2011.] http://www.ecured.cu/index.php/Flujo_de_Trabajo_Requerimiento.
41. **Leon Torres, Susel de la Caridad y Calderin Cosculluela, Saskia.** *Analisis y Diseño del módulo Juegos de la Colección Multisaber.* La Habana : UCI, 2008.
42. **Delgado Fernandez, Daylen y Benavides Triana, Yasnel.** *Análisis y Diseño del módulo Resultados de la Colección Multisaber en su versión multiplataforma.* La Habana : UCI, 2010.
43. **Marca Huallpara, Hugo y Quisbert Limachi, Susana Nancy.** *ANALISIS Y DISEÑO DE SISTEMAS II TRABAJO DE INVESTIGACIÓN Y EXPOSICIÓN “Diagrama de Despliegue”.* La Habana : s.n., 2010.
44. **Perez, Prof Cesar A.** IUTJAA / Informatica . [En línea] WordPress, Enero de 2011. [Citado el: 22 de Abril de 2011.] <http://bycpx.wordpress.com/2011/01/>.
45. **Terry Morales, Gabriel.** *Análisis, diseño e implementación de la capa lógica del negocio y acceso a datos del módulo Estadística perteneciente al Sistema de Investigación e Información Policial.* La Habana : UCI, 2010.
46. **UCI.** Material de caja blanca y caja negra. *Entorno Virtual de Ap.* [En línea] 2011. [Citado el: 14 de Abril de 2011.] http://eva.uci.cu/file.php/259/Curso_2010-2011/Semana_9/Conferencia_7/Materiales_Complementarios/Material_de_caja_b_y_caja_n.pdf.
47. —. Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje.* [En línea] 2011. [Citado el: 24 de Abril de 2011.] http://eva.uci.cu/file.php/259/Curso_2010-2011/Semana_9/Conferencia_7/Materiales_Basicos/Documentacion_sobre_Pruebas.pdf.
48. **Jaldín Rosales, Rolando.** Sistemas de Información. Herramientas y Tecnologías. *Sistemas de Información. Herramientas y Tecnologías.* [En línea] 5 de Noviembre de 2010. [Citado el: 24 de Abril de 2011.] <http://rolandojaldin.blogspot.com/2010/11/etapa-de-prueba-fase-de-transicion-rup.html>.
49. **Cruz, Yoel, Rodríguez, Mirtha y Gual, Jorge Félix.** GestioPolis. *GestioPolis.* [En línea] 2008. [Citado el: 2 de Diciembre de 2010.] <http://www.gestiopolis.com/organizacion-talento/pagina-web-herramienta-aprendizaje.htm>.

REFERENCIAS BIBLIOGRÁFICAS

50. **Matos Pérez, Sulay.** *Análisis y diseño del módulo Ejercicios de la colección Multisaber.* La Habana : s.n., 2008.

GLOSARIO DE TÉRMINOS

Flash: Programa que facilita la utilización de tecnología en la Web, permitiendo la creación de animaciones vectoriales. El interés en el uso de gráficos vectoriales es que éstos permiten llevar a cabo animaciones que tardan poco tiempo en ser cargadas por el navegador.

HTML: Lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

CSS: Lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Canvas: Etiqueta o elemento en HTML5 que permite la generación de gráficos en forma dinámica por medio de programación dentro de una página.

PHP: Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente para la interpretación del lado del servidor.

UML: Lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir.

OMMMA-L: Extensión del lenguaje de modelado UML para la modelación Orientada a Objetos de Aplicaciones Multimedia.

OCL: Lenguaje simple, para escribir restricciones y expresiones sobre elementos de un modelo. El OCL suele ser útil cuando se está especificando un dominio particular mediante el UML y es necesario restringir los valores permitidos para los objetos del dominio.

CASE: Aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Java: Lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

C#: Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

ASP: Tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente.

OSI: Modelo de red descriptivo creado por la Organización Internacional para la Estandarización en el año 1984. Marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

IIS: Servidor web y conjunto de servicios para el sistema operativo Microsoft Windows.

DOM: Interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

Extjs: Librería Javascript ligera y de alto rendimiento, compatible con la mayoría de navegadores para crear páginas web y aplicaciones dinámicas.

jQuery: Framework de Javascript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Scriptaculous: Biblioteca Javascript que permite el uso de controles AJAX, drag 'n drop, y otros efectos visuales en una página web.

GLOSARIO DE TÉRMINOS

Dojo: Framework que contiene APIs y controles para facilitar el desarrollo de aplicaciones web que utilicen tecnología AJAX.

AJAX: Técnica de desarrollo web para crear aplicaciones interactivas. Javascript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax.

OpenLaszlo: plataforma de código abierto, ofrecida bajo la Licencia Pública Común (Common Public License o CPL), para el desarrollo y la entrega de aplicaciones de Internet enriquecidas (RIA por su sigla en inglés).

BlueFlish: software con licencia GPL dirigido a diseñadores y programadores web experimentados que se enfoca en la edición de páginas dinámicas e interactivas. Es capaz de reconocer diversos lenguajes de programación.

Synfig: editor de gráficos vectoriales y una herramienta de animación por computadora libre.